INTEROP PROGRAM

PVP mit der .NET Plattform

Dokumenteninformation

Feld	Wert
Autor	Werner Kugler, Rubicon Informationstechnologie GmbH.
Titel	PVP mit der .NET Plattform
Projekt	Interop Program
Zuletzt gespeichert von	Werner Kugler, Rubicon Informationstechnologie GmbH.
Review	Mario Szpuszta, Microsoft Österreich GmbH.
Revision	12

Änderungen:

Version	Autor	Änderungen
1.0	Werner Kugler	Erste Version

Inhalt

1 Einführung

1.1 Inhalt dieses Dokuments

Dieses Dokument beschreibt die praktische Umsetzung von PVP mit Microsoft Technologien. Im ersten Kapitel geht es um den Portalverbund allgemein, im zweiten Kapitel um die Stammportal-Seite, im dritten Kapitel um die Anwendungsportal-Seite.

2 Portalverbund Allgemein

2.1 Portalverbund

Beim Portalverbund geht es um Benutzer- und Rechteverwaltung von Internetapplikationen, in erster Linie im behördlichen Umfeld. Die Benutzerverwaltung, mit den einem Benutzer zugeordneten Rechten, wird dabei von der Applikation zu einer vorgelagerten Instanz verlagert, dem Stammportal. Dieses Stammportal genießt eine Vertrauensstellung. Wie das Stammportal zu dieser Vertrauensstellung kommt, und welche Voraussetzungen gegeben sein müssen, ist im Portalverbund (PV) geregelt. Die technische Kommunikation zwischen Stammportal und den dahinter liegenden Applikationen ist im Portalverbund-Protokoll (PVP) spezifiziert. Hier wird eine Übersicht geboten, Details wurden zur leichteren Lesbarkeit absichtlich weggelassen. Das bedeutet, dass dieses Kapitel zum Ziel hat, einen Einstieg zu bieten, und nicht die PVP Spezifikation zu ersetzen.

Vorab sollen einige Begriffe konkretisiert werden:

Begriff	Erklärung
Authentifizierung	Feststellen der Identität eines Benutzers.
Autorisierung	Ausstatten eines Benutzers mit bestimmten Rechten, eine vorhergehende Authentifizierung ist Voraussetzung.
Recht	Ein Recht gibt dem Benutzer die Möglichkeit, in einer bestimmten Applikation eine bestimmte Aktion durchzuführen. Das bedeutet, ein Recht bezieht sich immer auf eine bestimmte Applikation. Eine Applikation stellt mehr oder weniger Aktionen zu Verfügung, ja nach den Rechten des Benutzers der Applikation.

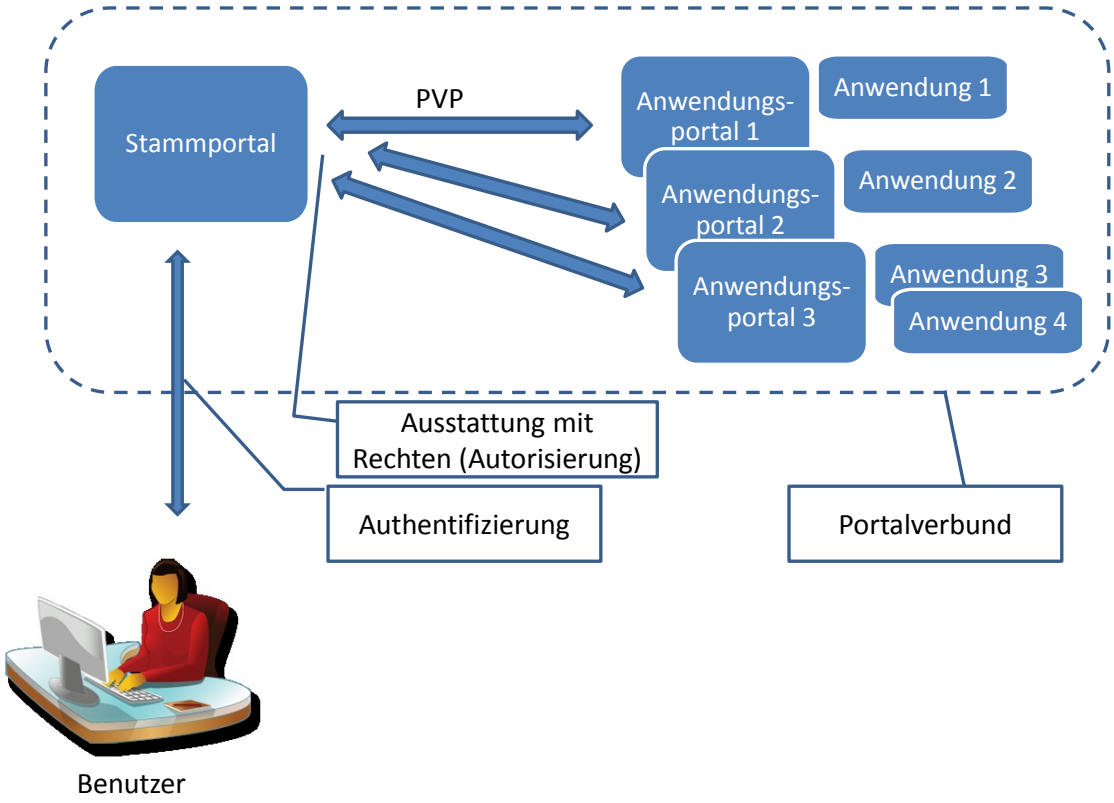

Abbildung 1: Portalverbund

2.2 Single Sign On

Ein Stammportal kann für einen Benutzer den Zugang zu mehreren PVP konformen Anwendungen anbieten. Daraus ergibt sich die administrative Vereinfachung. Ein Benutzer wird einmal am Stammportal eingerichtet, meldet sich einmal am Stammportal an, und kann dadurch viele Applikationen nutzen. Bei den einzelnen Applikationen wird der Benutzer nicht mehr geführt. Bei Nutzung des egora Stammportals entfällt auch die Anmeldung am Stammportal, da Windows Integrated Authentication verwendet werden kann.

2.3 Proxy

Die Authentifizierung eines Benutzers erfolgt gegenüber einem Stammportal. Auch alle weiteren Zugriffe auf die Applikation laufen über das Stammportal, das Stammportal agiert als Proxy. Das bedeutet, es gibt keine direkte Kommunikation zwischen Benutzer und Anwendung, sondern alle Kommunikation geht über das Stammportal.

2.3.1 Url Namensraum

Da ein Stammportal sinnvollerweise den Zugriff auf mehrere Anwendungen zu Verfügung stellt, die auf verschiedenen Servern laufen, könnte es sein, dass zwei verschiedene Anwendungen die gleiche Url relativ zum Servernamen verwenden wollen, beispielsweise https://awp.org1.gv.at/start.htm und https://awp.org2.gv.at/start.htm.

Um ein Umschreiben von Urls nicht erforderlich zu machen, wird der Stammportal Proxy oft so konfiguriert, dass die Urls bis auf den Servernamen nicht verändert werden. Dadurch käme es bei den Urls https://awp.org1.gv.at/start.htm und https://awp.org2.gv.at/start.htm zu einem Konflikt, weil beide am Stammportal auf https://stammportal/start.htm abgebildet wären.

Damit dieses Problem nicht auftritt, werden PVP Anwendungen so publiziert, dass sie die Domain in der Url wiederholen. So sind dann die Urls https://awp.org1.gv.at/org1.gv.at/start.htm und https://awp.org2.gv.at/org2.gv.at/start.htm.

2.4 Akteure

Am Portalverbund beteiligte technische Akteure:

Akteur	Aufgabe
Stammportal	Authentifiziert einen Benutzer. Stattet einen Benutzer mit Rechten aus (Autorisierung). Leitet alle Requests an die Anwendungsportale weiter. Die Requests werden dem PVP Protokoll entsprechend aufbereitet. Die Kommunikation erfolgt mittels https mit Client Certificate.
Anwendungsportal	Steht vor einer Anwendung. Prüft das Zertifikat, mit dem die Stammportale die https Verbindung aufbauen.
Anwendung	Stellt die gewünschte Funktion zu Verfügung, je nach Recht. Verlässt sich auf Stammportal und Anwendungsportal.

Am Portalverbund beteiligte organisatorische Akteure:

Akteur	Aufgabe
Portalverbund Teilnehmer	Eine Organisation, die dem Portalverbund beitritt, zum Beispiel eine Gemeinde.
Stammportal Betreiber	Eine Organisation oder Firma, die ein Stammportal betreibt. Wenn eine Gemeinde ein Stammportal nicht selbst betreiben will, kann sie einen Dienstleister damit beauftragen.
Anwendung Betreiber	Eine Organisation, die eine Anwendung zur Verfügung stellt. Sie muss PVP relevante Informationen (Rechte, URL, SecClass) veröffentlichen.

2.5 Portalverbund-Protokoll

Das Portalverbund-Protokoll regelt Aspekte der Kommunikation zwischen Stammportal und Anwendungsportal.

