CORISECIO
Corporate Information Security

Open SOA Gateway

User Guide

Copyrights

Conventions

Typographic representation:

Screen text and KEYPAD

Texts appearing on the screen, key pads like e.g. system messages, menu titles, - texts, or buttons are displayed as follows:

Example: Enter your name in the **User** field and click **OK**.

`Files and folders`

File and folder structures are marked as follows:

Example: Download the file `doSpellingSuggestion.xml` from the folder `Examples`.

`Entries`

User entries are displayed as follows:

Example: Enter `login` here.

Quotation

Quotations and references are displayed as follows:

Example: Further information can be found in chapter "*Overview*" on the following pages.

`Weblinks`

Web addresses and links are displayed as follows:

Example: `http://www.corisecio.com`

1 Introduction

The securityRunTime (secRT) is a capacious security infrastructure for service oriented architectures (SOA). The focus is the automation of Security Design, Roll-Out, Deployment and Security Management. The secRT itself is constructed according to SOA principles and is based on open standards like Java, XML and Web Services and can be integrated in all SOA infrastructures. Supported are:

- Administration of entities (Identity Management)
- Access Management
- Protection of messages (Message Protection)
- Cryptography (Encryptions and signatures)
- Key Management
- Privacy
- Security guidelines – Enforcement and Decision

The secRT plus loaded adapters is called Connector. As the installation of a Connector is analogous to the installation of the secRT, only secRT is used for simplification in the following.

2 System Requirements

The statements regarding processor, RAM and hard disc space can only be considered as reference values, since the need of system resources is mainly depending on the extend of use of the secRT. Reliable statements are received only by tests in your system environment.

Processor	**Intel Pentium IV** with 2,4 GHz or better
RAM	**1024 MB** or better
Free Hard-Disc Memory	**10 GB** or better (e. g. for Logging)
Operating System	• **Windows XP (SP 2) - 32/64 Bit** • **Windows 2003 (SP 1) - 32/64 Bit** • **Suse Linux Enterprise 10**
Software	• **Java Software Development Kit 1.5 or 1.6** • **Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy** • **Apache Tomcat 5.5**

3 Installation

A Connector is run as a web application on the Application Server. If necessary, please consider the documentation related to your Application Server.

For deployment in the Apache Tomcat, rename the Connector-War file, so that the file name equates the requested deployment path. Keep the file extension. Copy the file into the Tomcat's webapps directory. Re-start Tomcat if necessary.

After deployment, test the installation by accessing the web application.

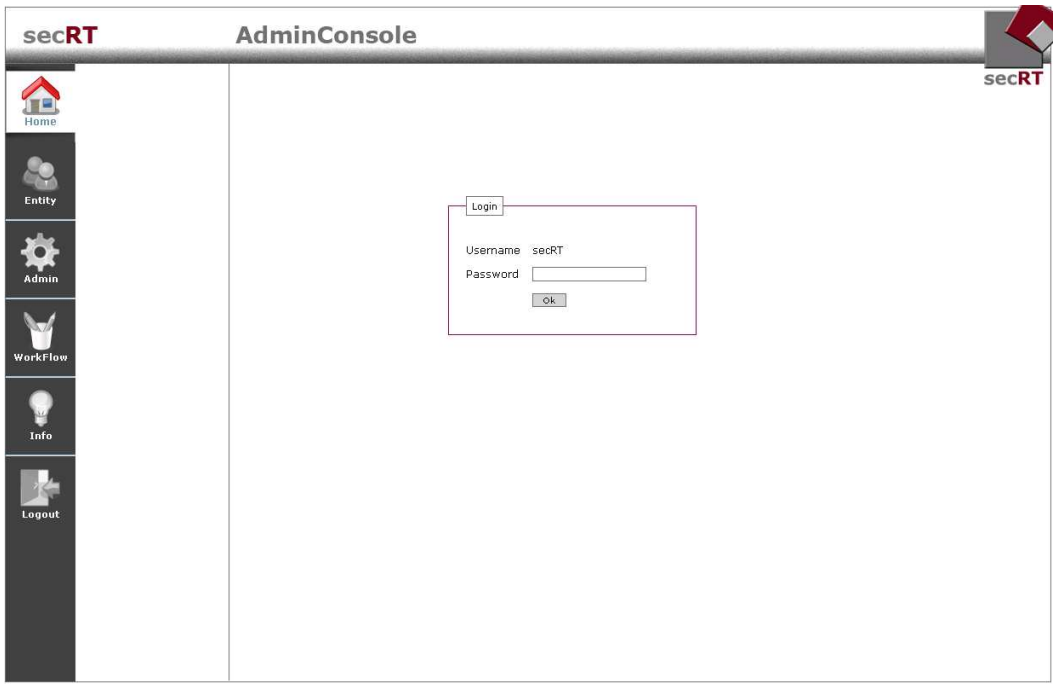
Enter the following into your browser's address bar:

```
http://<hostname>:<port>/<filename without extension>
```

