

WKO Inhouse GmbH
der Wirtschaftskammern Österreichs
Bereich WKO-IT
FB-Nr. 218433a | Handelsgericht Wien
Wiedner Hauptstraße 76 | 1040 Wien
T + 43 (0)5 90 900 | F + 43 (0)5 90 900-4799
E wko.it@inhouse.wko.at

Einbindungsdokumentation - Anhang Claims WKIS 2

Erstellt von / am: Dominik Amon / 15.9.2014
Dokument / Ablageort: https://projekte.oe.wknet/WKIS/Projektdokumente/Dokumentation_Einbindungsdokumentation-AnhangClaims.docx

INHALT

1	ALLGEMEINES	5
2	CLAIMS	5
2.1	Wichtige Claims im Überblick.....	6
2.1.1	Redirect-Claim	6
2.1.2	MyWKIS Url Claim.....	6
2.1.3	RoleSelection Url Claim	6
2.1.4	PersonID.....	7
2.1.5	FormerPersonID-Claims.....	7
2.1.6	SamAccountName / WindowsAccountName	7
2.1.7	UPN	7
2.1.8	Name-Claim (Anzeigename)	8
2.1.9	PersonenInformation-Claims	8
2.1.10	Email.....	8
2.1.11	Role-Claim (Berechtigungsinformationen)	8
2.1.12	RoleID.....	9
2.1.13	RoleDescription	9
2.1.14	RoleExtendedDescription	9
2.1.15	RoleTypeID.....	9
2.1.16	RoleTypeDescription	9
2.1.17	RoleRelationTypeID	10
2.1.18	RoleOwner-Claims	10
2.1.19	PossibleRole-Claim	11
2.1.20	PossibleRole-Claim (Erweiterung)	11
2.1.21	RollenID	11
2.1.22	RollenTypeID.....	11
2.1.23	KammerMitgliedsnummer	12
2.2	Technische Behandlung des Redirect-Claims	13
2.2.1	Vorgehen bei jedem Seitenaufruf	13
2.2.2	Zwischengespeicherte Aufrufseite	14
2.3	Rollenwechsel.....	14
2.3.1	Technische Behandlung des Rollenwechsels (ohne Inhouse Framework)	14
2.3.2	Technische Behandlung des Rollenwechsels (mit Inhouse Framework)	15
2.4	Link zum Benutzerprofil	15
2.4.1	Technische Behandlung des Benutzerprofillinks (ohne Inhouse Framework)	15
2.4.2	Technische Behandlung des Benutzerprofillinks (mit Inhouse Framework)	16
3	AUDITING UND PROTOKOLLIERUNG VON BENUTZERN	17

4	LOGIN	18
4.1	Globales Anmelden - SAML Protokoll	18
4.2	Globales Anmelden - WSFederation	18
5	LOGOUT.....	20
5.1	Globales Abmelden - SAML	20
5.2	Lokales Abmelden - WSFederation	20
5.2.1	Visual Basic	20
5.2.2	C#	20
5.3	Globales Abmelden - WSFederation	21
5.3.1	Visual Basic	21
5.3.2	C#	21
6	CODESNIPPETS.....	21
6.1	Auslesen von Claims	21
6.1.1	Logik zum Ermitteln, ob User angemeldet ist	22
6.1.2	Auslesen bestimmter Claims	22
7	ERSTELLEN DER FEDERATIONMETADATEN	22
8	GLOSSAR.....	27

DOKUMENTENHISTORIE

Datum	Autor	Version	Änderung	Begründung
10.09.2014	Amon	0.1	Entwurf	
15.09.2014	Amon	0.7	Erstversion & Korrektur	
17.04.2015	Houszka	0.8	Erweiterung	Zwei überschneidende Dokumentationen vorhanden (Zusammenführung in ein Dokument)
02.12.2015	Houszka	0.15	Kapitel „CodeSnippets“ hinzugefügt	
17.02.2016	Houszka	0.16	Kapitel umgeordnet & erweitert	

1 ALLGEMEINES

Dieses Dokument beschreibt die notwendigen Schritte für die Integration von WKIS aus Applikationssicht und beschreibt die einzelnen Aufgaben der zurückgelieferten Claims bzw. deren Verarbeitung.

Dieses Dokument versteht sich als Begleitdokument zu „Dokumentation_AdfsLogin“.

2 CLAIMS

Im Zuge des Login-Prozesses werden der Client-Applikation diverse Claims (auch SAML-Assertions / Attributes genannt) in einem Claim-Token ausgestellt.

Claims sind mit typisierten Key-Value Einträgen, der Claim-Token ist mit einer Key-Value Liste vergleichbar.

Die Ausgabe der verschiedenen Claims ist dabei individuell für die jeweilige Applikation abgestimmt und erfolgt nach dem Least-Privilege Prinzip. Dies bedeutet, es werden einer Applikation tatsächlich nur die Informationen übergeben, die sie für die korrekte Ausführung benötigt, aber auch nicht mehr.

Hintergründe hierfür sind neben Performancegründen, Größe des Tokens (welcher in einem Cookie zwischengespeichert wird) vor allem Sicherheitsbedenken.

Aufbau eines Claim Tokens in SAML (Saml Token)

```
<t:RequestSecurityTokenResponse xmlns:t="http://schemas.xmlsoap.org/ws/2005/02/trust">
...
  <t:RequestedSecurityToken>
    <saml:Assertion MajorVersion="1" MinorVersion="1" AssertionID="..." Issuer="..."
      IssueInstant="2014-09-05T07:41:04.230Z"
      xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
      ...
      <saml:AttributeStatement>
        ...
        <saml:Attribute AttributeName="ATTRIBUT_NAME"
          AttributeNamespace="ATTRIBUT_NAMESPACE">
          <saml:AttributeValue>ATTRIBUT_WERT</saml:AttributeValue>
        </saml:Attribute>
      </saml:AttributeStatement>
    </saml:Assertion>
  </t:RequestedSecurityToken>
</t:RequestSecurityTokenResponse>
```

Der zusammengesetzte vollqualifizierte Name eines Claims ist dabei ATTRIBUT_NAMESPACE und ATTRIBUT_NAME.

Beispiel:

ATTRIBUT_NAMESPACE: <http://schemas.wko.at/ws/2014/02/identity/claims>

ATTRIBUT_NAME: roletypeid

Ergibt zusammengesetzt den vollqualifizierten Namen:

<http://schemas.wko.at/ws/2014/02/identity/claims/roletypeid>

Der Attributwert wird dabei immer als String zurückgeliefert.



Allgemeiner Hinweis

Die Keys der Claims sind URIs. In Verwendung ist hier das URN (wird bevorzugt verwendet vom Bundesrechenzentrum im Portalverbundprotokoll) und URL Format (von Wirtschaftskammer, Microsoft und XmlSoap.org bevorzugt).

2.1 Wichtige Claims im Überblick

Die Datenquellen und Daten selbst für Claims können unterschiedlich sein. Je nach Anforderung kommen die Daten meist jedoch aus den Datenquellen „WKOBASE“, „WKIS-DB“ und „ActiveDirectory“.

Die meisten Claims sind jedoch auf folgenden Bereichen verteilt:

- Authentifizierungsdaten wie Benutzername(n) (aus ActiveDirectory)
- Personendaten (aus WKIS-DB oder WKOBASE)
- Rollendaten¹ (aus WKIS-DB und/oder WKOBASE)
- Rollenspezifische Daten (aus WKOBASE)
- Autorisierungsdaten (aus WKIS-DB oder WKOBASE)

2.1.1 Redirect-Claim

Vollqualifizierter Name

<http://schemas.wko.at/ws/2014/02/identity/claims/redirect>

Hierbei handelt es sich um eine http(s)-Adresse.