2.5.1 https

Die Kommunikation zwischen Stammportal und Anwendungsportal erfolgt mittels https, wobei beide Partner Zertifikate benötigen. Das Client Zertifikat muss die Zertifikatserweiterung "Verwaltungseigenschaft" aufweisen. Siehe auch http://www.digitales.oesterreich.gv.at/DocView.axd?CobId=19383

2.5.2 Http-Header

Bei gewöhnlichen (html) Requests werden PVP Informationen als Http-Header gesandt, also vor dem eigentlichen Inhalt des Requests.

2.5.3 SOAP-Header

Bei SOAP Nachrichten werden PVP Informationen als SOAP-Header gesandt, also als Teil der SOAP-Nachricht.

2.5.4 PVP Informationen

Die http-Header, die die PVP Informationen beinhalten, beginnen immer mit "X-", weil es sich um sogenannte custom header handelt. Der Name wird ohne diesem Präfix dargestellt.

Version

Die PVP Informationen enthalten eine Versionsnummer:

Name	Wert	Länge
Version	die vom Client implementierte PVP-Version.	4

AUTHENTICATE

Die PVP Informationen gliedern sich in mehrere Blöcke. Der Name des http-Headers besteht aus dem Präfix "X-", dem Block und dem in der Tabelle angeführten Namen, beispielsweise X-AUTHENTICATE-UserID.

Name	Wert	Länge
participantId	Org-ID des Participants, bei dem der Benutzer registriert ist.	21
UserID	UserID, mit der der Benutzer am Stammportal authentifiziert ist. (LDAP: gvOrgPerson/uid) oder abgekürzte Bezeichnung des System-Principals in der Form Anwendung.Subsystem.	128
Cn	Name des Benutzers (LDAP: gvOrgPerson/cn) oder des System-Principals in der Form Anwendung.Subsystem	64
gvOuId	Stammdienststelle: Verwaltungskennzeichen [VKZ] (LDAP: gvOrgUnit/gvOuVKZ) der Organisationsein-heit des Benutzers, bei System-Principals des Anwendungsverantwortlichen. Die zugehörige Organisationseinheit ergibt sich entweder aus einer eindeutigen Zuordnung oder eine Auswahl des Benutzers, wenn mehrere definiert sind.	32
Ou	Kurzbezeichnung der Organisationseinheit	64
SecClass	Sicherheitsstufe des Benutzers Fehlt dieser Header, wird die Sicherheitsklasse „1“ angenommen.	1

Name	Wert	Länge
Mail	E-Mail-Adresse des Benutzers. Hauptzweck ist die direkte Erreichbarkeit des Benutzers, aber auch die Verwendung in zukünftigen PV-Anwendungen. Die Darstellung ist ohne Display Name und ohne Quotes (also im Format name@domain).	128
Tel	Telefonnummer des Benutzers (gvOrgperson/telephoneNumber)	32
gvGid	Global Identifier des BenutzersLDAP: gvOrgPerson/gvGid	128
gvFunction	Entspricht Funktion in gvPersonFunction. Verpflichtend, wenn für eine Person Funktionen definiert sind.LDAP: gvPersonFunction/gvFunction	32
Bpk	bereichsspezifisches Personenkennzeichen mit einem Präfix bestehend aus „bPK:“, dem Kürzel laut Bereichsabgrenzungsverordnung, „:“ und dem bPK. Im Falle eines verschlüsselten bPK lautet das Präfix nur „vbPK:“ ohne Spezifikation des Bereiches. Beispiel: bPK:PV:NxdRQhp+tNyE9WhHdBSYuy3hA=	256

AUTHORIZE

Der Name des http-Headers besteht aus dem Präfix "X-", dem Block und dem in der Tabelle angeführten Namen, beispielsweise X-AUTHORIZE-roles.

Name	Wert	Länge
gvOuId	Auftraggebende Dienststelle: Eindeutige Kennung für die Organisationseinheit des Benutzers (LDAP: gvOrgUnit/gvOuId) bei System-Principals: Organisationseinheit des Anwendungsverantwortlichen	32
Ou	Auftraggebende Dienststelle: Verwaltungskennzeichen [VKZ] der mit AUTHENTICATE-gvOuId bezeichneten Organisationseinheit (LDAP: gvOrgUnit/ou). Bei System-Principals: Organisationseinheit des Anwendungsverantwortlichen	64
roles	Anwendungsrechte, optional mit Rechte-Parametern. LDAP: gvApplicationRight/cn cn (Rechte) und gvUserRestriction (Rechte-Parameter, z.B. GKZ, DST, BL, gvOuId) Wenn die Einschränkung der Zeichencodierung auf ISO-Latin für die Anforderungen der Anwendung nicht ausreichend ist, soll die Anwendung eine Codierung wie etwa BASE64 vorgeben.	32767

2.6 Weiterführende Informationen

Einstieg: http://www.digitales.oesterreich.gv.at/site/5288/default.aspx

Referenz: http://reference.e-government.gv.at/Portalverbund.577.0.html

3 Stammportal

Im Folgenden wird die Funktionsweise und Verwendung des egora Stammportals erläutert.

3.1 Architektur

3.1.1 Authentifizierung

Das egora Stammportal läuft als Anwendung im IIS. Prinzipiell stehen daher alle vom IIS zur Verfügung gestellten Authentifizierungsmechanismen zur Verfügung. Darüber hinaus könnte man Authentifizierungsmodule einhängen.

Windows Integrated Authentication

In der Praxis wird Windows Integrated Authentication verwendet.

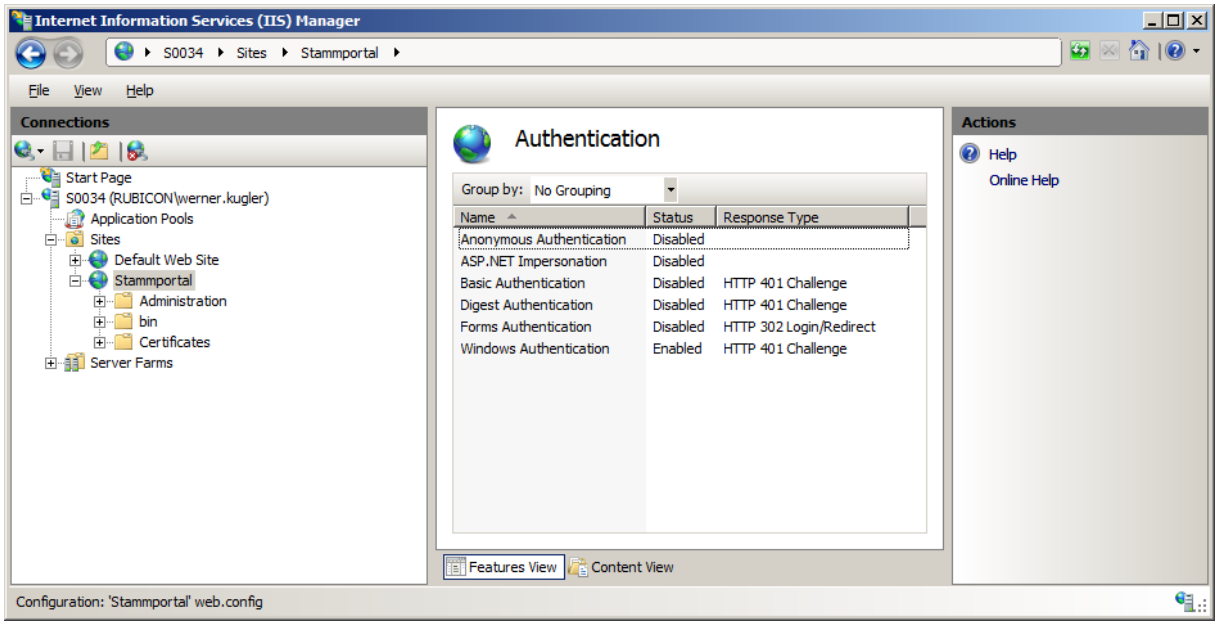

Abbildung 2: Windows Authentication

Client Certificate Authentication

Falls SecClass 3 benötigt wird (Wissen und Besitz), kann Client Certificate Authentication verwendet werden. Dazu muss in der Web.config unter dem Pfad system.webServer/security/authentication/clientCertificateMappingAuthentication der Wert true eingetragen sein. In folgender Abbildung zu sehen mit dem Configuration Editor aus dem IIS Administration Pack.
http://www.iis.net/extensions/AdministrationPack

