**Abstract**

In this document you will find a quick installation and configuration guide for the openRBAC software. This software is implemented in PHP5 and is available under the LGPL (Lesser Gnu Public License). As backend database it uses one or more LDAP servers. The configuration of these servers is not part of this guide.

# Contents

# 1 Introduction

This software is an implementation of the RBAC Standard[ANS04]. This standard defines the minimum of functionality for role based access control systems. If you are interested in details about RBAC or the RBAC Standard, please read the accompanying openRBAC documentation.

# 2 Installation

## 2.1 Requirements

Because this software is written in PHP5 you need an PHP5 interpreter. It has to be compiled with XSL and XML support. If you plan to use RBAC in combination with SOAP services, you have to add SOAP support as well, but you do *not* need to install PEAR to run openRBAC on your PHP5 installation. Further you need one or more LDAP servers that are ready to accept connections from openRBAC which will read and store its data there. If you don not have an LDAP server yet, consider to use openLDAP. You can download it from the project page[1] and you will also find an installation guide there. If you want to use openRBAC with Active Directory as data storage, you have to add LDAPS support to the underlying LDAP class by yourself, which offers only TLS support at the moment.

## 2.2 Platform

The software was developed and tested on openSuSE Linux 10.1. So if you use a Linux on your machine, you shouldn't have any problems. It has not been tested on other platforms, but if you have a PHP5 interpreter running properly on your system, it should work as well.

## 2.3 Getting the code

This project is rather young, so at the moment you can only download the code via a SVN client. The software consists of two parts that are required for openRBAC to run:

**mwlib** This part is a collection of rather small classes that provide general functionality, as for example parsing and reading XML files, or encapsulate LDAP commands. The openRBAC software only needs the following files:

- iNode.interface.php
- iXML.interface.php
- iLDAP.interface.php
- iHelper.interface.php
- Node.class.php
- XML.class.php
- LDAP.class.php

[1] http://www.openldap.org

- Helper.class.php

All other files could be deleted or simply left where they are, because they do no harm as long as you do not include them.

**openRBAC** This part is the RBAC implementation. It is split into multiple files, each containing one class or interface.

You have to download both parts. To make it easier to get updates and patches via SVN, it is recommended that you make an SVN checkout rather then an SVN export. But if you do so, make sure you download *mwlib* and *openRBAC* into two separate directories. Otherwise you will get an error from your SVN client. If you use the Linux subversion client, you can go to your target directory and run the following commands (don't forget the single dot at the end of line 4 and 6!):

```
mkdir rbac
mkdir lib
cd rbac
svn checkout svn://markus-widmer.de/openRBAC/trunk/ .
cd ../lib
svn checkout svn://markus-widmer.de/mwlib/trunk/ .
```

Now you have installed openRBAC on your system. Continue with section 2.6 to learn how to configure the software or read more about the different possibilities using the software in section 2.4.

## 2.4 Make a decision

Before you start to configure openRBAC, you should think about your requirements. You can use all of the software or only parts of it. The differences are described below.

### 2.4.1 The RBAC Framework

You can decide whether you want to use the RBAC Framework or not. It is wrapped around the RBAC libraries and allows you not only to include the required classes from the *mwlib* via a configuration file, but also to use the SSD and/or DSD extension for openRBAC or to write custom extensions. An extension allows you to modify the behaviour of the software in a way it is needed for your application. RBAC, for example, does not know any public resources. So you would have to work around this by implementing the "is public" statement directly into your application. Using an extension for the *checkAccess* function is the other way. You can write your own code that decides when you wish to grant access although RBAC would not.
Using the RBAC Framework gives you a lot more flexibility, but of cores it makes a higher complexity.

### 2.4.2 Core RBAC

This library contains the core features of RBAC as defined in the standard. It allows you to add and delete users, roles and sessions as well as managing their rights. This library can be used within the RBAC Framework or can be included directly into your application. The configuration is almost independent of your decision.