Der Redirect-Claim wird in folgenden Situationen ausgestellt und beinhaltet je nach Situation unterschiedliche Werte:

- Der Benutzer muss seine Stammdaten ergänzen
- Der Login erfolgte über eine nicht zugeordnete Bürgerkarte
- Zur Rollenauswahl - diese geschieht entweder implizit (wenn der Benutzer nur eine Rolle besitzt) oder explizit (der Benutzer wählt manuell eine von mehreren Rollen aus)

Wird ein Redirect-Claim ausgestellt, so muss die Anwendung darauf reagieren.

Details siehe Kapitel 2.2, Technische Behandlung des Redirect-Claims.

Beispielwert: <https://tae.dev.oe.wknet/MyWkisFrontend/RoleSelection/RoleSelection.aspx>

2.1.2 MyWKIS Url Claim

Vollqualifizierter Name

<http://schemas.wko.at/ws/2014/06/identity/claims/mywkisurl>

Dieser Claim enthält die https-Adresse zum Root-Verzeichnis der MyWkis-Applikation inklusive abschließendem Schrägstrich (/).

Anhand dieser Informationen kann ein Verweis auf einen Link zum Editieren der Profilinformationen erfolgen.

Beispielwert: <https://tae.dev.oe.wknet/MyWkisFrontend/>

2.1.3 RoleSelection Url Claim

Vollqualifizierter Name

<http://schemas.wko.at/ws/2014/02/identity/claims/roleselectionurl>

Dieser Claim enthält die https-Adresse zur Rollenauswahl.

Anhand dieser Informationen kann ein Verweis für den Rollenwechsel innerhalb einer Applikation erfolgen.

Beispielwert: <https://wkis.dev.wko.at/RoleSelection/RoleSelection.aspx>

¹ Der Begriff „Rolle“ ist mehrfach belegt und kann daher schnell zu Verwechslungen führen. Grundsätzlich ist zwischen zwei unterschiedlichen Arten von Rollen zu unterscheiden: Rollen welche für Berechtigungen in Applikationen gelten, wie „Administrator“ oder „Lesebenutzer“, die andere Art von Rollen sind solche die für Personen gelten, wie zum Beispiel: „Mitarbeiter“ oder „WK-Mitglied“.

2.1.4 PersonID

Vollqualifizierter Name

<http://schemas.wko.at/ws/2011/12/identity/claims/personid>

Hierbei handelt es sich um eine GUID (Format: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX) welche eine Person in der WKIS-Datenbank eindeutig identifiziert.

2.1.5 FormerPersonID-Claims

Vollqualifizierter Name

<http://schemas.wko.at/ws/2014/05/identity/claims/formerpersonid>

<http://schemas.wko.at/ws/2014/05/identity/claims/formerpendingpersonid>

Hierbei handelt es sich um eine GUID (Format: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX).

Durch das Zusammenführen von Personen (FormerPersonID) oder durch das Fertigstellen einer Registrierung (FormerPendingPersonID), können sich PersonIDs ändern bzw. gelöscht werden.

Mit diesem Claim können alle (dies kann auch eine Liste sein!) bisherigen PersonIDs die eine Person hatte, ausgewertet werden. Werden auch als „FormerRoleOwner-Claims“ bezeichnet.

2.1.6 SamAccountName / WindowsAccountName

Vollqualifizierter Name

<http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname>

Der SamAccountName oder im Claim „WindowsAccountName“ ist der Benutzername einer Person. Dieser ist mit weniger als 20 Zeichen begrenzt.

Jeder Account besitzt auch einen SamAccountName, da der SamAccountName intern ein Pflichtfeld ist. Wenn ein Benutzer keinen wählt, wird ein dynamischer Benutzername generiert, welcher an den Oberflächen jedoch nicht angezeigt wird.

Der SamAccountName wird intern verwendet für die Protokollierung von Datenänderungen.



Wichtiger Hinweis

Ein Benutzer kann den SamAccountName jederzeit ändern. Diese Änderungen werden in unseren Systemen intern protokolliert, können aber nicht über eine Schnittstelle abgefragt werden. Der SamAccountName ist daher *nicht geeignet* zur eindeutigen Identifizierung einer Person. Es empfiehlt sich hierbei die PersonID zu verwenden, da hier auch bei Bedarf vorherige PersonIDs mitgeliefert werden können.

2.1.7 UPN

Vollqualifizierter Name

<http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn>

Der UPN (UserPrincipalName) ist neben dem SamAccountName ein Benutzername, die eine Person verwenden kann, um in ein System einzusteigen. Der UPN ist im Emailadress-Format.

Der UPN ist für eine Person nicht verpflichtend festzulegen, wird ein UPN angefordert wird in dem Fall automatisch immer der SamAccountName herangezogen und als Suffix der interne Domainname angefügt.



Wichtiger Hinweis

Ein Benutzer kann wählen, ob seine Emailadresse auch zum Anmelden verwendet werden soll. In diesem Fall wird die Emailadresse in das UPN Feld übertragen. Es handelt sich jedoch trotzdem um getrennte Felder, weswegen es technisch auch möglich ist, einen anderen UPN zu besitzen als die eingetragene Emailadresse. Möchte man die Emailadresse eines Benutzers ermitteln bzw. verwenden, so ist der Email-Claim zu verwenden!

Ein Benutzer kann den UPN jederzeit ändern. Diese Änderungen werden jedoch nicht protokolliert. Der UPN ist daher weder geeignet zur eindeutigen Identifizierung einer Person

noch zur Protokollierung von Änderungen. Es empfiehlt sich hierbei die PersonID zu verwenden, da hier auch bei Bedarf vorherige PersonIDs mitgeliefert werden können.

2.1.8 Name-Claim (Anzeigename)

Vollqualifizierter Name

<http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name>

Ist der Anzeigename eines Benutzers. Bei natürlichen Personen wird hier eine Zusammensetzung von Titel, Vorname, Nachname und Titelnachname geliefert, bei juristischen Personen die Unternehmensbezeichnung.

Dieser Claim wird für die „Angemeldet als ...“ Anzeige verwendet.

Bei Verwendung des Inhouse-Frameworks:

- `WkisClientModule.GetInstance().DisplayName`

2.1.9 PersonenInformation-Claims

Vollqualifizierter Name

<http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname>

<http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname>

<http://schemas.xmlsoap.org/ws/2005/05/identity/claims/gender>

<http://schemas.xmlsoap.org/ws/2005/05/identity/claims/dateofbirth>

<http://schemas.wko.at/ws/2013/11/identity/claims/title>

<http://schemas.wko.at/ws/2013/11/identity/claims/postgraduatetitle>

Personendaten können auch aufgeteilt in Vorname, Nachname etc. als Claims übermittelt werden.

In den meisten Fällen ist dies jedoch nicht notwendig, da mit dem Anzeigename bereits der zusammengesetzte Anzeigename mitgeliefert wird.

PersonenInformationen gibt es nur bei natürlichen Personen. Die Daten können wahlweise aus der WKIS Datenbank oder WKOBASE stammen.

2.1.10 Email

Vollqualifizierter Name

<http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress>

Beinhaltet die Emailadresse des Benutzers. Diese kann je nach Datenquelle aus der WKOBASE oder WKIS-Datenbank kommen.