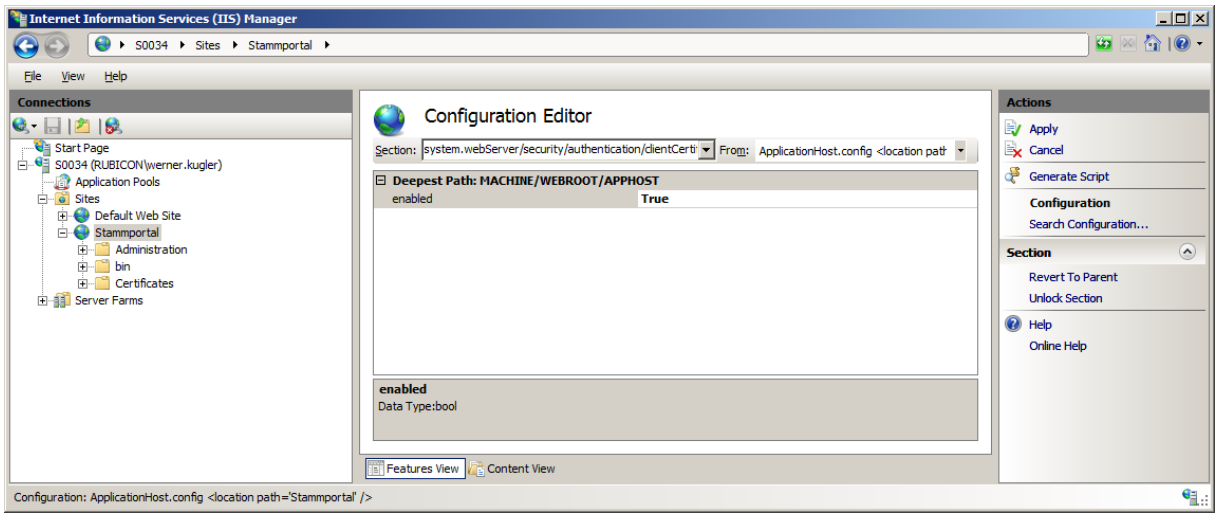

Abbildung 3: ClientCertificateMappingAuthentication

Der User muss dabei ein Zertifikat aus der Active Directory Infrastruktur zur Verfügung haben.
http://technet.microsoft.com/de-de/library/cc534992(en-us).aspx
Dieses Zertifikat muss auf einem externen Medium, z.B. USB-Stick, liegen und mit einem PIN gesichert sein, um die Forderung nach Wissen und Besitz zu erfüllen.

Eine weitere Möglichkeit, die Forderung nach Wissen und Besitz zu erfüllen, wäre bereits beim Windows Logon die Verwendung einer Smartcard zu erzwingen. Dieses Thema sprengt aber den hier vorgegebenen Rahmen.

3.1.2 Autorisierung

Die Autorisierungsdaten, die das Stammportal wissen muss, um den Request PVP konform an das Anwendungsportal weiterleiten zu können, werden von einem Webservice abgeholt. Dabei werden folgende Informationen ausgetauscht:

IN:

- UserId, wie durch die Authentifizierung gewonnen
- RootUrl, als Identifizierung der Anwendung

OUT:

- Headers als Name-Value Pairs

  oder

- XmlFragment,
  falls es sich um einen SOAP Request handelt

Durch die Anbindung der Autorisierung mittels Webservice ergibt sich eine große Flexibilität was die Quelle der Autorisierungsinformationen anlangt. Im Rahmen des egora Stammportals wurde ein Autorisierungsservice implementiert, das die Informationen aus einem LDAP Verzeichnis holt, vorzugsweise dem Active Directory.

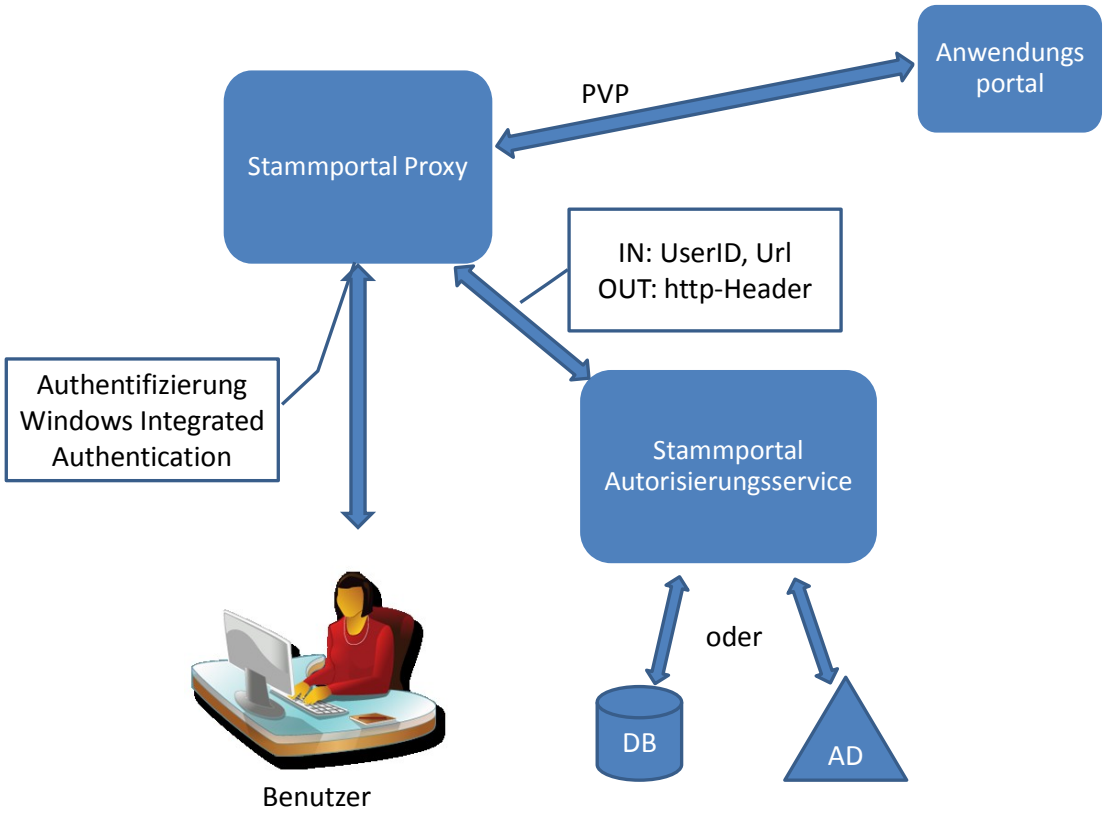

Abbildung 4: Autorisierungsservice

3.2 Voraussetzungen Software

3.2.1 Betriebssystem

Als Betriebssystem wird Windows Server 2008 – Standard Edition empfohlen. Die im Folgenden angeführten Komponenten des Betriebssystems sind erforderlich:

- Windows Server Betriebssystem mit IIS 6 oder IIS 7
- .NET Framework 2.0 mit ASP.NET Komponenten

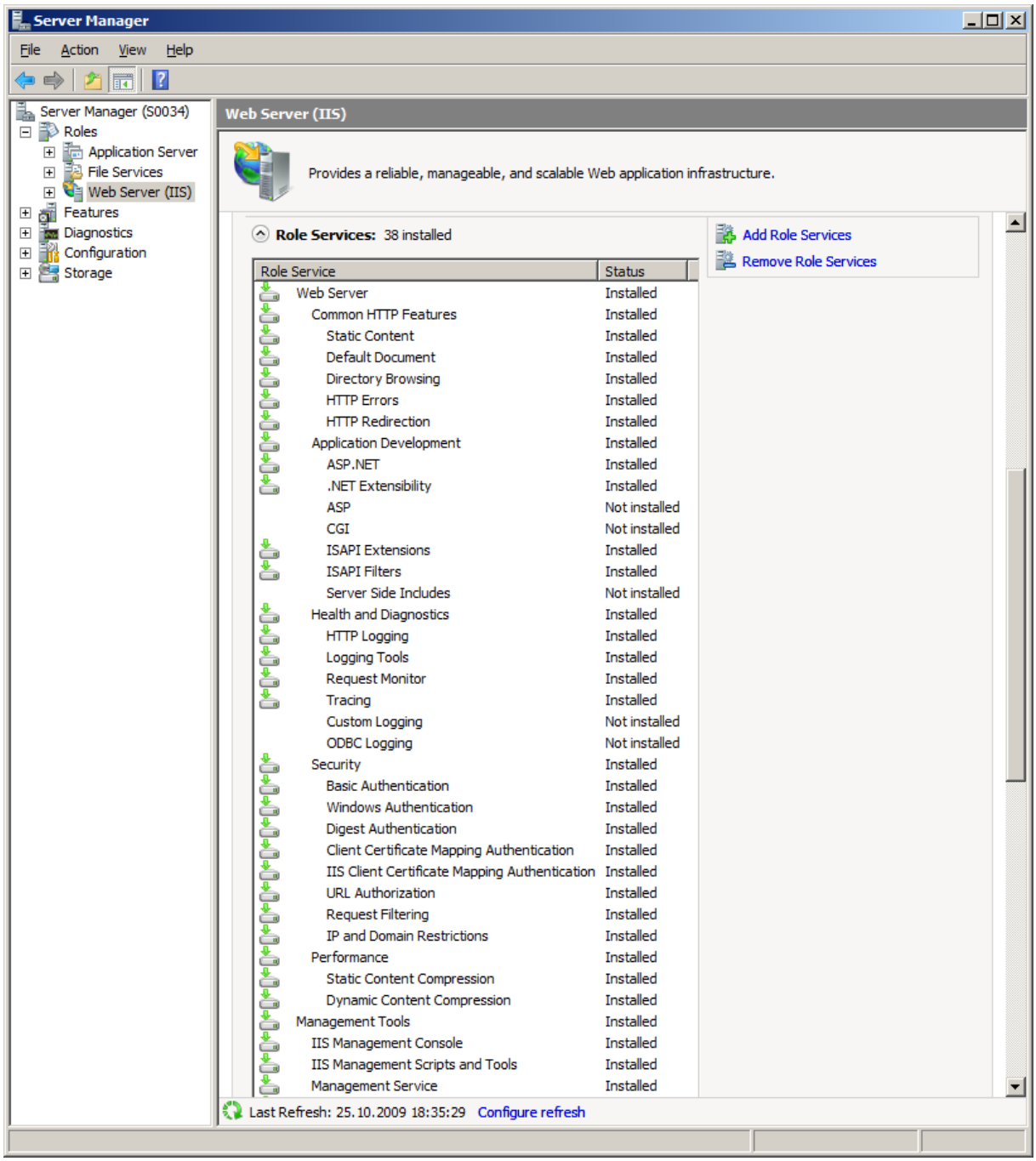

Abbildung 5: Installierte Applikationsserverkomponenten

3.3 Konfiguration Internet Information Services

3.3.1 Quellen

Das egora Stammportal ist als Sourcecode verfügbar unter http://egora.codeplex.com. Es ist eine VisualStudio 2008 Solution. Kompilieren Sie die Solution. Kopieren Sie folgende Dateien aus dem Verzeichnis HttpReverseProxy in das Verzeichnis C:\Inetpub\wwwroot\HttpReverseProxy bzw in die entsprechenden Unterordner:

- Mapping.xml

- PathMap.xsd
- Web.config
- bin*.*
- Administration*.aspx

3.3.2 Konfiguration Application-Pool

Für den Betrieb des Reverse Proxy empfehlen wir einen eigenen Application Pool.

Legen Sie dazu einen neuen Application-Pool mit der Bezeichnung „Stammportal“ an und übernehmen Sie ansonsten die Standard-Einstelllungen. Wir empfehlen, die Netzwerkkonfiguration so vorzunehmen, dass sich der Reverse Proxy direkt mit dem Internet (bzw. den Zielapplikationen) verbinden kann. Der Zugriff auf das Internet über einen Proxy ist prinzipiell möglich und kann konfiguriert werden (siehe Kapitel 3.5.1). Der Application Pool soll unter Network Service laufen, dazu unter Advanced Settings folgenden Dialog aufrufen:

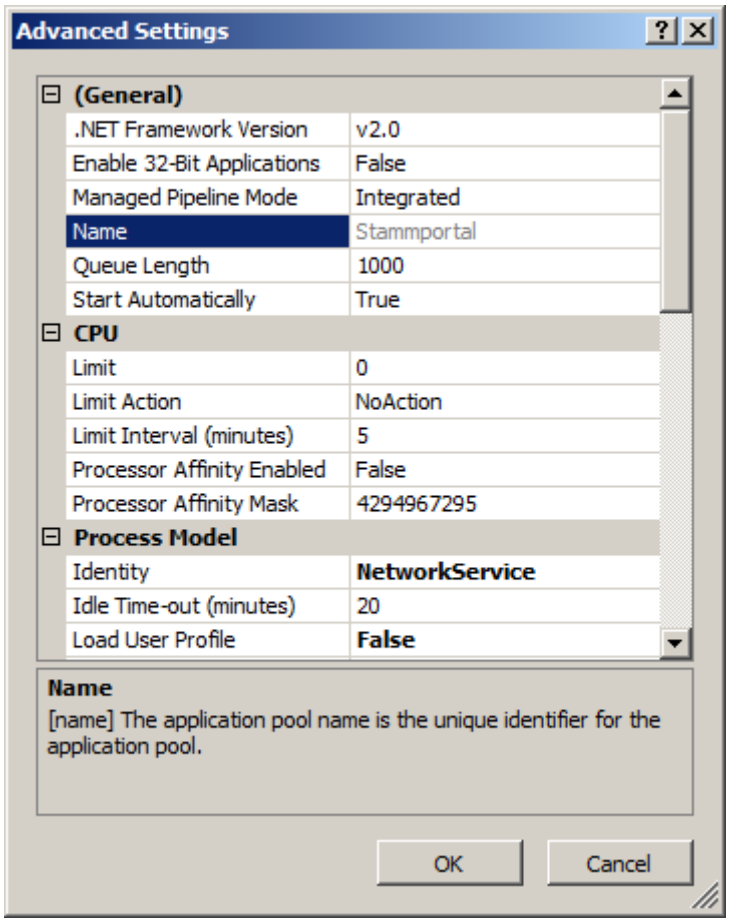

Abbildung 6: Identity Application Pool

3.3.3 Konfiguration Web-Applikation

Legen Sie eine neue Site „Stammportal“ im Microsoft Internet Information Services Manager an.

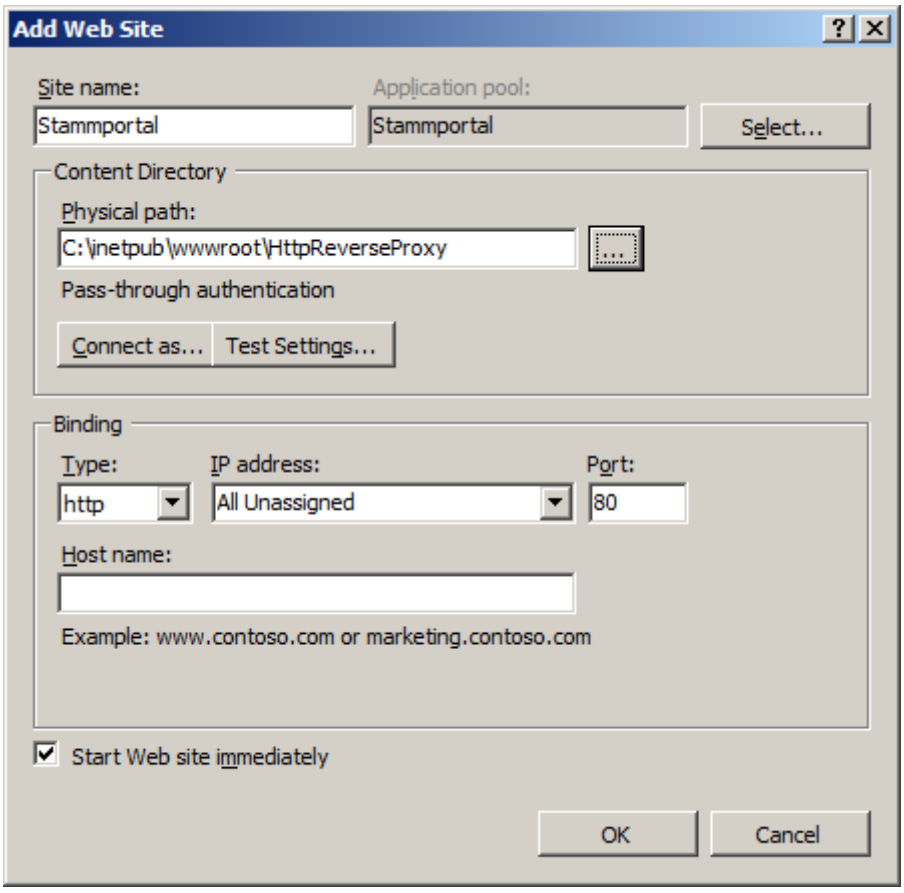

Abbildung 7: Anlegen Site

3.3.3.1 Authentication

Schalten Sie Windows Authentication ein, schalten Sie Anonymous Authentication aus.

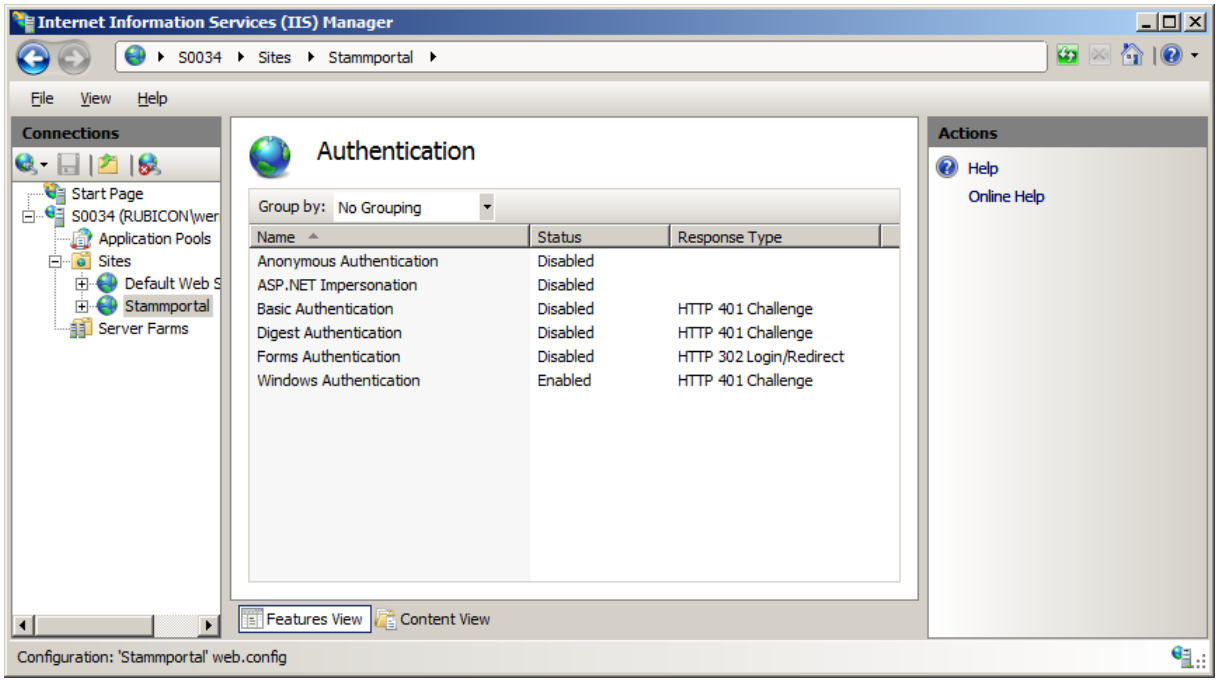

Abbildung 8: Site Stammportal Authentication

3.4 Zertifikate

Damit der Reverse Proxy bei dem Aufbau der https Verbindung das erforderliche Zertifikat zur Verfügung hat, sind folgende Schritte erforderlich:

1. Öffnen Sie die Microsoft Management Konsole.
2. Fügen Sie ein Snap-In für die Hinterlegung von Zertifikation am lokalen Computer ein.
3. Öffnen Sie den Zertifikate-Zweig und rufen Sie aus dem Kontextmenü am Ordner *Personal* aus dem Menü ALL TASKS den Menüpunkt IMPORT ... auf.