Example: If the war file is renamed to connector.war and if your Tomcat installation runs under the address localhost:8080, then the address to be entered will be:

```
http://localhost:8080/connector
```

The Connector's log-in page is displayed.



4 Administration

The configuration is conducted completely through the Connector's web interface, the AdminConsole.

4.1 Log-in

First you have to log in at the AdminConsole. Access the log-in page of the AdminConsole in your browser as described in chapter 3. User name and password are predefined for the Open Source version of the SOA Security Framework.

User Name: secRT

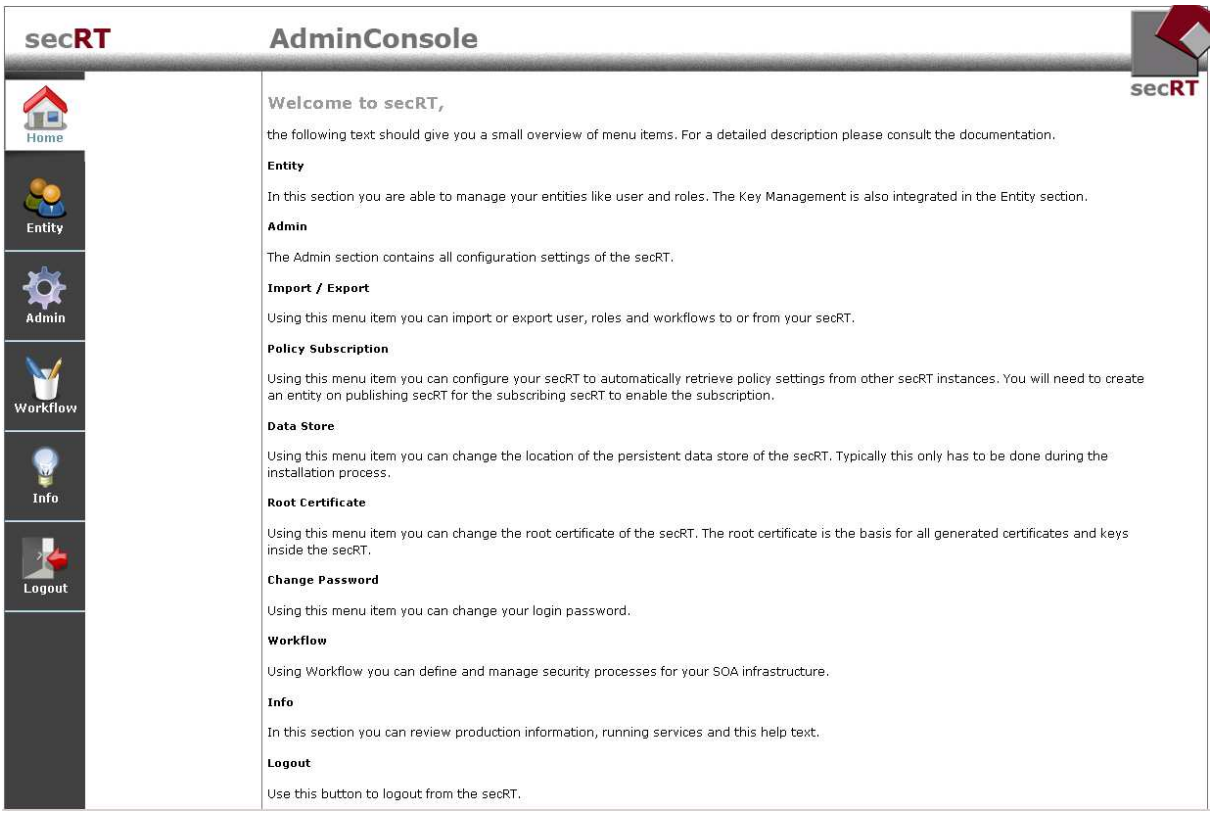
Password: secRT

Enter user name and password and click **Ok**. After correct entry, the overview page of the AdminConsole will appear.