### 2.4.3 Hierarchical RBAC

This library contains the functionality of limited hierarchical RBAC as defined in the standard. It is derived from Core RBAC and allows you to manage role hierarchies and redefines some of the functions from Core RBAC. This library can be used within the RBAC Framework or can be included directly into your application. The configuration is almost independent of your decision.

## 2.5 Preparing the LDAP server

Before you start to configure openRBAC, you should prepare your LDAP server. If you have it running, it is recommended that you create a subtree for each of the different parts of RBAC, for example:

**users** Usually the users in a LDAP server are stored under the RDN[2] "ou=people". If you ask openRBAC to add a user via the *addUser* function, it will create it there. Respectively openRBAC will search for users under that subtree. If you already have your users defined somewhere else, you should use this DN, of course.

**roles** RBAC is based on roles. It grants permissions to roles and therefore uses roles to make access decisions. To make it possible for openRBAC to store and manage roles, you have to create a subtree, for example "ou=roles" in your LDAP server. If you have an LDAP server where there are already groups defined, you could also try to re-use them as roles and configure the RBAC library respectively. But unless you are familiar with the differences between roles and groups and how openRBAC creates, deletes and manages roles, you should be careful.

**sessions** The RBAC standard defines sessions as continous context for a user. Therefore openRBAC needs a place to store session information in the LDAP server, for example a subtree named "ou=sessions".

**resources** The RBAC standard does not define any functionality to manage resources. Nevertheless you have to create a subtree in your LDAP server where openRBAC can find your resources. To learn more about how to create a resource in your LDAP directory, please read the accompanying documentation of openRBAC.

**SSD/DSD sets** If you decided to use SSD/DSD (Static/Dynamic Separation of Duty) sets in your application, you have to create a subtree for each of them. openRBAC will store

[2]RDN: Relative Distinguished Name

and read informations about restrictions there. You could create subtrees "ou=dsd" as the case may be "ou=ssd".

## 2.6 Configuration

### 2.6.1 Configuring the RBAC Framework

In the "rbac/conf/" directory, you will find two configuration files. You are free to rename them if you wish to.

**rbac.conf.dist** This is an example configuration for the RBAC library. Here you configure the connections to the LDAP server(s) and details about your attributes used in these servers. This file will further be referred as "rbac.conf" (see appendix A.1).

**system.conf.dist** This is an example configuration for the RBAC Framework. Here you configure the path to the rbac.conf, the paths of the files you have to include and the extensions you want to use. This file will further be referred as "system.conf" (see appendix A.2).

Please start by editing rbac.conf. The file is split into sections each containing multiple variables. For each of the sections "user", "role", "session" and "resource" you have to define the appropriate LDAP parameters:

**host** The hostname where the LDAP server is installed on

**port** The port on which the LDAP server is listening

**version** The version to use to talk to the LDAP server

**tls** If you want to use TLS for the connection say 'yes', otherwise say 'no'

**base** The base-DN where the RBAC library starts searching for an entry

**binddn** The DN with which the RBAC library binds to the LDAP server

**password** The password corresponding to the binddn parameter

**filter** A basic filter that is used in every search

**namingattribute** The naming attribute of an entry (f.e. "uid' for a user entry)

For the role section there is the additional "assignedattribute" parameter. In this attribute the RBAC library stores the users assigned to a role.
For the resource section there is the additional "aliasattribute" parameter. If you specify an attribute different from the "namingattribute", the RBAC library searches for both when searching for a resource.
The ssd and dsd sections are only needed if you decide to use one of these extensions. Otherwise you can ignore them.
The "errorCode" and "errorDescription" sections can be left untouched unless you want to change the output messages the RBAC library should give.

Please edit the system.conf next. There you find various configuration parameters you have to adapt to your installation. Please keep in mind that every path has to be either absolut or relative to your application that is using openRBAC.

**configuration** This parameter tells the RBAC Framework where to find the configuration file for the RBAC library that you have already modified.