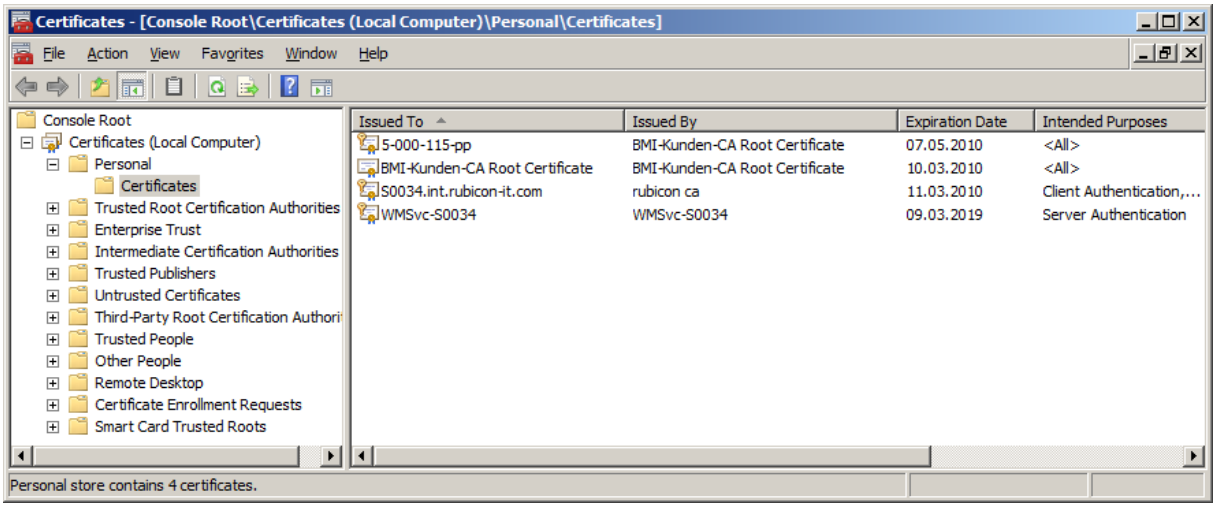

Abbildung 9: Zertifikat - Zertifikat importieren

4. Importieren Sie das Zertifikat. Das Zertifikat muss den öffentlichen als auch privaten Schlüssel beinhalten.
5. Wählen Sie beim importierten Zertifikat im Kontektmenü den Menüpunkt All Tasks -> Manage Private Keys ... und geben Sie dem User des Application Pools Leserechte.

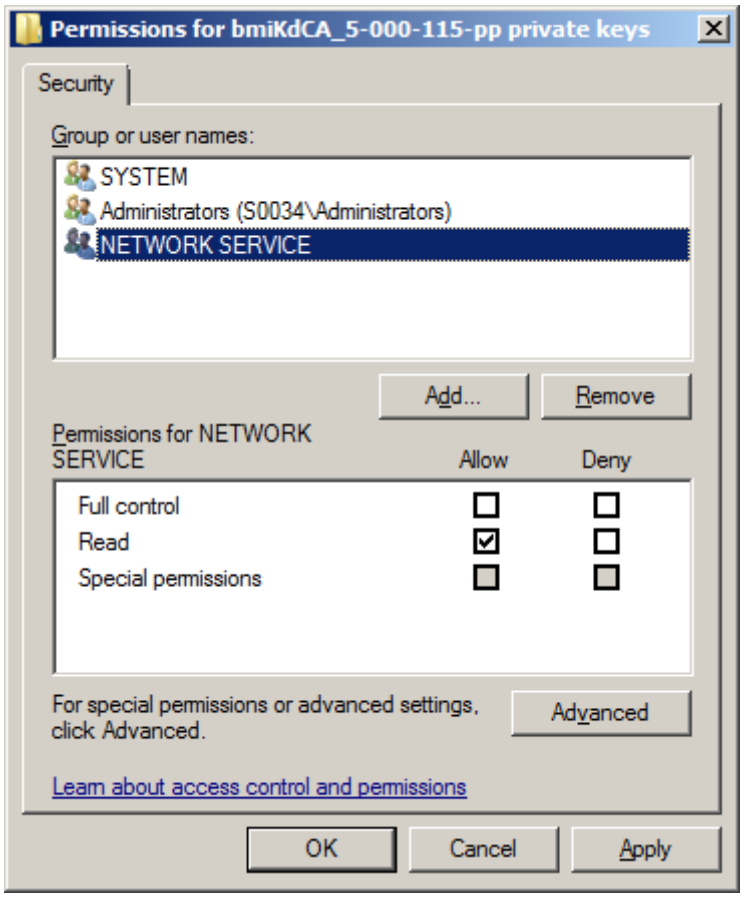

Abbildung 10: Zertifikate - Zusätzliche Rechte

6. Wechseln Sie zurück in die geöffnete Microsoft Management Console und rufen auf dem zuvor importieren Zertifikat das Kontextmenü mit der rechten Maustaste auf, um das Zertifikat zu exportieren.

Abbildung 11: Zertifikat - Zertifikat exportieren

7. Speichern Sie das Zertifikat ohne private Key im Verzeichnis C:\Inetpub\wwwroot\httpreverseproxy\Certificates ab. Den Dateinamen, den Sie hier vergeben, müssen Sie im Mapping.xml im HttpReverseProxy-Verzeichnis entsprechend eintragen.

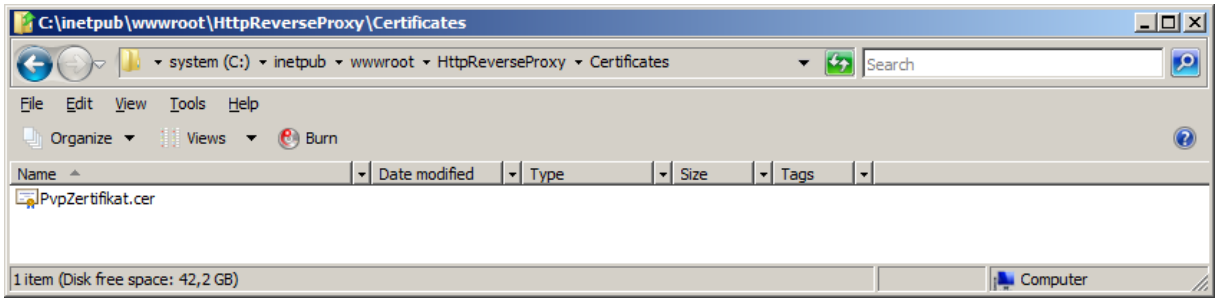

Abbildung 12: Zertifikat - Zertifikat exportieren, Speicherort

8. Beim Exportieren des Zertifikates wird der private Schlüssel des Zertifikates nicht mit exportiert!

3.5 Konfiguration

3.5.1 web.config

Die Datei web.config im Verzeichnis HttpReverseProxy kann angepasst werden.

Die Konfiguration einer ASP.NET Anwendung ist sehr mächtig, beispielsweise kann für http Requests ein Defaultproxy definiert werden. Siehe
http://msdn.microsoft.com/library/default.asp?url=/library/en-

us/cpguide/html/cpconASPNETConfiguration.asp Wir können hier nicht alle Aspekte schildern, sondern erwähnen nur das Tracing von Requests.

3.5.1.1 Tracing

```
<switches>
  <add name="reverseProxySwitch" value="All"/>
  <add name="System.Net" value="Verbose" />
  <add name="System.Net.Sockets" value="Verbose" />
</switches>
```

Hier kann eingestellt werden, wieviel Informationen in den Tracefiles landet.