Aus der WKOBASE stammt diese meist aus der RolleKommenDaten-Tabelle vom Typ Email.

Aus der WKIS-DB aus der Person-Tabelle.

Dies kann je nach Anforderung abweichen.

2.1.11 Role-Claim (Berechtigungsinformationen)


Vollqualifizierter Name

<http://schemas.microsoft.com/ws/2008/06/identity/claims/role>

Hierbei handelt es sich um die Berechtigungsrolle wie zum Beispiel „Administrator“ oder „Lesebenutzer“. Der Role Claim wird in vielen Fällen mehrfach ausgestellt, da ein Benutzer mehrere Rollen besitzen kann.

Als Datenquelle kann hier zwischen WKIS-DB und WKOBASE gewählt werden, als Verwaltungsapplikationen sind hier jeweils die Berechtigungsverwaltung oder SMC in Verwendung.

Zur Abfrage von Berechtigungsrollen muss für die jeweilige Applikation jedoch eine URI in den Claim-Rules hinterlegt sein.

 **Wichtiger Hinweis**
Nicht verwechseln mit Personen-Rollen!

2.1.12 RoleID

Vollqualifizierter Name
<http://schemas.wko.at/ws/2011/12/identity/claims/roleid>

Hierbei handelt es sich um eine GUID (Format: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX) welche eine Rolle in der WKIS-Datenbank eindeutig identifiziert.

Die Role-ID entspricht bei vielen Rollen in der WKOBASE der Security.Account_ID.

 **Wichtiger Hinweis**
Nicht verwechseln mit RollenID aus der WKOBASE!

2.1.13 RoleDescription

Vollqualifizierter Name
<http://schemas.wko.at/ws/2014/02/identity/claims/roledescription>

Eine Bezeichnung für die jeweilige Rolle. Je nach Rollentyp kann dies im Falle eines WK Mitarbeiters beispielsweise Vor- & Nachname des Mitarbeiters sein, im Falle von WK Mitgliedern auch die Firmenbezeichnung.

2.1.14 RoleExtendedDescription


Vollqualifizierter Name
<http://schemas.wko.at/ws/2014/02/identity/claims/roleextendeddescription>

Eine erweiterte optionale Beschreibung für die jeweilige Rolle, welche unterschiedlich nach Rollentyp befüllt sein kann (derzeit nur bei WK Mitgliedsrollen). Bei WK Mitgliedern ist hier zum Beispiel „Wirtschaftskammer Mitglied Tirol“.

2.1.15 RoleTypeID

Vollqualifizierter Name
<http://schemas.wko.at/ws/2014/02/identity/claims/roletypeid>

Eine ID vom Typ long, welche die RoleTypeID einer Rolle auf der WKIS-Datenbank zuordnet.

 **Wichtiger Hinweis**
Nicht verwechseln mit RollenTypeID aus der WKOBASE.
RoleTypeID und RollenTypeID sind unterschiedlich.

2.1.16 RoleTypeDescription

Vollqualifizierter Name
<http://schemas.wko.at/ws/2014/02/identity/claims/roletypedescription>

Bezeichnung des Rollentyps aus der WKIS-DB.

2.1.17 RoleRelationTypeID

Vollqualifizierter Name

<http://schemas.wko.at/ws/2014/02/identity/claims/rolerelationtypeid>

Gibt an in welcher Beziehung eine Rolle zu einer Person steht. Folgende long-Werte sind möglich:

- 1 = Original
Die Person besitzt diese Rolle.
- 2 = Impersonate
Die Person übt diese Rolle über einen administrativen Zugang aus
- 3 = Delegation
Die Person hat diese Rolle delegiert bekommen.
- 4 = WeakDelegation
Die Person hat eine sogenannte „schwache Delegation“ geholt.

2.1.18 RoleOwner-Claims

Vollqualifizierter Name

<http://schemas.wko.at/ws/2014/02/identity/claims/roleownerpersonid>

<http://schemas.wko.at/ws/2014/02/identity/claims/roleownerpersonname>

<http://schemas.wko.at/ws/2014/02/identity/claims/roleownerpersongivenname>

<http://schemas.wko.at/ws/2014/02/identity/claims/roleownerpersonsurname>

<http://schemas.wko.at/ws/2014/02/identity/claims/roleownerpersongender>

<http://schemas.wko.at/ws/2014/02/identity/claims/roleownerpersondateofbirth>

<http://schemas.wko.at/ws/2014/02/identity/claims/roleownerpersonemailaddress>

<http://schemas.wko.at/ws/2014/02/identity/claims/roleownerpersontitle>

<http://schemas.wko.at/ws/2014/02/identity/claims/roleownerpersonpostgraduatetitle>

Die RoleOwner-Claims beinhalten Informationen zur Person, die die Rolle besitzt (in einer „Original“ Beziehung zur Rolle steht).

RoleOwner-Claim	Entsprechung auf der besitzenden Person
RoleOwnerPersonID	PersonID
RoleOwnerPersonName	Name
RoleOwnerPersonGivenName	GivenName
RoleOwnerPersonSurname	Surname
RoleOwnerPersonGender	Gender
RoleOwnerPersonDateOfBirth	DateOfBirth
RoleOwnerPersonEmailAddress	EmailAddress
RoleOwnerPersonTitle	Title
RoleOwnerPostGraduateTitle	PostGraduateTitle

2.1.19 PossibleRole-Claim

Vollqualifizierter Name

<http://schemas.wko.at/ws/2014/06/identity/claims/possiblerole>

In diesem Claim sind Rolleninformationen enthalten, um einen Rollenwechsel innerhalb einer Applikation durchführen zu können.

Anders als die anderen Claims sind hier durch ein Trennzeichen weitere Informationen verschachtelt, um diese entsprechend für eine Auswahlliste anzeigen zu können.

Das Format sieht dabei wie folgt aus:

RoleID|RoleTypeID|RoleRelationTypeID|RoleTypeDescription|RoleDescription

Zum Beispiel:

D7882A96-1DD2-0000-97CC-42613E239DA9|4|1|WK-Mitarbeiter|Dominik Amon, WK Österreich

Details siehe Kapitel 2.3, Rollenwechsel.

2.1.20 PossibleRole-Claim (Erweiterung)

Vollqualifizierter Name

<http://schemas.wko.at/ws/2014/06/identity/claims/possiblerole>

Aufgrund einiger Anfragen gibt es zum PossibleRole-Claim auf 2.1.18 eine Erweiterung um die Kammermitgliedsnummer. Diese wird am Ende mit einem weiteren Trennzeichen (Pipe) angefügt.

Wenn es sich um keine Mitgliedsrolle handelt, so bleibt die „Spalte“ für die Kammermitgliedsnummer leer, der Pipe „|“ davor bleibt jedoch erhalten.

Das Format sieht dabei wie folgt aus:

RoleID|RoleTypeID|RoleRelationTypeID|RoleTypeDescription|RoleDescription|Kammermitgliedsnummer

Zum Beispiel:

4832EB5C-B88D-4682-8E3F-00004136634D|2|1|Wirtschaftskammer Mitglied Wien|Roshbina Real Estate GmbH (0781994)|0781994

D7322A96-1DD2-0000-97CC-436135239DA9|4|1|WK-Mitarbeiter|Dominik Amon, WK Österreich|

Details siehe Kapitel 2.3, Rollenwechsel.