**rbac** The "class" attribute defines which RBAC library you wish to use. Possible values are "RBACcore" and "RBAClimitedHirarchical". As child nodes you see a lot of "require" statements. As mentioned in section 2.3 the code consists of two parts. You now have to tell the RBAC Framework where it can find all needed classes and interfaces. Depending on how you decided where to store the code, you have to adapt the pathes.

**extension** If you want to use extensions, you have to configure them. For each extension class you have to define an "extension" statement. The "class" attribute contains the name of the class and the "file" attribute contains the path and filename in which the class is defined.

A simple example on how to integrate the RBAC Framework into your application could be:

```
<?php
require_once( "RBAC.class.php" );

$rbac = new RBAC( "../conf/system.conf", "./", "../lib" );
$rbac->createSession( "martin.haase@daasi.de", Array(), "abc" );
?>
```

### 2.6.2 Configuring openRBAC without the RBAC Framework

If you want to use the Core RBAC or limited hierarchical RBAC *without* the RBAC Framework, you have to adapt the rbac.conf as described in section 2.6.1, but you can omit the part that describes the adaption of system.conf. Because the RBAC Framework now does not include the required classes and interfaces, you have to define the paths via constants in your application so that the RBAC library can do this on its own:

**RBAC_LIB_PATH** This constant has to contain the path to the mwlib files either absolute or relative to your application.

**RBAC_PATH** This constant has to contain the path to the openRBAC files either absolute or relative to your application.

You have to define these constants in your application before you include the RBAC library! A simple example on how to integrate the Core RBAC into your application could be:

```
<?php
define( "RBAC_LIB_PATH", "../lib" );
define( "RBAC_PATH", "./" );
require_once( "RBACcore.class.php" );
```

```
$rbac = new RBACcore( "conf/rbac.conf" );
$rbac->createSession( "jondoe", Array( "student", "hiwi,staff" ), "s13243355a46" );
?>
```

### 2.7 Closing words

You now have installed openRBAC. Remember that openRBAC is just a library and cannot replace an application. Even if you just want to have a graphical interface that maps the RBAC functionality defined in the standard, you have to write it on your own. It is not an "out-of-the-box" management tool.

## 3 RBAC with XACML via SOAP

Additional to the RBAC library and RBAC Framework, you will find a SOAP[W3C07] server and client in the *SOAP* directory. Both use the SAML 2.0 profile of XACML v2.0[AL05] to communicate with each other. The clients gives you the possibility to call the RBAC function *checkAccess* via a SOAP service. SOAP transports a SAML Request and Response adapted to XACML. An example SOAP request is given in App. C.4 and an appropriate SOAP response is given in App. C.5.

### 3.1 Preparations

To use RBAC with SOAP you have to have a webserver like Apache having PHP5 module included. You than have to configure Apache to serve the content of the *SOAP* directory. Make sure that it is possible for xacml.php to include all of the RBAC library and RBAC Framework.

### 3.2 Configuration

At this point it is possible to reach the file *xacml.php* over the HTTP protocol using a browser or something similar. Now you have to edit three files and replace the placeholders within them:

**xacml.wsdl** Go to the end of the file. In the section *Servicedefinition* you will find the placeholder <*YOUR_LOCATION*>. Replace it with the path where a client will be able to call your *xacml.php*.

**xacml.php** At the top of the file you will find the placeholder <*PATH_TO_RBAC*>. Replace it with a relative or absolute path to your RBAC Framework. This value will be used twice in *xacml.php*. First it is used to include the RBAC Framework and second to tell the RBAC Framework where it is located. Afer having set this, you need to replace the placeholder <*PATH_TO_WSDL*> in the same file. Set the path to the WSDL file you have just edited to fit your installation.