ActivityTracing	Allows the Stop, Start, Suspend, Transfer, and Resume events through.
All	Allows all events through.
Critical	Allows only Critical events through.
Error	Allows Critical and Error events through.
Information	Allows Critical, Error, Warning, and Information events through.
Off	Does not allow any events through.
Verbose	Allows Critical, Error, Warning, Information, and Verbose events through.
Warning	Allows Critical, Error, and Warning events through.

```
<add
  name="NetTraceFile"
  type="System.Diagnostics.TextWriterTraceListener"
  initializeData=" c:\log\System.Net.trace.log" />

<add name="fileListener"
  type="System.Diagnostics.DelimitedListTraceListener"
  delimiter="|"
  initializeData="c:\log\ReverseProxy.log" traceOutputOptions="ProcessId, ThreadId, DateTime">
  <filter type="System.Diagnostics.EventTypeFilter"
    initializeData="All"/>
</add>
```

Hier kann eingestellt werden, wie die Tracefiles heißen und in welchem Verzeichnis sie liegen. Der Benutzer des Application Pools muss Schreibrechte auf die angegebenen Dateien haben (bzw. das Recht des Anlegens). Für ein Produktionssystem empfehlen wir, den TraceLevel auf Error oder geringer zu stellen, da sonst das Tracefile sehr rasch sehr groß wird.

3.5.1.2 Settings

Eine Änderung der Einstellung ist normalerweise nicht nötig. Die Einstellungen sind der Vollständigkeit halber gelistet.

```
<Egora.Stammportal.HttpReverseProxy.Properties.Settings>
 <setting name="AdministrationGroup" serializeAs="String">
  <value>BUILTIN\Administrators</value>
 </setting>
 <setting name="AdministrationPath" serializeAs="String">
  <value>admin</value>
 </setting>
 <setting name="PathMapFile" serializeAs="String">
  <value>~/Mapping.xml</value>
 </setting>
 <setting name="AuthorizationWebServiceDefault" serializeAs="String">
  <value> http://localhost/TestAuthorizationWebService/PvpAuthorizer.asmx</value>
```

```
</setting>
<setting name="HistoryLength" serializeAs="String">
  <value>100</value>
</setting>
<setting name="ImpersonateWebRequest" serializeAs="String">
  <value>False</value>
</setting>
<setting name="AuthenticationLevel" serializeAs="String">
  <value>MutualAuthRequested</value>
</setting>
<setting name="ProcessRequestWithoutAuthorization" serializeAs="String">
  <value>False</value>
</setting>
<setting name="RemoveLeftSideAuthorization" serializeAs="String">
  <value>True</value>
</setting>
<setting name="RequestTimeoutSeconds" serializeAs="String">
  <value>300</value>
</setting>
<setting name="ConnectionsPerServer" serializeAs="String">
  <value>50</value>
</setting>
<setting name="ConnectionMaxIdleTimeSeconds" serializeAs="String">
  <value>10</value>
</setting>
<setting name="RetryableErrorMessages" serializeAs="String">
  <value>connection that was expected to be kept alive was closed by the server</value>
</setting>
<setting name="RemoveAuthorizationHeader" serializeAs="String">
  <value>Negotiate NTLM</value>
</setting>
<setting name="BufferLeftSide" serializeAs="String">
  <value>False</value>
</setting>
<setting name="BufferRightSide" serializeAs="String">
  <value>False</value>
</setting>
<setting name="NetworkRetryDelay" serializeAs="String">
  <value>500</value>
</setting>
<setting name="NetworkRetryCount" serializeAs="String">
  <value>3</value>
</setting>
<setting name="RetryableHosts" serializeAs="String">
  <value>pvawp.bmi.gv.at;localhost</value>
</setting>
</Egora.Stammportal.HttpReverseProxy.Properties.Settings>
```

AdminstratorGroup

Mitglieder dieser Active Directory Gruppe können die Administrationsseiten des Reverse Proxy aufrufen.

AdministrationPath

Der virtuelle Pfad zu den Administrationsseiten. Standard ist admin. Die Übersicht der Urls, die vom Proxy bedient werden, findet man daher standardmäßig unter http://server/admin/Applications.aspx

PathMapFile

Diese Angabe verweist zu der Konfiguration für die Urls, die vom Reverse Proxy bedient werden (siehe Kapitel 0).

AuthorizationWebServiceDefault

Default Url des Autorisierungsservice.

HistoryLength

Der Reverse Proxy merkt sich für die Administrationsseiten die hier eingestellte Anzahl an Requests pro konfigurierter Applikation.

ImpersonateWebRequest

Falls ein Proxy definiert ist, der eine Impersonifizierung des Endbenutzers verlangt, muss der Wert auf true gestellt werden.

AuthenticationLevel

MutualAuthRequested	The client and server should be authenticated. The request does not fail if the server is not authenticated. To determine whether mutual authentication occurred, check the value of the WebResponse.IsMutuallyAuthenticated property.
MutualAuthRequired	The client and server should be authenticated. If the server is not authenticated, your application will receive an IOException with a ProtocolViolationException inner exception that indicates that mutual authentication failed
None	No authentication is required for the client and server.

Wir empfehlen die Einstellung None.

ProcessRequestWithoutAuthorization

Falls vom Autorisierungsservice keine Autorisierung geliefert wird (SoapHeaderXmlFragment ist null und HttpHeaders ist null oder NoAuthorization), so wird der Request dennoch weitergereicht, wenn diese Einstellung auf „True“ steht.

RemoveLeftSideAuthorization

Falls diese Einstellung auf „True“ steht, so werden alle Pvp-Headers der am Stammportal **ankommenden** (linke Seite) Requests entfernt. Falls es sich um einen Soap Request handelt, werden alle pvpToken (Elementname: pvpToken, Namespace: http://egov.gv.at/pvp1.xsd) entfernt.

RequestTimeoutSeconds

Der Wert für den Timeout des Requests zum Server (rechte Seite), in Sekunden.

ConnectionMaxIdleTimeSeconds

Connections, die diese Zeitspanne inaktiv waren, werden geschlossen. Standardwert 10, weil der Standardwert für einen Apache-Server 15 ist.

ConnectionsPerServer

Pro Zielserver werden maximal diese Anzahl an Connections aufgemacht. Auf der Administrationseite Connection.aspx kann man die aktuell verwendete Anzahl sehen.

RetryableErrorMessages

Hier können Teile von Fehlemeldungen mit „;“ getrennt angegeben werden. Falls bei einem Request zum upstream Server eine Exception auftritt, kann dieser Request wiederholt werden. Es werden nur solche Request wiederholt, bei denen die Fehlermeldung einen der angegebenen Teile enthält.

RemoveAuthorizationHeader

Grundsätzlich werden alle Headers vom Client zum upstream Server weitergegeben. Authorization Header mit den hier angeführten Typen werden nicht zum upstream Server wetergegeben. Das egora Stammportal läuft unter Windows Integrated Authentication, daher kommen am Stammportal die Authorization Header Negotiate odeer NTLM an. Diese werden nicht zum Anwendungsportal weitergeleitet.

BufferLeftSide

Der Inhalt eines Requests vom Client wird normalerweise direkt in den Request zum upstream Server geschriegen. Steht diese Einstellung auf True, so wird der Inhalt zuerst in einen Puffer geschrieben. Falls NetworkRetryCount > 0 ist, wird unabhängig von dieser Einstellung immer in einen Puffer geschrieben.

BufferRightSide

Der Inhalt einer Response vom upstream Server wird normalerweise direkt in die Response zum Client geschriegen. Steht diese Einstellung auf True, so wird der Inhalt zuerst in einen Puffer geschrieben.

NetworkRetryCount

Ist dieser Wert größer als 0, werden Request zum upstream Server, die entweder mit einer Exception enden oder mit dem Statuscode 500 enden, wiederholt. Dieser Wert bestimmt die Anzahl der Wiederholungen.

NetworkRetryDelay

Ist dieser Wert bestimmt, wielange vor einer Wiederholung eines Requests gewartet wird. Der Wert ist in Millisekunden.

RetryableHosts

Hier können Hostnamen mit „;“ getrennt angegeben werden. Falls einRequest zum upstream Server mit dem Statuscode 500 beantwortet wird, kann dieser Request wiederholt werden. Es werden nur solche Request wiederholt, die zu einem der genannten Server gehen.

SubstituteHostInLocationHeader

Wenn dieser Switch auf True gesetzt ist, wird bei einem Location Header (redirect), der die Authority (Servername) enthält, die Authority ersetzt durch die Authority des Stammportals.

3.5.2 Admin Seiten

Unter dem in der web.config eingestellen Pfad (siehe 3.5.1.2) können aspx Seiten aufgerufen werden, die Auskunft geben über den aktuellen Status des Proxy. Der Aufruf ist Case sensitiv.

3.5.2.1 Applications.aspx

Diese Seite listet die Applikationen, die vom Proxy derzeit bedient werden. Wenn eine Applikation im Mapping eingetragen ist, aber noch nie aufgerufen wurde, ist sie hier nicht gelistet. Bei Klick auf Details sieht man eine Historie der Requests.

3.5.2.2 Authorization.aspx

Hier sieht man die im Cache befindlichen Autorisierungsdaten.

3.5.2.3 Connection.aspx

Hier sieht man die aktuell geöffneten Connections zu den Zielservern.

3.5.2.4 Reset.aspx

Bei Aufruf dieser Seite wird der Proxy neu initialisiert. Das Mapping wird neu eingelesen und der Cache der Autorisierungsdaten geleert.