4.2 Home

After log-in, the start page of the secRT is displayed. Here, you will find an overview of the individual menu items and a brief help text.

If you log-in for the first time, the configuration page for the Security Repository will be displayed. In this case, please see chapter 4.4.3 for further information.

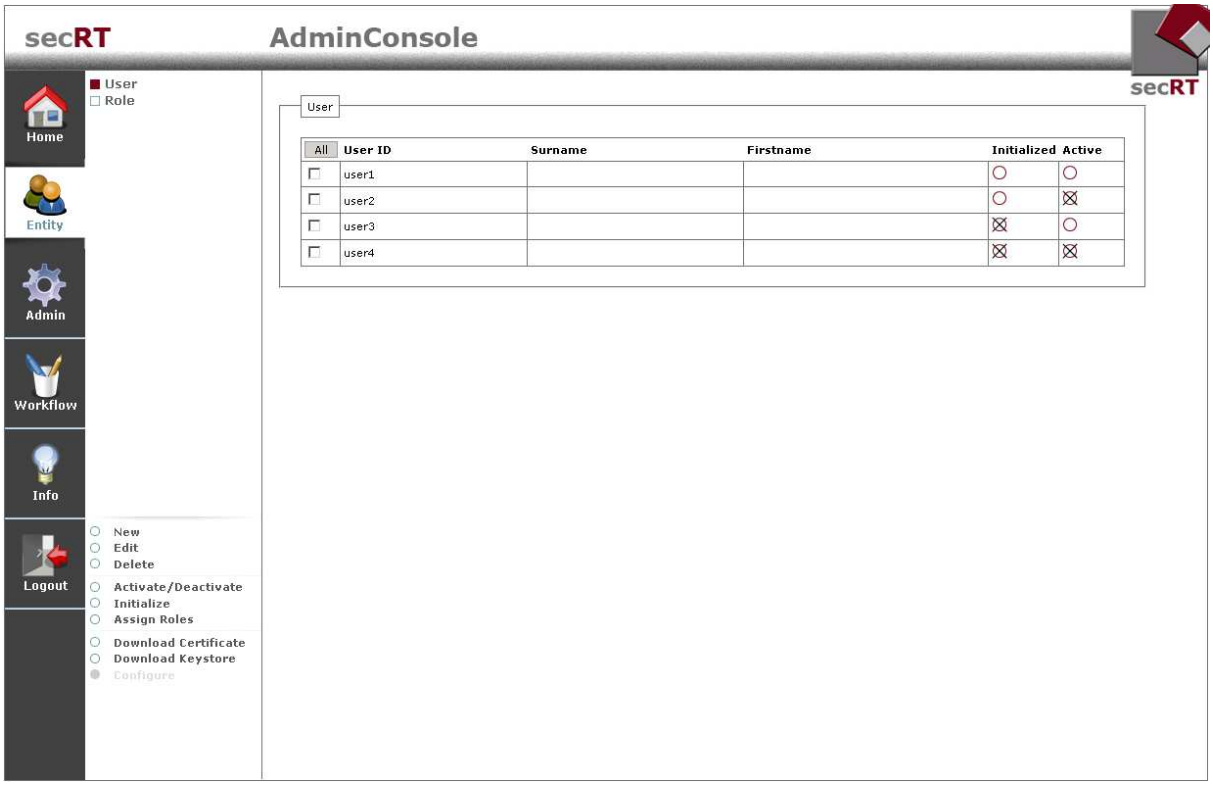


4.3 Entity

In this menu the Entity Administration is located.

4.3.1 User

Users are entities, which can be assigned with private keys or roles. These assignments can be used at modeling of workflows. Click on User to enter the user administration.



All existing users are listed on the overview page. Four different user status types are possible, depending on whether a user is initialized or activated.