**xacmlCheckAccess.php** This is the client. If you want to use it you have to replace the placeholder *<PATH_TO_WSDL>* as you did in *xacml.php*. *Notice*: If you have a close look at the WSDL file (see App. B), you might recognice that the imported XML schema is not located at the OASIS web page but at the web page of DAASI International GmbH. The reason for this is, that the XML schema you can download at OASIS is flawed and cannot be used for an XSD import. To verify this, download the files from OASIS and make a *diff* between the corresponding files served at the DAASI web page and included in this document (see App. C).

### 3.3 Closing words

If your RBAC is configured well and found to work properly, you now should be able to determin decisions of RBAC via a SOAP service using standardised SAML/XACML. But remember that this project is still very young. So at the moment it is not possible to tell the SOAP servive to examine certificates send within the SAML request to decide if the client is trusted.

## A Configuration files

### A.1 rbac.conf.dist

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>

  <section name="user">
    <var name="host">ldap.example.org</var>
    <var name="port">389</var>
    <var name="version">3</var>
    <var name="tls">no</var>
    <var name="base">ou=users,ou=rbac,dc=example,dc=org</var>
    <var name="binddn">cn=application,ou=dsa,dc=example,dc=org</var>
    <var name="password">secret</var>
    <var name="filter">(objectClass=inetorgperson)</var>
    <var name="namingattribute">uid</var>
  </section>

  <section name="role">
    <var name="host">ldap.example.org</var>
    <var name="port">389</var>
    <var name="version">3</var>
    <var name="tls">no</var>
    <var name="base">ou=roles,ou=rbac,dc=example,dc=org</var>
    <var name="binddn">cn=application,ou=dsa,dc=example,dc=org</var>
    <var name="password">secret</var>
    <var name="namingattribute">rbacname</var>
    <var name="filter">(objectClass=rbacrole)</var>

    <!-- May also be member to have greater compatibility with
         allready existing entries in your LDAP-server //-->
    <var name="assignedattribute">rbacperformer</var>
  </section>

  <section name="project">
    <var name="base">ou=Projekt-Teilnehmer,ou=roles,ou=rbac,dc=example,dc=org</var>
  </section>

  <section name="session">
    <var name="host">ldap.example.org</var>
    <var name="port">389</var>
    <var name="version">3</var>
    <var name="tls">no</var>
    <var name="base">ou=sessions,ou=rbac,dc=example,dc=org</var>
    <var name="binddn">cn=application,ou=dsa,dc=example,dc=org</var>
    <var name="password">secret</var>
    <var name="namingattribute">rbacname</var>
    <var name="filter">(objectClass=rbacsession)</var>
  </section>
```

```
  <section name="resource">
    <var name="host">ldap.example.org</var>
    <var name="port">389</var>
    <var name="version">3</var>
    <var name="tls">no</var>
    <var name="base">ou=rbac,dc=example,dc=org</var>
    <var name="binddn">cn=application,ou=dsa,dc=example,dc=org</var>
    <var name="password">secret</var>
    <var name="namingattribute">rbacname</var>
    <var name="aliasattribute">rbacname</var>
    <var name="filter">(objectClass=rbacresource)</var>
  </section>

  <section name="ssd">
    <var name="host">ldap.example.org</var>
    <var name="port">389</var>
    <var name="version">3</var>
    <var name="tls">no</var>
    <var name="base">ou=ssd,ou=rbac,dc=example,dc=org</var>
    <var name="binddn">cn=application,ou=dsa,dc=example,dc=org</var>
    <var name="password">secret</var>
    <var name="filter">(objectClass=rbacssd)</var>
  </section>

  <section name="dsd">
    <var name="host">ldap.example.org</var>
    <var name="port">389</var>
    <var name="version">3</var>
    <var name="tls">no</var>
    <var name="base">ou=dsd,ou=rbac,dc=example,dc=org</var>
    <var name="binddn">cn=application,ou=dsa,dc=example,dc=org</var>
    <var name="password">secret</var>
    <var name="filter">(objectClass=rbacdsd)</var>
  </section>