3.5.3 Mapping.xml

Diese Datei definiert die Urls, die der Reverse Proxy bedient. Hier ein Beispiel:

```
<?xml version="1.0" encoding="utf-8" ?>
<PathMap xmlns="http://www.egora.at/Stammportal/PathMap/1.0" >
 <Directories>
  <Directory Name="bmi.gv.at">
   <Directories>
    <Directory Name ="soap">
     <Directories>
      <ApplicationDirectory
       Name="zmr"
       AuthorizationWebService="http://localhost/AuthorizationWebsite/LdapAuthorizer.asmx"
       RootUrl="https://pvawp.bmi.gv.at/bmi.gv.at/soap/zmr/"
       CertificateFile="~/Certificates/PvpCertificate.cer" />
     </Directories>
    </Directory>
    <ApplicationDirectory
     Name="zmr"
     AuthorizationWebService="http://localhost/AuthorizationWebsite/LdapAuthorizer.asmx"
     RootUrl="https://pvawp.bmi.gv.at/bmi.gv.at/zmr/"
     CertificateFile="~/Certificates/PvpCertificate.cer" />
   </Directories>
  </Directory>
 </Directories>
</PathMap>
```

3.5.3.1 Directory

Ein Element Directory kann im Knoten Directories Unterelemente des Typs Directory oder ApplicationDirectory enthalten. Dadurch können Verzeichnisse geschachtelt werden.

3.5.3.2 ApplicationDirectory

Ein Element ApplicationDirectory definiert eine entfernte Anwendung, die über den Proxy laufen soll. Im obigen Beispiel wird ein Request auf http://server/stammportal/zmrweb/index.htm umgesetzt auf https://portal.bmi.gv.at/portal/zmr-gw/index.htm. Dabei wird beim Aufbau der https Verbindung das Zeritfikat PvpZmrCertificate.cer verwendet. Der private Key wird dabei von Windows automatisch aus dem Private Key Store verwendet, das Programm selbst hat keinen Zugriff auf den private Key.

RootUrl

Die Basis Url der entfernten Anwendung.

AuthorizationWebService

Die Url für das Autorisierungsservice.

Name

Der relative Name unter dem die entfernte Applikation über den proxy angesprochen wird.

CertificateFile

Die Datei, die Sie beim Exportieren des Zertifikates ohne private Key (siehe Kapitel 3.4) erzeugt haben.

Im obigen Beispiel sind die beiden Anwendungen https://pvawp.bmi.gv.at/bmi.gv.at/zmr/ und https://pvawp.bmi.gv.at/bmi.gv.at/soap/zmr/ definiert. Sie sind am Stammportal unter dem gleichen Pfad anzusprechen.

Beispielsweise könnte eine aufgerufene Url https://EgoraPortal/bmi.gv.at/zmr/start.html sein, oder https:// EgoraPortal/bmi.gv.at/soap/zmr/soap/ZMRService.

Wir empfehlen die Installation auf diese Weise, d.h. egora Stammportal läuft in einer eigenen Website unter /, die Directories werden 1:1 abgebildet, weil dadurch die Urls bis auf den Servernamen vor und hinter dem Stammportal gleich sind.

3.6 Autorisierungsservice

Das Autorisierungsservice ist als WebSite angelegt, um eine einfache Anpassung zu erleichtern.

3.6.1 Installation

Kopieren Sie den Inhalt des Verzeichnises AuthorizationWebsite nach C:\Inetpub\wwwroot\AuthorizationWebsite. Richten Sie einen Application Pool für das Autorisierungsservice ein. Richten Sie eine Application mit diesem Application Pool ein.

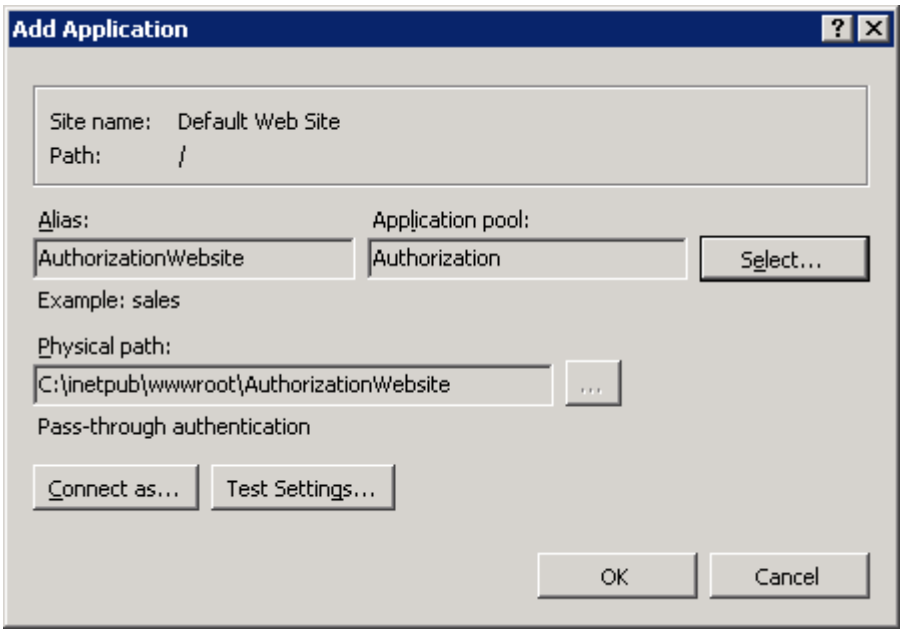

Abbildung 13: Application AuthorizationWebsite

3.6.2 Web.config

```
<Egora.Stammportal.LdapAuthorizationService.Properties.Settings>
  <setting name="ConfigFile" serializeAs="String">
    <value> C:\inetpub\wwwroot\AuthorizationWebsite\Configuration.xml</value>
  </setting>
```

```
    <setting name="CacheGroupResolution" serializeAs="String">
      <value>True</value>
    </setting>
    <setting name="UserFilter" serializeAs="String">
      <value>samAccountName={0}</value>
    </setting>
    <setting name="GroupFilter" serializeAs="String">
      <value>(distinguishedName={0}) </value>
    </setting>
    <setting name="ApplicationGroupFilter" serializeAs="String">
      <value>(&amp;(objectCategory=group)(member={0}))</value>
    </setting>
    <setting name="PvpTokenFormat" serializeAs="String">
      <value>&lt;pvpToken version="{0}" xmlns="http://egov.gv.at/pvp1.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"&gt;
&lt;authenticate&gt;
&lt;participantId&gt;{1}&lt;/participantId&gt;
{2}
&lt;/authenticate&gt;
&lt;authorize&gt;
{3}
&lt;/authorize&gt;
&lt;/pvpToken&gt;</value>
    </setting>
  </Egora.Stammportal.LdapAuthorizationService.Properties.Settings>
```

ConfigFile

Dieses Setting gibt an, wo die Konfigurationsdatei liegt. Die Konfigurationsdatei steuert die Rechtevergabe pro Applikation und wird im nächsten Kapitel erläutert.

CacheGroupResolution

Wenn dieses Setting auf True steht, wird die Schachtelung von Gruppen nur ein mal ausgewertet und gespeichert. Bei False wird die Schachtelung von Gruppen jedes mal ausgewertet.

UserFilter

Gibt die LDAP Query Einschränkung als Format String an, mit der Benutzer gesucht werden.

GroupFilter

Gibt die LDAP Query Einschränkung als Format String an, mit der Gruppen gesucht werden.

ApplicationGroupFilter

Gibt die LDAP Query Einschränkung als Format String an, mit der Gruppen gesucht werden, in denen ein Benutzer Mitglied ist.

Die Settings UserFilter, GroupFilter und ApplicationGroupFilter müssen im Normalfall nicht geändert werden.

PvpTokenFormat

Gibt als Format String an, wie der Pvp Token aufgebaut ist.

Die Parameter sind:

0: Pvp Version

1: Participant ID

2: Informationen zum Benutzer (Principal)

3: Autorisierungsinformationen

Dieses Setting ist nur für SOAP Autorisierung relevant.

3.6.3 Configuration.xml

Mit dieser Datei wird konfiguriert, aus welchen LDAP Informationen die PVP relvanten Autorisierunginformationen geholt werden. In dieser Datei werden Element mit Namen Application angeführt. Eine Element davon kann den Namen Global haben. Die anderen Element erben die Einstellungen von diesem Element.