A user is initialized, if a key pair consisting of certificate and private key is assigned to him.

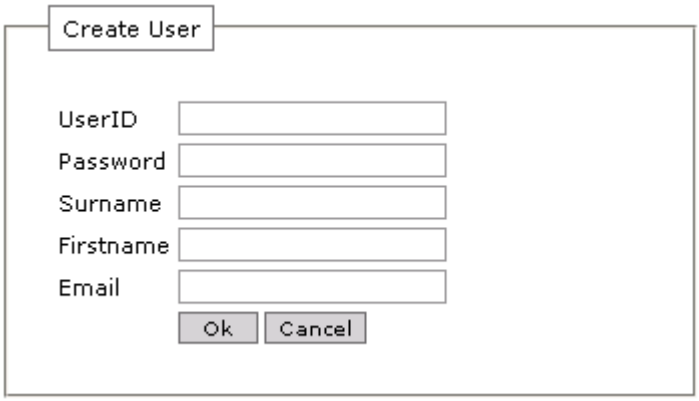
A user has to be activated in addition, e. g. to be able to use his keys for modeling of workflows.

Depending on the user status, different actions are possible. These are executable through the links besides the main navigation bar from the user management and always refer to the selected user/s.

Exception here is the creation of users. This can be done through the link **Create**. A description of the available user management actions follows.

4.3.1.1 New

When clicking this link, the form for creation of users appears:



Permitted entry values are:

Field	Description	Acceptance Criteria
UserID	ID of the user to be created	4-50 characters according to the regular expression ([-]|[_]|[.]|[a-z]|[0-9])+, precisely
Password	User password	0-60 characters
Surname	Surname	0-60 characters
First Name	First name	0-60 characters
Email	Email address	0-60 characters, valid email address if set

Clicking on the button will send the data and create the new user if the acceptance criteria are fulfilled.

4.3.1.2 Edit

Through a click on Edit, you may edit the properties of the selected user. The UserID cannot be edited. The acceptance criteria are the same as those for creating a user.

4.3.1.3 Delete

Through this link, selected users are deleted. The user information is completely removed from the database, the issued certificates are deleted.