2.1.21 RollenID

Vollqualifizierter Name

<http://schemas.wko.at/ws/2011/12/identity/claims/rollenid>

Die RollenID ist ein long-Wert und identifiziert eine Rolle in der WKOBASE Datenbank.



Wichtiger Hinweis

Nicht verwechseln mit der RoleID aus der WKIS-DB.

2.1.22 RollenTypeID

Vollqualifizierter Name

<http://schemas.wko.at/ws/2011/12/identity/claims/rollentypid>

Die RollenTypeID ist ein long-Wert und identifiziert einen RollenTyp in der WKOBASE Datenbank.



Wichtiger Hinweis

Nicht verwechseln mit RoleTypeID aus der WKIS-DB.
RollenTypeID und RoleTypeID sind unterschiedlich.

2.1.23 KammerMitgliedsnummer

Vollqualifizierter Name

<http://schemas.wko.at/ws/2014/03/identity/claims/kammermitgliedsnummer>

Die Kammermitgliedsnummer ist ein Text-Wert und beinhaltet die siebenstellige Mitgliedsnummer.
Die erste Stelle der Kammermitgliedsnummer ist die Kammernummer.

Der KammerMitgliedsnummer-Claim wird nur bei Wirtschaftskammer Mitgliedern ausgestellt.

Bei sogenannten MultipleAccounts wird auch bei Gold-Zusatz und Silber-Pins die
KammerMitgliedsnummer des Mitglieds zurückgeliefert.

Das Quellsystem ist hierbei immer die WKOBASE.



Wichtiger Hinweis

Die Kammermitgliedsnummer kann führende Nullen enthalten, sie wird daher als Text und
nicht als Zahl zurückgeliefert.

2.2 Technische Behandlung des Redirect-Claims

Wird ein Redirect-Claim ausgestellt, so muss auf diesen zwingend reagiert werden, da andernfalls ein fehlerhaftes Verhalten resultieren kann. Wann ein Redirect-Claim ausgestellt wird, ist in „*Dokumentation_AdfsLogin*“ in Kapitel 14 „*Rollenauswahl*“ beschrieben.

Wird das Inhouse-Framework und in Folge das WkisClientModule verwendet, ist nichts weiter zu machen.

Für alle anderen Fälle ist wie folgt auf den Redirect-Claim zu reagieren.

2.2.1 Vorgehen bei jedem Seitenaufruf

- Ist die aufgerufene Seite öffentlich erreichbar bzw. eine Authentifizierung nicht relevant, muss keine weitere Logik ausgeführt werden.
- Ist eine zwischengespeicherte Aufrufseite (Raw-Url) hinterlegt, muss auf diese weitergeleitet werden und der Aufruf der aktuellen Seite terminiert werden.

Ist ein Redirect-Claim vorhanden:

- Die aktuelle Aufrufseite (Raw-Url) zwischenspeichern.
- Nach Möglichkeit keine Benutzersitzung aufbauen. Geschieht dies jedoch automatisch, dann die aktuelle Sitzung terminieren und ggf. ein lokales Abmelden (siehe Kapitel 6 „CodeSnippets“) erzwingen. Das bedeutet, dass nur die eigenen Authentifizierungscookies und die Session terminiert werden sollen, **nicht** aber ein Abmelden am ADFS erzwungen werden soll.
- Weiterleiten auf die im RedirectClaim hinterlegte Adresse und Anfügen des „wtrealm“-Parameters der jeweiligen Applikation.

Beispiel: Im Redirect Claim ist „https://mywkis.wko.at/Sample/“ enthalten, der WTRrealm einer Beispiel-Applikation lautet „https://mycustomapplication.at/“, so muss auf <https://mywkis.wko.at/Sample/?wtrealm=https%3a%2f%2fmycustomapplication.at%2f> weitergeleitet werden.

SAML2-Beispiel:

```
<saml:Attribute AttributeName="redirect"
  AttributeNamespace="http://schemas.wko.at/ws/2014/02/identity/claims">
  <saml:AttributeValue> https://mywkis.wko.at/Sample/ </saml:AttributeValue>
</saml:Attribute>
```

- Folgende weitere Parameter können noch hinzugefügt werden:
 - `idr=true`: Dabei eine mögliche gesetzte Hauptrolle ignoriert
 - `prtctx=[1-7]`: Dabei kann ein bevorzugter Rollentyp angegeben werden

Beispiel:

<https://mywkis.wko.at/Sample/?wtrealm=https%3a%2f%2fmycustomapplication.at%2f&idr=true&prtctx=2>



Allgemeiner Hinweis

Wenn mehr als eine Rolle des bevorzugten Typs vorliegt, wird eine Rollenwahl ALLER unterstützter Rollentypen angeboten. Die Hauptrolle wird normal berücksichtigt außer sie wird via URL-Parameter (siehe oben) unterdrückt!

Der WTRrealm-Parameter identifiziert eindeutig eine Anwendung im System und ist gleichzeitig die ID der RelyingParty / des ServiceProviders.

**Allgemeiner Hinweis**

Es stehen auch weitere Parameter bei der Weiterleitung zur Verfügung. Zum Beispiel ist es möglich eine Vorauswahl einer bestimmten Rolle zu treffen, unter Verwendung des „RoleID“ Parameters. Entsprechende Absprache mit WKO-IT

- Terminierung der weiteren Ausführung der Seite.
Das bedeutet, dass nach dem Redirect die Logik der Applikation nicht fortgesetzt werden darf.

2.2.2 Zwischengespeicherte Aufrufseite

In dem obigen Kapitel wird darauf hingewiesen, dass auf eine zwischengespeicherte Aufrufseite weitergeleitet werden soll bzw. die aktuelle aufgerufene Seite zwischengespeichert werden soll.

Als Aufrufseite versteht die aktuelle Seite der Anwendung inklusive aller Query-String-Parameter.

Verwendet man das InhouseFramework, so kann man dafür auf das WKISClientModule zurückgreifen. Das WKISClientModule ist so konfiguriert, dass es noch vor Initialisierung eines Session-Objektes ausgeführt wird. Damit einher geht eine technische Einschränkung, welche es notwendig macht, die aktuell aufgerufene Seite in einem Cookie zu speichern. Dieses Cookie hat einen Ablauf von wenigen Minuten und ist HttpOnly und als Secure-Cookie (nur über https) markiert, der Name des Cookies ist „__WKIS_CurrentUrl“. Nach dem Auslesen der zwischengespeicherten Adresse im Cookie nach dem Redirect, wird das Cookie automatisch gelöscht.

Für externe Anbindungen ist der Prozess eigenständig zu implementieren.

Technisch ist ein Cookie jedoch nicht zwingend notwendig, auch andere Zwischenspeicherung ist möglich, beispielsweise in einer Session (Achtung, dieser Wert darf im Zuge des Redirects und einem Beenden der Session dann natürlich nicht gelöscht werden!).

**Wichtiger Hinweis**

Nach einem Redirect zurück zur Anwendung, muss auf die zwischengespeicherte Aufrufseite weitergeleitet werden, und danach entsprechend geleert werden! Andernfalls kann es zu einer Endlosschleife kommen.

**Wichtiger Hinweis**

Mit dem Redirect-Claim (Url aus Claim) kann auch innerhalb der Applikation wieder zur Rollenauswahl geleitet werden, es sollte standardmäßig jedoch der Rollenwechsel (Kapitel 2.3) verwendet werden.