  <section name="errorCode">
    <var name="OK">1</var>
    <var name="RESOURCE_OPERATION_ERROR">2</var>
    <var name="RESOURCE_UNKNOWN">4</var>
    <var name="USER_SESSION_ERROR">8</var>
    <var name="SESSION_ALLREADY_EXISTS">16</var>
    <var name="SESSION_DOES_NOT_EXISTS">32</var>
    <var name="USER_UNKNOWN">64</var>
    <var name="USER_ROLE_ERROR">128</var>
    <var name="USER_ALLREADY_EXISTS">256</var>
    <var name="INVALID_USER_FORMAT">512</var>
    <var name="ROLE_ALLREADY_EXISTS">1024</var>
    <var name="ROLE_UNKNOWN">2048</var>
    <var name="LDAP_ERROR">4096</var>
    <var name="UNKNOWN_ERROR">8192</var>
    <var name="SD_ALLREADY_EXISTS">16384</var>
    <var name="SD_CARDINALITY">32768</var>
    <var name="SD_UNKNOWN">65536</var>
  </section>

  <section name="errorDescription">
    <var name="OK">Ok</var>
    <var name="RESOURCE_OPERATION_ERROR">This resource-operation-combination is invalid.</var>
    <var name="RESOURCE_UNKNOWN">The resource is not known or not uniqueue.</var>
    <var name="USER_SESSION_ERROR">The user you gave is not the owner of this session.</var>
    <var name="SESSION_ALLREADY_EXISTS">The session allready exists so you can not create it.</var>
    <var name="SESSION_DOES_NOT_EXISTS">The session does not exist.</var>
    <var name="USER_UNKNOWN">The user is not known.</var>
    <var name="USER_ROLE_ERROR">This user-role-combination is invalid.</var>
    <var name="USER_ALLREADY_EXISTS">The user allready exists.</var>
    <var name="INVALID_USER_FORMAT">The user has to have the the format: &lt;username&gt;@&lt;domain&gt;
                                (foo@example.org)</var>
    <var name="ROLE_ALLREADY_EXISTS">The role allready exists.</var>
    <var name="ROLE_UNKNOWN">The role is unknown.</var>
    <var name="LDAP_ERROR">An LDAP-Error occured, see description: </var>
    <var name="UNKNOWN_ERROR">An error occured.</var>
    <var name="SD_ALLREADY_EXISTS">The Separation of Duty Set allready exists.</var>
    <var name="SD_CARDINALITY">The given cardinality is invalid! Make sure it is >= 2</var>
    <var name="SD_UNKNOWN">The Separation of Duty set is unknown</var>
  </section>

</configuration>
```

## A.2 system.conf.dist

```
<?xml version="1.0" encoding="UTF-8"?>
<system>

  <!-- This is the system-configuration for the RBAC-Framework.
       The RBAC libraries need an additional configuration-file
       where you have to define all LDAP-specific settings. Tell
       the RBAC-Framework where it can find this configuration.
       Specify a full path or a path relative to your main
       application. -->
  <configuration file="PATH_TO/rbac.conf" />


  <!-- Here you can specify if you want to use Core RBAC or the
       Limited Hierarchical RBAC. In order to work properly both
       libraries need a couple of classes specified through
       the "require" statements. Again you have to specify a full
       path or a path relative to your main application. -->
  <rbac class="RBAClimitedHirarchical">
    <require file="PATH_TO/iContext.interface.php" />
    <require file="PATH_TO/iHelper.interface.php" />
    <require file="PATH_TO/iCrypto.interface.php" />
    <require file="PATH_TO/iLDAP.interface.php" />
    <require file="PATH_TO/iRBACcore.interface.php" />
    <require file="PATH_TO/iRBAClimitedHirarchical.interface.php" />

    <require file="PATH_TO/RBACException.class.php" />
    <require file="PATH_TO/RBACExtension.class.php" />
    <require file="PATH_TO/Context.class.php" />
    <require file="PATH_TO/Helper.class.php" />
    <require file="PATH_TO/LDAP.class.php" />
    <require file="PATH_TO/Crypto.class.php" />
    <require file="PATH_TO/SimpleConfig.class.php" />
    <require file="PATH_TO/RBACcore.class.php" />
    <require file="PATH_TO/RBAClimitedHirarchical.class.php" />
  </rbac>