3.6.3.1 Application

name

Der Name dient nur zur Information, außer dem Name Global, der hat wie oben beschrieben eine besondere Funktion.

ldapRoot

Hier wird der LDAP Pfad angegeben, der als Ausgangspunkt für alle LDAP Queries dient.

webUrls

Hier wird angegeben, für welche Webapplikation dieses Konfigurationselement dient. Als Identifikation dient dabei die Root-Url. Es können auch mehrere Urls mit Leerzeichen getrennt angegeben werden. Wird dieses Attribut angegeben, so handelt es sich um eine Webapplikation und http-Header werden als Autorisierung verwendet.

soapUrls

Hier wird angegeben, für welches Service dieses Konfigurationselement dient. Als Identifikation dient dabei die Url. Wird dieses Attribut angegeben, so handelt es sich um ein SOAP Service und ein SOAP Header wird als Autorisierung verwendet. Es können auch mehrere Urls mit Leerzeichen getrennt angegeben werden.

groupContainer

Hier wird der LDAP Pfad angegeben unter dem die Gruppen liegen, die zur Rollenvergabe herangezogen werden.

recurseGroupMembership

Gibt an, ob Schachtelung von Gruppen ausgewertet wird.

userProperties

Ein Liste von LDAP Eigenschaften, die auf einmal vom User Objekt ausgelesen wird, mit "," getrennt.

authorizationTimeToLive

So lange bleiben die Autorisierungsdaten im ReverseProxy gecached. Die Angabe ist in Sekunden. Nach Ablauf dieser Zeitspanne holt der ReverseProxy die Informationen wieder vom Autorisierungsservice.

domainPrefix

Das Autorisierungsservice erhält als Input den Usernamen. Falls der Username mit domainPrefix\ beginnt, wird dieser Teil abgeschnitten und nur mit dem verbleibenden String im LDAP Verzeichnis gesucht.

soapPrincipal

Hier wird als Format String der Aufbau des soap Principals angegeben. Die Parameter sind:

0: userID

1: cn

2: gvOuId

3: ou

4: mail

5: tel

6: gvSecClass

7: gvGid

8: gvFunction

3.6.3.2 PvpAttribute

In einem Element Application können Elemente mit Namen PvpAttribute angeführt werden.

name

Der Pvp Name des Attributes, wie er als PVP http-Header definiert ist.

source

Die Quelle des Attributewertes im LDAP. Erlaubte Werte sind

User: der Werte wird aus einer Eigenschaft des Benutzer geholt, das ist standard.

Group: der Wert wird aus einer Eigenschaft der Gruppen geholt, in denen der Benutzer Mitglied ist. Diese Einstellung wird beispielsweise für die Rollen (X-AUTHORIZE-roles) verwendet.

UserOrGroup: Der Wert wird zuerst vom User-Objekt gelesen, falls hier kein Wert hinterlegt ist, von den Gruppen-Objekten.

format

Hier kan ein Format String hinterlegt werden, der den Wert aus dem LDAP Objekt formatiert. Standard ist "{0}".

ldapAttribute

Das Attribut, aus dem der Wert gelesen wird

defaultValue

Falls kein Wert im LDAP gefunden wurde, wird dieser Wert benutzt.

3.6.3.3 Beispiel

Das Autorisierungsservice erhält als Input eine UserID und eine Url. aufgrund der Url wird ermittelt, welches Element Application der Konfiguration für die Ermittlung der PVP Informationen herangezogen wird. Aus dem Element Application wird auf ldapRoot eine Query nach der UserID abgesetzt. Eventuell wird hierbei der Domainprefix der UserID entfernt. Danach wird in groupContainer gesucht nach Gruppen, wo der User Mitglied ist. Je nach Einstellung erfolgt die Suche rekursiv für die Gruppen in den Gruppen. Somit sind User-Objekt und Gruppen-Objekte verfügbar und die Elemente PvpAttribute werden ausgewertet.

```
<PvpAttribute name="X-AUTHENTICATE-cn" ldapAttribute="cn" />
```

Der Wert des PVP http-Headers X-AUTHENTICATE-cn wird aus dem User-Objekt gelesen, aus der Eigenschaft cn.

```
<PvpAttribute name="X-AUTHORIZE-roles" ldapAttribute="description" source="Group"/>
```

Der Wert des PVP http-Headers X-AUTHORIZE-roles wird aus den Gruppen, in denen der User Mitglied ist, aus dem LDAP Attribute description geholt.

Je nachdem, ob die das Element Application ausgewählt wurde, weil die Url in den webUrls oder in den soapUrls vorkommt, werden http-Header oder ein Xml-Fragment zusammengestellt. Die Gültigkeitsdauer der Autorisierungsinformationen wird dem Attribut authorizationTimeToLive entnommen. Damit ist der Output des Autorisierungsservices ermittelt.

4 Anwendungsportal

4.1 Aufgaben

Ein Anwendungsportal prüft einen Request bevor er die eigentliche Anwendung erreicht.

4.1.1 Zertifikatsprüfung

Das Client Zertifikat muss gültig sein, das heißt hinsichtlich Zertifikatskette, Gültigkeitsdatum und Widerrufsliste geprüft und für in Ordnung befunden werden. Diese Prüfungen kann der IIS durchführen.

4.1.2 PVP Informationsprüfungen

Das Anwendungsportal kann die PVP Informationen prüfen hinsichtlich verschiedener Kriterien:

- Ist das Client Zertifikat für die participantId zugelassen?
- Ist die participantId für die Applikation generell berechtigt?
- Passen die Rollen mit der Applikation zusammen?
- Ist die für die Anwendung vorgeschriebene Sicherheitsklasse gegeben?

4.2 HttpModule

Im Rahmen der Interop Initiative wurde ein Modul zur Verarbeitung der PVP Informationen implementiert. Diese Modul stellt keine eigenständige Applikation da, ist kein Proxy. Es wird im IIS bei der Anwendung eingehängt und läuft in der Verarbeitung des Requests mit, beim Event AuthorizeRequest. Es läuft somit im Prozess der eigentlichen Applikation. Siehe auch http://msdn.microsoft.com/de-de/library/bb470252.aspx.

4.3 Voraussetzungen Software

Als Betriebssystem wird Windows Server 2008 – Standard Edition empfohlen. Die im Folgenden angeführten Komponenten des Betriebssystems sind erforderlich:

- Windows Server Betriebssystem mit IIS 6 oder IIS 7
- .NET Framework 2.0 mit ASP.NET Komponenten

4.4 Konfiguration Internet Information Services

4.4.1 Quellen

Das egora PVP Modul ist als Sourcecode verfügbar unter http://egora.codeplex.com. Es ist Teil einer VisualStudio 2008 Solution. Kompilieren Sie die Solution. Kopieren Sie folgende Dateien aus dem Verzeichnis PvpHttpModule\bin\Debug bzw. PvpHttpModule\bin\Release in das Verzeichnis bin Ihrer Applikation:

Egora.Pvp.*
Egora.PvpHttpModule.*

4.4.2 Konfiguration Web.config

Im Fall eines IIS 7 müssen Sie folgende Zeilen einfügen:

```
<system.webServer>
  <modules>
    <add name="PvpHttpModule" type="Egora.PvpHttpModule.PvpModule, Egora.PvpHttpModule"/>
  </modules>
</system.webServer>
```

Im Falle eines IIS 6:

```
<system.web>
  <httpModules>
    <add name="PvpHttpModule" type="Egora.PvpHttpModule.PvpModule, Egora.PvpHttpModule"/>
  </httpModules>
</system.web>
```

Wenn es in Ihrer Web.config das Element modules bzw. httpModule bereits gibt, ist nur die Zeile add hinzuzufügen.

Siehe auch http://msdn.microsoft.com/de-de/library/ms178683(VS.80).aspx

4.5 Verwendung

4.5.1 IPrincipal

Im .NET Framework ist das Interface IPrincipal definiert:
http://msdn.microsoft.com/de-de/library/system.security.principal.iprincipal.aspx .
Wenn Applikationen nur dieses Interface heranziehen, um Userinformationen abzufragen, dann sind sie unabhängig von der tatsächlich vorliegenden Art der Authentifizierung und Autorisierung. So kann beispielsweise die gleiche Webapplikation über Windows Integrated Security für den Intranet Bereich betrieben werden und mit dem PvpHttpModule für PVP Zugriff.

In der Webapplikation können Sie über den HttpContext auf den User zugreifen. Also beispielsweise

```
HttpContext.Current.User
```

oder auf einer aspx Seite einfach mit

```
Context.User
```

Bei aktiviertem HttpPvpModule erhalten Sie dabei ein Objekt das IPvpPrincipal implementiert, eine Erweiterung von IPrincipal.

4.5.2 IPvpPrincipal

4.5.2.1 Authentifizierung

Wie bei IPrincipal können Sie die Authentifizierung des Users abfragen: mit Identitiy kommen Sie zu einem Objekt, das IIdentity implementiert. Hier gibt es die Informationen:

IsAuthenticated: ist true, wenn die Header X-AUTHENTICATE-UserID, X-Version, X-AUTHENTICATE-gvOuId, X-AUTHENTICATE-Ou gesetzt sind, das sind die verpflichtenden Header bei PVP.

AuthenticationType: ist " PVP Version " und der Wert des Headers X-Version, bzw. null, wenn der Header nicht gesetzt ist.

Name: ist der Wert des Headers X-AUTHENTICATE-UserID.

4.5.2.2 Autorisierung

Mit
bool IsInRole(string roleName)
können Sie Abfragen, ob der Benutzer eine bestimmte Rolle hat. Die bei PVP möglichen Parameter von Rollen werden hierbei nicht berücksichtigt. Falls Sie das wünschen, kommt die Erweiterung gegenüber dem Interface IPrincipal zum Tragen: verwenden Sie die Methode
bool IsInRole(string roleName, NameValueCollection roleParameters)
um Rollenparameter zu berücksichtigen.

5 Anhang

5.1 Abbildungsverzeichnis