2.3 Rollenwechsel

Die allgemeine Beschreibung zum Rollenwechsel und dessen Ablauf aus Benutzersicht ist in „Dokumentation_AdfsLogin“ in Kapitel 9 „Rollenwechsel innerhalb einer Applikation“ näher beschrieben und wird in diesem Dokument daher nicht weiter erklärt.

2.3.1 Technische Behandlung des Rollenwechsels (ohne Inhouse Framework)

- Zunächst muss die RoleID ermittelt werden, auf die gewechselt werden soll. Die möglichen Rollen können bereits als Claims übertragen werden (siehe Kapitel 2.1.19, PossibleRole-Claim) oder über andere Datenquellen bezogen werden.
- Es ist notwendig die Adresse der Rollenauswahl zu kennen. Diese wird idealerweise ebenfalls über einen Claim übertragen werden (siehe Kapitel [2.1.3, RoleSelection Url Claim](#)), wahlweise kann die Adresse zur Rollenauswahl auch aus einer Konfiguration stammen.
- Beim Aufruf des Rollenwechsels muss die aktuelle lokale Benutzersitzung (siehe Kapitel 6 „CodeSnippets“) beendet werden (die zentrale ADFS Benutzersitzung darf dabei nicht

beendet werden!)

- Zwischenspeichern der aktuell aufgerufenen Adresse (Vergleiche 2.2.2, Zwischengespeicherte Aufrufseite)
- Eine Weiterleitung auf die wie folgt aufgebaute Adresse:
[Wert aus RoleSelectionUrl-Claim] + ?wtrealm=APPLICATION_WTREALM&rctx=ROLEID

Beispiel:

Der WTRrealm einer Beispiel-Applikation lautet „https://mycustomapplication.at/“ die RoleID auf die gewechselt werden soll „D7882A96-1DD2-4F61-97CC-42613E239DA9“, so sieht dies wie folgt aus:

<https://wkis.wko.at/RoleSelection/RoleSelection.aspx?wtrealm=https%3a%2f%2fmycustomapplication.at%2f&rctx=D7882A96-1DD2-4F61-97CC-42613E239DA9>

- Terminierung der weiteren Ausführung der Seite.
Das bedeutet, dass nach dem Redirect die Logik der Applikation nicht fortgesetzt werden darf.
- Nachdem der Rollenwechsel vom Benutzer durchgeführt wurde, wird der Benutzer wieder auf die Root-Seite der Anwendung zurückgeleitet. Es muss daher weitergeleitet werden auf die zwischengespeicherte aufgerufene Adresse zurückverwiesen werden und die zwischengespeicherte Adresse wieder aus dem Speicher gelöscht werden (analog Redirect-Claim).

2.3.2 Technische Behandlung des Rollenwechsels (mit Inhouse Framework)

Das WkisClientModule bietet zwei Methoden hierfür an:

- GetPossibleRoles(currentRoleID) bzw. GetPossibleRoles()
- ChangeRole(roleID)

Mit der Methode GetPossibleRoles können alle möglichen Rollen ermittelt werden, aus welcher die RoleID für ChangeRole ermittelt werden kann.

Durch das Aufrufen von ChangeRole wird der Rollenwechsel automatisch durchgeführt.

Wichtiger Hinweis

Damit der Rollenwechsel automatisch funktioniert, müssen die Claims vom Typ „RoleSelectionUrl“ und „PossibleRole“ geliefert werden.

2.4 Link zum Benutzerprofil

Um die Anforderung „Link zum Benutzerprofil“ umsetzen zu können, muss der Benutzer zunächst angemeldet sein. Zusätzlich muss der MyWKIS Url Claim mitgeliefert werden (siehe Kapitel 2.1.2, MyWKIS Url Claim).

- **Allgemeiner Hinweis**
In MyWkis gibt es keinen Link „Zurück zur Applikation“ oder ähnliches. Ist gewünscht, dass die Anwendung beim Öffnen des Benutzerprofiles nicht verschwindet, so muss der Link in einem neuen Fenster/Tab geöffnet werden.

2.4.1 Technische Behandlung des Benutzerprofillinks (ohne Inhouse Framework)

Der Link zur Profileditieren setzt sich wie folgt zusammen:

[MyWkisUrlClaim] + /DataManagement/

Zum Beispiel:

<https://mywkis.wko.at/DataManagement/>

2.4.2 Technische Behandlung des Benutzerprofillinks (mit Inhouse Framework)

Das WkisClientModule stellt hierfür eine eigene Methode zur Verfügung:

- `GetMyWkisDataManagementUrl()`

Wichtiger Hinweis

Wird die Methode aufgerufen und der Benutzer ist nicht angemeldet, so wird eine `MyWkisUrlNotFoundException` geworfen.

3 AUDITING UND PROTOKOLLIERUNG VON BENUTZERN

Für Auditing bzw. Protokollierung gibt es verschiedene Möglichkeiten dies umzusetzen. Da sich jedoch alle IDs inklusive Benutzername von Benutzern ändern können, sind folgende Hinweise und Empfehlungen zu beachten.

- Die RoleID ist nicht veränderbar und gibt an, in welcher Rolle gearbeitet wird. Allerdings ist es möglich, dass jemand eine Rolle einer anderen Person delegiert hat. Ist dies der Fall, so tritt diese Person mit der gleichen RoleID im System auf.
- Die PersonID ist der Identifier der Person, kann sich aber durch Zusammenführen von Accounts verändern bzw. sogar gelöscht werden.
- Der SamAccountName oder UPN kann jederzeit vom Benutzer geändert oder zusammengeführt werden!

Nicht geeignet für die Identifizierung sind SAMAccount-Name oder UPN.

Die Empfehlung:

Eine eindeutige Identifizierung kann daher nur aus der Kombination von PersonID und RoleID erfolgen.

Zu beachten ist, dass die PersonID, wie bereits beschrieben, durch das Zusammenführen von Personen wegfallen kann. Im Sinne der Nachverfolgung bietet hierfür das WKISCoreService die Methode `GetFormerIDsByPersonID` an (Details siehe `Dokumentation_WkisCoreService`).



Allgemeiner Hinweis

Datenbank intern werden die Benutzernamen protokolliert. Die Änderungen von Benutzernamen können im Vergleich zur PersonID jedoch nicht automatisiert ausgewertet werden und sollen daher speziell bei externen Herstellern nicht verwendet werden.

4 LOGIN

WKIS-Applikationen verfügen über SSO(Single-Sign-On), wodurch der Benutzer über mehrere (WKIS-)Applikationen hinweg angemeldet bleibt.

4.1 Globales Anmelden - SAML Protokoll

Weitere Details zum SAML2-Proctol-Login für SingleLogin:

<https://docs.microsoft.com/en-us/azure/active-directory/develop/active-directory-single-sign-on-protocol-reference>

Beispiel eines Request gegen WKIS-QSS:

```
<?xml version="1.0" encoding="UTF-8"?>
<samlp:AuthnRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:protocol"
  ID="GENERATEDID" Version="2.0"
  IssueInstant="2017-02-16T14:43:14Z"
  ForceAuthn="0"
  IsPassive="0"
  ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
  AssertionConsumerServiceURL="LOGINURL">
  <saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">IDENTIFIER der RP</saml:Issuer>
  <samlp:NameIDPolicy Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient" AllowCreate="true"/>
</samlp:AuthnRequest>
```

4.2 Globales Anmelden - WSFederation

Arbeitet man mit <location>-Tags in der web.config, so wird der Benutzer automatisch zur konfigurierten Loginseite weitergeleitet um sich anzumelden.