  <!-- Specify the extensions you want to use through the
       RBAC-Framework. Give a full path or a path relative
       to your main application. -->
  <extension class="SSD" file="PATH_TO/SSD.class.php" />
  <extension class="DSD" file="PATH_TO/DSD.class.php" />
  <extension class="UserEntry" file="PATH_TO/UserEntry.class.php" />
  <extension class="Logger" file="PATH_TO/Logger.class.php" />

</system>
```

# B xacml.wsdl

```
<?xml version="1.0" encoding="UTF-8"?>

<wsdl:definitions name="xacml"
                  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
                  xmlns:tns="http://daasi.de/namespaces/rbac/xacml"
                  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                  targetNamespace="http://daasi.de/namespaces/rbac/xacml"
                  xmlns:xacml-samlp="urn:oasis:xacml:2.0:saml:protocol:schema:os"
                  xmlns:xacml-saml="urn:oasis:xacml:2.0:saml:assertion:schema:os">


  <!--
  ####################
  # Type Definitions #
  ####################
  //-->
  <wsdl:types>
    <xsd:schema targetNamespace="http://daasi.de/namespaces/rbac/xacml">
      <xsd:import namespace="urn:oasis:xacml:2.0:saml:assertion:schema:os"
                  schemaLocation="http://www.daasi.de/schema/oasis/access_control-xacml-2.0-saml-assertion-schema-os.xsd" />
      <xsd:import namespace="urn:oasis:xacml:2.0:saml:protocol:schema:os"
                  schemaLocation="http://www.daasi.de/schema/oasis/access_control-xacml-2.0-saml-protocol-schema-os.xsd"/>

    </xsd:schema>

  </wsdl:types>


  <!--
```

```
  #################
  # WSDL-Messages #
  #################
  //-->
  <!-- #### checkXACMLaccess #### //-->
  <wsdl:message name="checkXACMLaccessRequest">
    <wsdl:part element="xacml-samlp:XACMLAuthzDecisionQuery" name="checkXACMLaccessInput" />
  </wsdl:message>
  <wsdl:message name="checkXACMLaccessResponse">
    <wsdl:part element="xacml-saml:XACMLAuthzDecisionStatement" name="checkXACMLaccessOutput" />
  </wsdl:message>


  <!--
  ########################
  # Port-Type-Definitions #
  ########################
  //-->
  <wsdl:portType name="port_xacml">

    <!-- #### checkXACMLaccess #### //-->
    <wsdl:operation name="checkXACMLaccess">
      <wsdl:input message="tns:checkXACMLaccessRequest" />
      <wsdl:output message="tns:checkXACMLaccessResponse" />
    </wsdl:operation>

  </wsdl:portType>


  <!--
  ###########
  # Binding #
  ###########
  //-->
  <wsdl:binding name="binding_xacml" type="tns:port_xacml">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />

    <!-- #### checkXACMLaccess #### //-->
    <wsdl:operation name="checkXACMLaccess">
      <soap:operation soapAction="http://daasi.de/rbac/xacml/checkXACMLaccess" />
      <wsdl:input><soap:body use="literal" /></wsdl:input>
      <wsdl:output><soap:body use="literal" /></wsdl:output>
    </wsdl:operation>

  </wsdl:binding>


  <!--
  ####################
  # Servicedefinition #
  ####################
  //-->
  <wsdl:service name="xacml">
    <wsdl:port binding="tns:binding_xacml" name="tns:xacml">
      <soap:address location="http://<YOUR_LOCATION>/xacml.php" />
    </wsdl:port>
  </wsdl:service>