Beispiel:

```
<location path="Administration">
  <system.web>
    <authorization>
      <allow roles="SuperAdmin"/>
      <deny users="*" />
    </authorization>
  </system.web>
</location>

<system.web>
  <authorization>
    <deny users="?" />
  </authorization>
</system.web>
```

Ist das nicht möglich, muss der Link manuell zusammengebaut werden. Ein Beispiellink auf die Loginseite des QSS-Systems sieht folgendermaßen aus:

<https://wkis.qss.wko.at/adfs/ls/?wa=wsignin1.0&wtrealm=https://localhost:44300/&wctx=rm=0&id=passive&ru=%2fdefault%2f&wct=2015-11-17T07:06:27Z>

- wtrealm: <https://localhost:44300/> (Identifiziert, der in den FederationMetadaten als „entityID“ eingetragen ist)
- ru: mögliche return url, wenn sie das Login zB von <https://localhost:44300/default/> starten dann ist der ru-Wert „%2fdefault%2f“
- wct: Timestamp, mit derzeitigem Zeitstempel

Genaue Informationen, was die einzelnen Parameter sind, finden sie hier:
<https://msdn.microsoft.com/en-us/library/ff359114.aspx>

5 LOGOUT

Analog zum Single-Sign-On (SSO) gibt es auch einen Single-Logout-Prozess (SLO). Analog zum Login werden sowohl WS-* als auch SAML2-Protocol Logout unterstützt.

Wichtig ist bei einer eigenen Implementierung, dass neben dem Abmelden aus der eigenen Applikation auch ein globales standard-konformes Logout erfolgen muss.

5.1 Globales Abmelden - SAML

Weitere Details zum SAML2-Protocol-Logout für SingleLogout:

<http://msdn.microsoft.com/en-us/library/azure/dn195588.aspx>

Beispiel eines Request gegen WKIS-QSS:

```
<?xml version="1.0" encoding="UTF-8"?>
<samlp:LogoutRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
                    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
                    ID="GENERATEDID" Version="2.0"
                    IssueInstant="2017-02-15T09:26:40Z"
                    Destination="https://wkis.qss.wko.at/adfs/ls/" >
  <saml:Issuer>IDENTIFIER DER RP</saml:Issuer>
  <saml:NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient">GENERATED NAMEID FROM
RP</saml:NameID>
  <samlp:SessionIndex>GENERATEDID</samlp:SessionIndex>
</samlp:LogoutRequest>
```

5.2 Lokales Abmelden - WSFederation

Das Lokale Abmelden wird vor dem Redirect auf den Wert im Redirect-Claim ausgeführt. Es wird benötigt, damit die Claims neu ausgestellt werden.

5.2.1 Visual Basic

```
Dim [module] As WSFederationAuthenticationModule =
TryCast(HttpContext.Current.ApplicationInstance.Modules("WSFederationAuthenticationModule"),
WSFederationAuthenticationModule)

[module].SignOut("~/Portal.aspx", False)
```

5.2.2 C#

```
/// <summary>
/// Meldet den Benutzer Lokal ab, NICHT aber von ADFS (dh. SSO ist weiterhin möglich!)
/// </summary>
public static void LocalSignOff()
{
    //Authentifizierungsmodul laden und lokales Abmelden erzwingen (Damit der AD FS2.0 den Claim
    //Token neu ausstellt)
    WSFederationAuthenticationModule module = HttpContext.Current.ApplicationInstance.Modules["W
SFederationAuthenticationModule"] as WSFederationAuthenticationModule;
    module.SignOut(true);
}
```

5.3 Globales Abmelden - WSFederation

Das Globale Abmelden wird für das Abmelden über alle Applikationen verwendet.

5.3.1 Visual Basic

```
Dim authModule = FederatedAuthentication.WSFederationAuthenticationModule

Dim signoutUrl As String =
    (WSFederationAuthenticationModule.GetFederationPassiveSignOutUrl(authModule.Issuer,
    authModule.Realm, Nothing))

WSFederationAuthenticationModule.FederatedSignOut(New Uri(authModule.Issuer), New
Uri(authModule.Realm + "Portal.aspx"))
```

5.3.2 C#

Bei Verwendung des Inhouse-Frameworks (nur intern):

```
WkisClientModule.GetInstance().SignOut();
```

Bei Verwendung von Microsoft WIF:

```
// .NET 4.0
private void SignOff()
{
    Session.Abandon();
    Microsoft.IdentityModel.Web.WSFederationAuthenticationModule.FederatedSignOut(null, new Uri(
    loginPageUri));
}

// .NET 4.5
private void SignOff()
{
    Session.Abandon();
    System.IdentityModel.Services.WSFederationAuthenticationModule.FederatedSignOut(null, new Ur
    i(loginPageUri));
}
```

Achtung: ab .NET4.5 darf der erste Parameter der Methode „FederatedSignOut“ nicht mehr NULL sein:

<https://msdn.microsoft.com/en-us/library/system.identitymodel.services.wsfederationauthenticationmodule.federatedsignout%28v=vs.110%29.aspx>

6 CODESNIPPETS

In diesem Kapitel werden einige CodeSnippets angeboten, die bei der Einbindung verwendet werden.

6.1 Auslesen von Claims

Claims werden nach erfolgreicher Anmeldung vom ADFS ausgestellt und in einem SAML-Token übermittelt. Sie haben folgendes Format:

```
<saml:Attribute AttributeName="redirect"
    AttributeNamespace="http://schemas.wko.at/ws/2014/02/identity/claims">
    <saml:AttributeValue> https://mywkis.wko.at/Sample/</saml:AttributeValue>
</saml:Attribute>
```

Weiters gibt es Unterschiede zwischen WIF 3.5 und WIF 4.5, welche auch hier kurz behandelt werden. Als Hilfe soll auch dieser Artikel dienen:

<http://nzpcmad.blogspot.co.at/2013/07/wif-migrate-from-wif-35-to-wif-45-and.html>

6.1.1 Logik zum Ermitteln, ob User angemeldet ist

```

/// <summary>
/// Logik zum Ermitteln, ob ein Benutzer angemeldet ist
/// </summary>
/// <param name="claimsPrincipal">Ermittelt zu einem ClaimPrincipal ob ein Benutzer angemeldet ist</param>
/// <returns>true, wenn ein Benutzer angemeldet ist</returns>
protected virtual bool IsAuthenticated(IClaimsPrincipal claimsPrincipal)
{
    return (claimsPrincipal != null
        && claimsPrincipal.Identities != null
        && claimsPrincipal.Identities.Count > 0
        && claimsPrincipal.Identity.IsAuthenticated)
        && claimsPrincipal.Identity is ClaimsIdentity
        && ((ClaimsIdentity)claimsPrincipal.Identity) != null
        && ((ClaimsIdentity)claimsPrincipal.Identity).Claims != null
        && String.IsNullOrEmpty(((ClaimsIdentity)claimsPrincipal.Identity).Claims.GetFirstClaimValue(
            http://schemas.wko.at/ws/2011/12/identity/claims/personid))
    )
}

```

6.1.2 Auslesen bestimmter Claims

- .NET 4.0

```

using Microsoft.IdentityModel.Claims;

IClaimsPrincipal claimsPrincipal = HttpContext.Current.User as IClaimsPrincipal;
IClaimsIdentity claimIdentity = claimsPrincipal.Identity as IClaimsIdentity;
string test
= claimIdentity.Claims.GetFirstClaimValue("http://schemas.wko.at/ws/2014/12/identity/claims/pvp/givenname");

```

- .NET 4.5

```

using System.Security.Claims;

ClaimsPrincipal claimsPrincipal = this.Page.User as ClaimsPrincipal;
ClaimsIdentity claimsIdentity = (ClaimsIdentity)claimsPrincipal.Identity;

string test
= claimsIdentity.Claims.GetFirstClaimValue("http://schemas.wko.at/ws/2014/12/identity/claims/pvp/givenname");

```

7 ERSTELLEN DER FEDERATIONMETADATEN

Zum Einrichten der RelyingParty am ADFS wird eine FederationMetadata.xml benötigt, über welche die RP konfiguriert werden kann.