</wsdl:definitions>
```

# C Schema files

## C.1 XACML Saml Protocol

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
    targetNamespace="urn:oasis:xacml:2.0:saml:protocol:schema:os"
    xmlns:tns="urn:oasis:xacml:2.0:saml:protocol:schema:os"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os"
    xmlns:xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
  <xs:import namespace="urn:oasis:names:tc:SAML:2.0:protocol"
      schemaLocation="http://www.daasi.de/schema/oasis/saml-schema-protocol-2.0.xsd"/>
  <xs:import namespace="urn:oasis:names:tc:xacml:2.0:context:schema:os"
```

```
      schemaLocation="http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-context-schema-os.xsd"/>
  <xs:annotation>
    <xs:documentation>
        Document identifier: access_control-xacml-2.0-saml-protocol-schema-os.xsd
        Location: http://docs.oasis-open.org/xacml/2.0/
                        access_control-xacml-2.0-saml-protocol-schema-os.xsd
    </xs:documentation>
  </xs:annotation>
  <!--    -->
  <xs:element name="XACMLAuthzDecisionQuery"
          type="tns:XACMLAuthzDecisionQueryType"/>
  <xs:complexType name="XACMLAuthzDecisionQueryType">
    <xs:complexContent>
      <xs:extension base="samlp:RequestAbstractType">
        <xs:sequence>
          <xs:element ref="xacml-context:Request"/>
        </xs:sequence>
        <xs:attribute name="InputContextOnly"
                      type="boolean"
                      use="optional"
                      default="false"/>
        <xs:attribute name="ReturnContext"
                      type="boolean"
                      use="optional"
                      default="false"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <!--    -->
  <xs:element name="XACMLPolicyQuery"
          type="tns:XACMLPolicyQueryType"/>
  <xs:complexType name="XACMLPolicyQueryType">
    <xs:complexContent>
      <xs:extension base="samlp:RequestAbstractType">
        <xs:choice minOccurs="0" maxOccurs="unbounded">
          <xs:element ref="xacml-context:Request"/>
          <xs:element ref="xacml:Target"/>
          <xs:element ref="xacml:PolicySetIdReference"/>
          <xs:element ref="xacml:PolicyIdReference"/>
        </xs:choice>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</schema>
```

## C.2 XACML Saml Assertion

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
    targetNamespace="urn:oasis:xacml:2.0:saml:assertion:schema:os"
    xmlns:tns="urn:oasis:xacml:2.0:saml:assertion:schema:os"
    xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os"
    xmlns:xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
    elementFormDefault="unqualified"
    attributeFormDefault="unqualified"
    blockDefault="substitution"
    version="2.0">
  <xs:import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
      schemaLocation="http://www.daasi.de/schema/oasis/saml-schema-assertion-2.0.xsd"/>
  <xs:import namespace="urn:oasis:names:tc:xacml:2.0:context:schema:os"
      schemaLocation="http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-context-schema-os.xsd"/>
  <xs:annotation>
    <xs:documentation>
        Document identifier: access_control-xacml-2.0-saml-assertion-schema-cd-02.xsd
        Location: http://docs.oasis-open.org/xacml/2.0/
                        access_control-xacml-2.0-saml-assertion-schema-cd-os.xsd
    </xs:documentation>
  </xs:annotation>
  <!--    -->
  <xs:element name="XACMLAuthzDecisionStatement"
          type="tns:XACMLAuthzDecisionStatementType"/>
  <xs:complexType name="XACMLAuthzDecisionStatementType">
    <xs:complexContent>
      <xs:extension base="saml:StatementAbstractType">
        <xs:sequence>
          <xs:element ref="xacml-context:Response"/>
          <xs:element ref="xacml-context:Request"  minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
```

```
    </xs:complexContent>
  </xs:complexType>
  <!--    -->
  <xs:element name="XACMLPolicyStatement"
           type="tns:XACMLPolicyStatementType"/>
  <xs:complexType name="XACMLPolicyStatementType">
    <xs:complexContent>
      <xs:extension base="saml:StatementAbstractType">
        <xs:choice minOccurs="0" maxOccurs="unbounded">
          <xs:element ref="xacml:Policy"/>
          <xs:element ref="xacml:PolicySet"/>
        </xs:choice>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</schema>
```

## C.3 RBAC with XACML via SOAP

## C.4 SOAP request

```
<SOAP-ENV:Envelope
      xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
      xmlns:ns1="urn:oasis:names:tc:xacml:2.0:context:schema:os"
      xmlns:ns2="urn:oasis:xacml:2.0:saml:protocol:schema:os">

<SOAP-ENV:Body>
  <ns2:XACMLAuthzDecisionQuery ID="abcde1234" Version="2.0" ReturnContext="true">
    <ns1:Request>
      <ns1:Subject>
        <ns1:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
                       DataType="http://www.w3.org/2001/XMLSchema#string">
          <ns1:AttributeValue>
            SID_ryytp55xUIvmoXEv663QWeYoj0t2s6COeX56nbGpUkxj1R7pCXNqT49umfFs1I9JvI
          </ns1:AttributeValue>
        </ns1:Attribute>
      </ns1:Subject>
      <ns1:Resource>
        <ns1:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
                       DataType="http://www.w3.org/2001/XMLSchema#anyURI">
          <ns1:AttributeValue>
            textgrid:TGPR3:Die+Leiden+des+jungen+Werther+-+Zweyter+Theil:20080327T170038:xml%2Ftei:1
          </ns1:AttributeValue>
        </ns1:Attribute>
      </ns1:Resource>
      <ns1:Action>
        <ns1:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
                       DataType="http://www.w3.org/2001/XMLSchema#string">
          <ns1:AttributeValue>
            read
          </ns1:AttributeValue>
        </ns1:Attribute>
      </ns1:Action>
      <ns1:Environment/>
    </ns1:Request>
  </ns2:XACMLAuthzDecisionQuery>
</SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

## C.5 SOAP response

```
<SOAP-ENV:Envelope
      xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
      xmlns:ns1="urn:oasis:names:tc:xacml:2.0:context:schema:os"
      xmlns:ns2="urn:oasis:xacml:2.0:saml:assertion:schema:os>

<SOAP-ENV:Body>
  <ns2:XACMLAuthzDecisionStatement>
    <ns1:Response>
      <ns1:Result>
        <ns1:Decision>
          Permit
        </ns1:Decision>
      </ns1:Result>
    </ns1:Response>
    <ns1:Request>
      <ns1:Subject>
        <ns1:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
                       DataType="http://www.w3.org/2001/XMLSchema#string">
```

```
          <ns1:AttributeValue>
            SID_ryytp55xUIvmoXEv663QWeYoj0t2s6COeX56nbGpUkxj1R7pCXNqT49umfFs1I9JvI
          </ns1:AttributeValue>
        </ns1:Attribute>
      </ns1:Subject>
      <ns1:Resource>
        <ns1:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
                       DataType="http://www.w3.org/2001/XMLSchema#anyURI">
          <ns1:AttributeValue>
            textgrid:TGPR3:Die+Leiden+des+jungen+Werther+-+Zweyter+Theil:20080327T170038:xml%2Ftei:1
          </ns1:AttributeValue>
        </ns1:Attribute>
      </ns1:Resource>
      <ns1:Action>
        <ns1:Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
                       DataType="http://www.w3.org/2001/XMLSchema#string">
          <ns1:AttributeValue>
            read
          </ns1:AttributeValue>
        </ns1:Attribute>
      </ns1:Action>
      <ns1:Environment/>
    </ns1:Request>
  </ns2:XACMLAuthzDecisionStatement>
</SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

# References


[AL05] Anne Anderson and Hal Lockhart. Saml 2.0 profile of xacml v2.0. Technical report, OASIS, 2005.

[ANS04] ANSI incits. *Role Based Access Control*, 2004.

[W3C07] W3C. *SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)*, 2007.