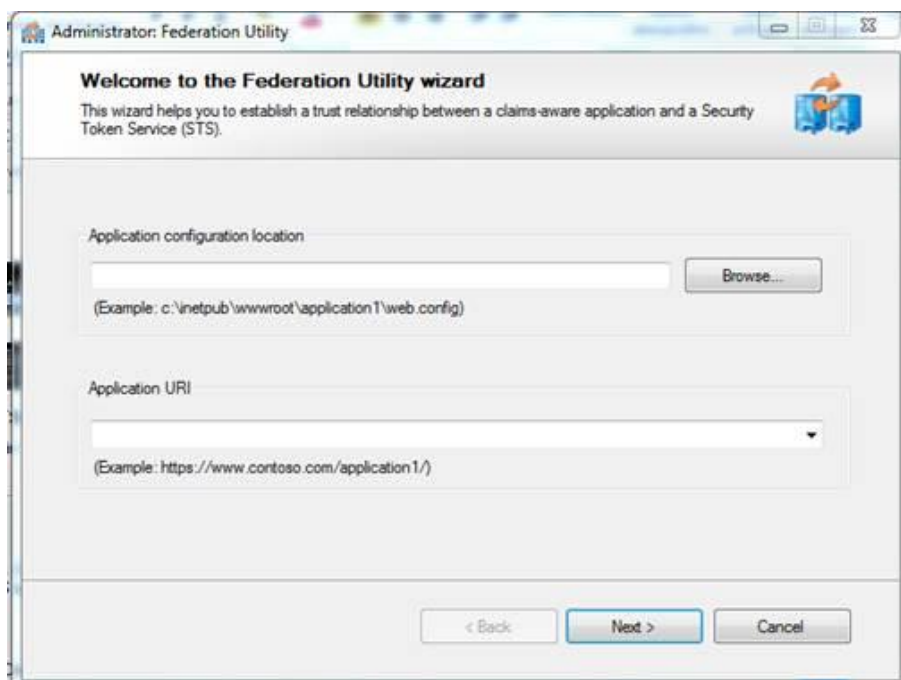
Bis .NET 4.0 können die Metadaten über die FedUtil.exe erstellt werden. Ab .NET 4.5 steht die FedUtil.exe nicht mehr zur Verfügung. Hier müssen die einzelnen Konfigurationsparameter zur Einbindung auf andere Art bekannt gegeben werden.

Wie das FederationMetadata.xml-File generiert werden kann, wird in Folge beschrieben:

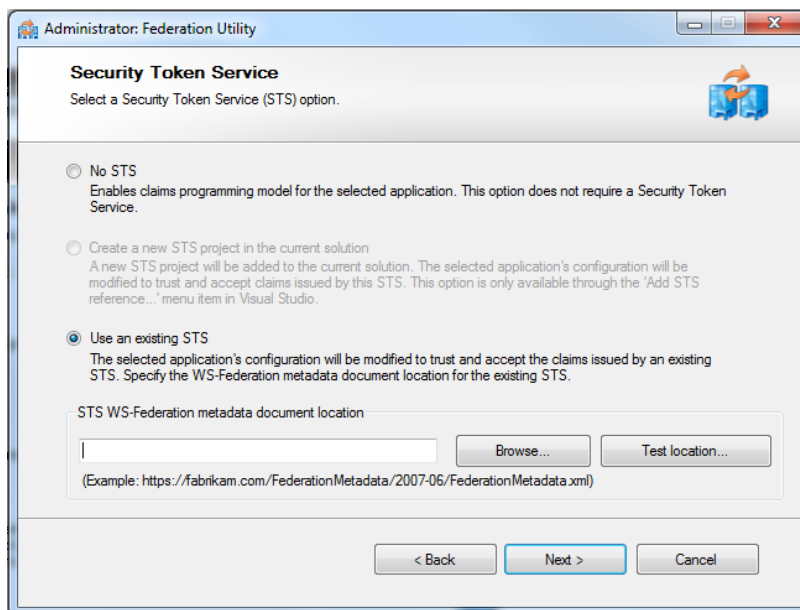
Schritt 1 - FedUtil.exe ausführen:

```
C:\Program Files (x86)\Windows Identity Foundation SDK\v4.0\FedUtil.exe
```

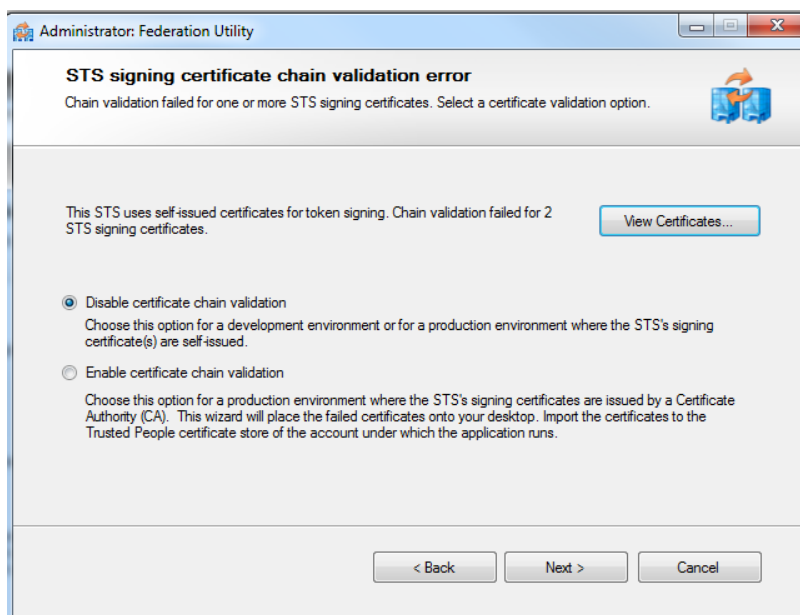
Schritt 2 - web.config & ApplicationURI (muss https sein) ihrer Applikation angeben:



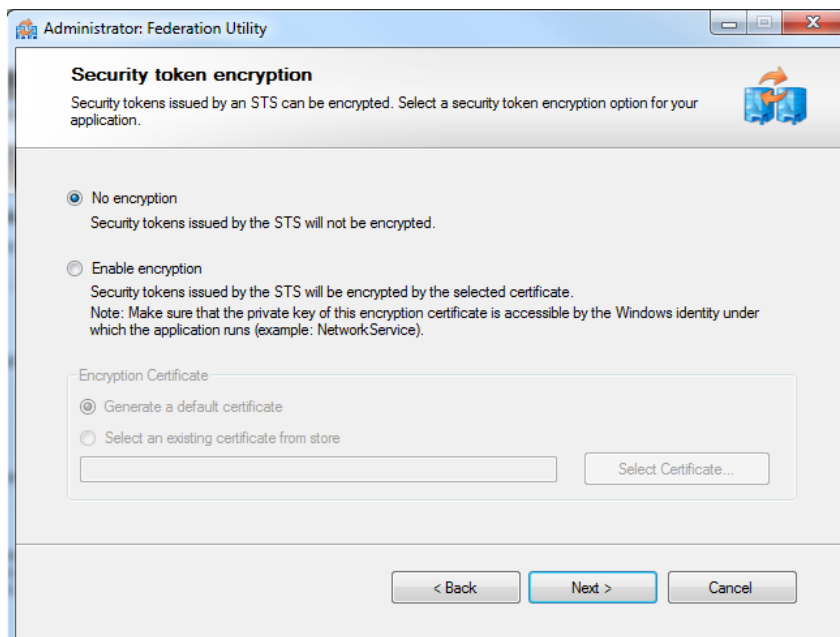
Schritt 3 - FederationMetadata.xml auswählen (wird im Anhang geschickt). Das sind die Metadaten unseres QSS ADFS, der für Sie das STS (Secure Token Service) darstellt:



Schritt 4 - Disable Certificate Chain Validation:



Schritt 5 - No Encryption. Es wird keine zusätzliche Verschlüsselung der Metadaten benötigt, da die Applikation über https (verpflichtend!) läuft und damit geschützt ist.

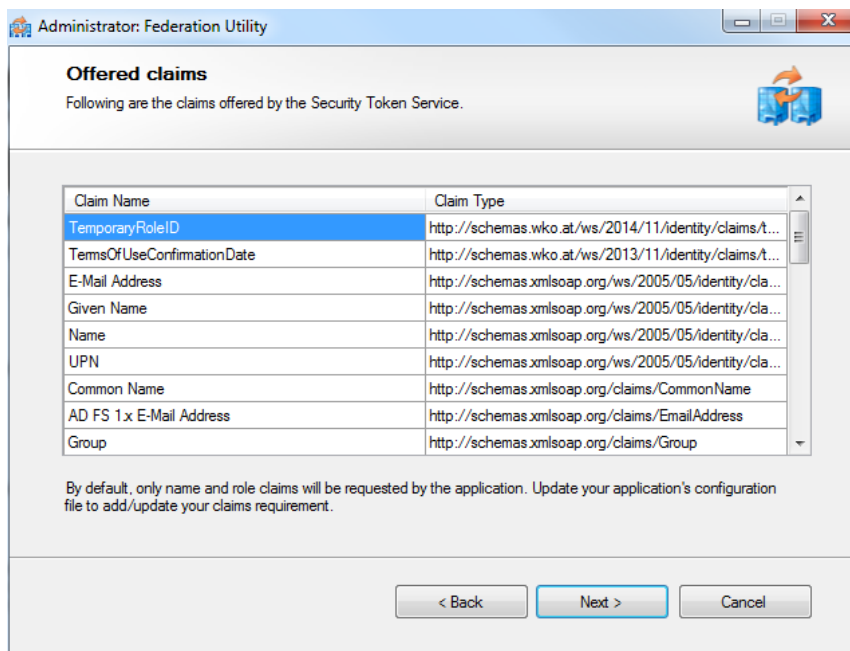


Schritt 6 - Claims auswählen. Es wird eine Liste an Claims angezeigt, die von unserem QSS ADFS bezogen werden. Wenn Sie sich nicht sicher sind, welche sie benötigen oder wenn sie welche vergessen ist das kein Problem, diese können im Nachhinein individuell abgeändert werden, ohne dass weitere Metadaten ausgetauscht werden müssen.

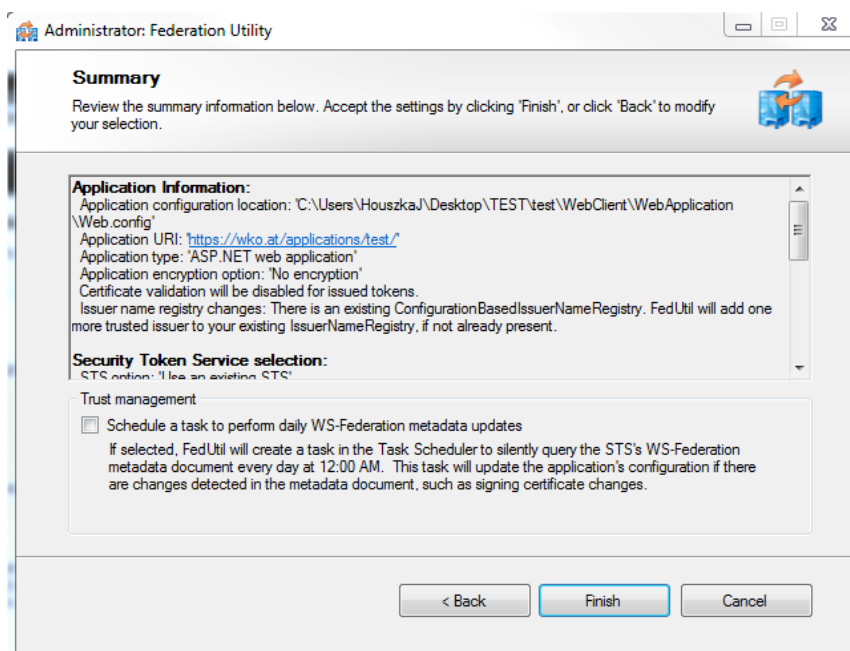
Per Default werden nur Name und Role-Claim über die Metadaten angefragt. Dies ist in der Config zu sehen:

```
<applicationService>
  <claimTypeRequired>
    <claimType type="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name" optional="true"/>
    <claimType type="http://schemas.microsoft.com/ws/2008/06/identity/claims/role" optional="true"/>
  </claimTypeRequired>
</applicationService>
```

Hier könnten Sie noch welche hinzufügen (anhand untenstehender Liste), ist aber wie gesagt nicht zwingend und kann im Nachhinein geändert werden.



Schritt 7 - Fertigstellen:



Nach diesem Vorgang wird die web.config automatisch aktualisiert und es wird eine Metadaten.xml generiert, die alle Informationen beinhaltet.

Diese Metadaten.xml schicken Sie an den Ansprechpartner der WKO-IT. Damit wird dann die RelyingParty angelegt.

Wichtig dabei ist, dass die Endpoints in den Metadaten enthalten sind. Diese sollten aber über die FedUtil.exe automatisch übernommen werden (aus dem config.File).

Auszug aus Dokumentation_EinbindungWebApplikationWKIS:

Welche Adresse soll verwendet werden, um den SAML-Token zu posten.

Bei WS-Federation Protokoll:

- Url als URI für „POST“

Bei SAML2.0 Protokoll:

- SAML Assertion Consumer:
 - Bindungsart: Artifact | POST | Redirect
 - Index: (Zahl des verwendeten Index, Standard ist 0)
 - Url als URI
- SAML Logout:
 - Bindungsart: POST | Redirect
 - Url als URI

Response Url als URI

8 GLOSSAR

Begriff	Beschreibung
WKIS	Wirtschaftskammer Identity System
ADFS	Active Directory Federation Services
AD	Active Directory
DB	Datenbank
WkisCoreService	Zentrales Service, welches bestimmte Methoden zur Registrierung zur Verfügung stellt.
RP	RelyingParty oder auch SP (Service Provider) genannt ist die Applikation, welche mit WKIS (als IDP/STS) geschützt ist.
SAML	Security Assertion Markup Language http://de.wikipedia.org/wiki/Security_Assertion_Markup_Language