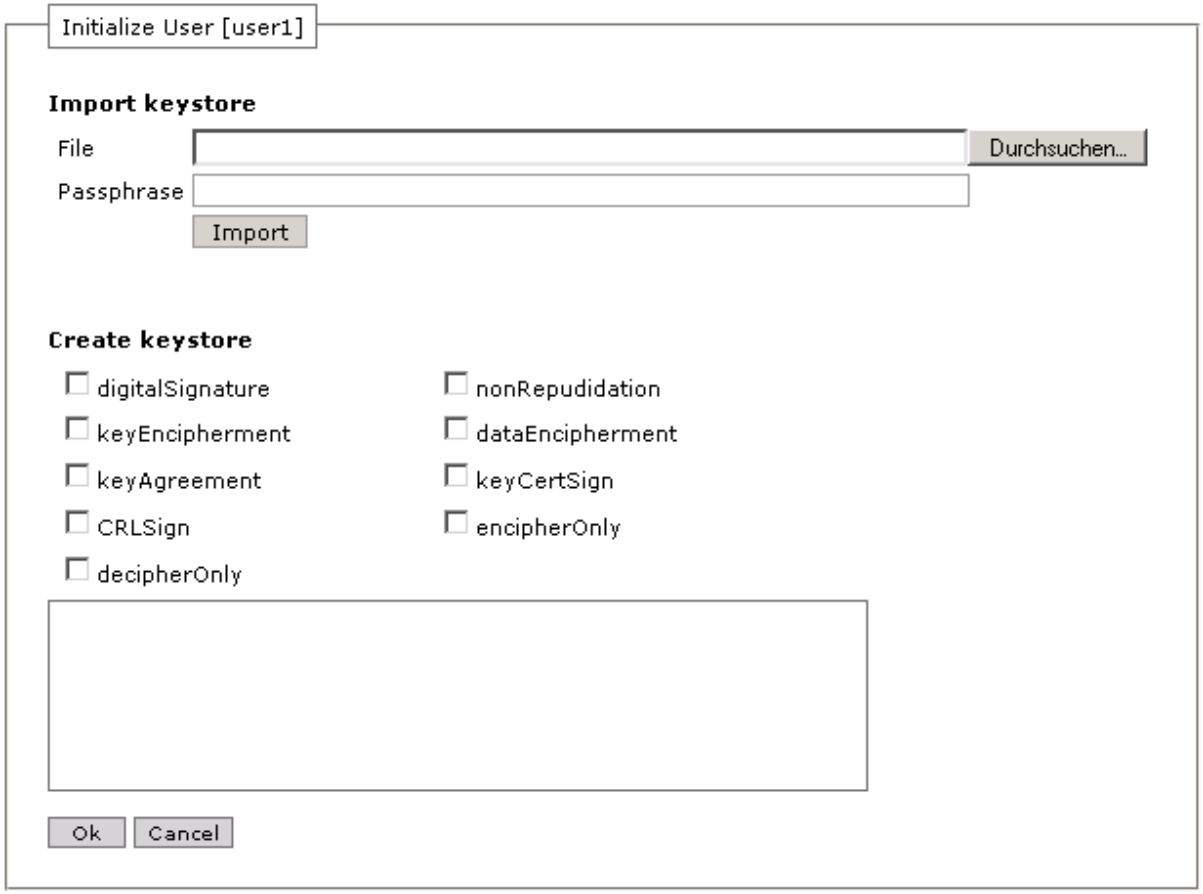
4.3.1.4 Activate / Deactivate

Through this link, a user is activated resp. deactivated. Contrary to deleting, at deactivating of a user, no database entries are erased. Deactivated users may be reactivated.

4.3.1.5 Initialize

Here, you can generate or import a key set for a user. At generating a key set, the user password is used as password for the PKCS#12 Key store.

This form appears:



Import a PKCS#12 Key store via the **Import** button. You have to enter a key store file and the corresponding password.

Alternatively, generate a new key set in the **Create Keystore** area. Activate the intended key usage of the certificate. Additionally, certificate extensions may be entered in the text field in OID-Notation (Example: 1.3.6.1.5.5.7.3.1). These have to be in a separate row each.

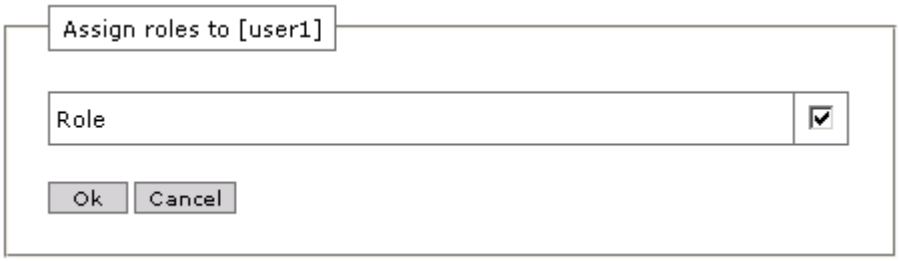
After configuration of usage, click on the **Ok** button to send the data.

The button **Cancel** terminates the action.

Already initialized users may be initialized again. Previously existing certificates are deleted.

4.3.1.6 Assign Roles

After clicking this link, a form is displayed showing user´s role assignment, which can also be changed thereupon. The roles that are activated in the list are assigned to the user.



4.3.1.7 Download Certificate

A user certificate can be downloaded here. A file in PEM format is created. This action is available for initialized users only.

4.3.1.8 Download Keystore

A user´s keypair can be downloaded here. A file in PKCS#12 format is created. The password protecting the file is the same as the actual user password.

Role

In the Open Source secRT, entities like users can be grouped through role assignment. For Role Management, click Role in the respective menu of the AdminConsole. It will lead you to the Role Management page.

4.3.1.9 New

Here, you can create a new role. The following form is displayed:

Cancel terminates the process of generating roles. Click **Ok** and - if the values entered do fulfill the acceptance criteria - the group will be created.

Field	Description	Acceptance Criteria
Name	Name of the role to be created	4-200 characters, no inverted comma
Description	Description of role	0-1000 characters

4.3.1.10 Edit

With a click on **Edit**, you can change the description text of the selected role.

4.3.1.11 Delete

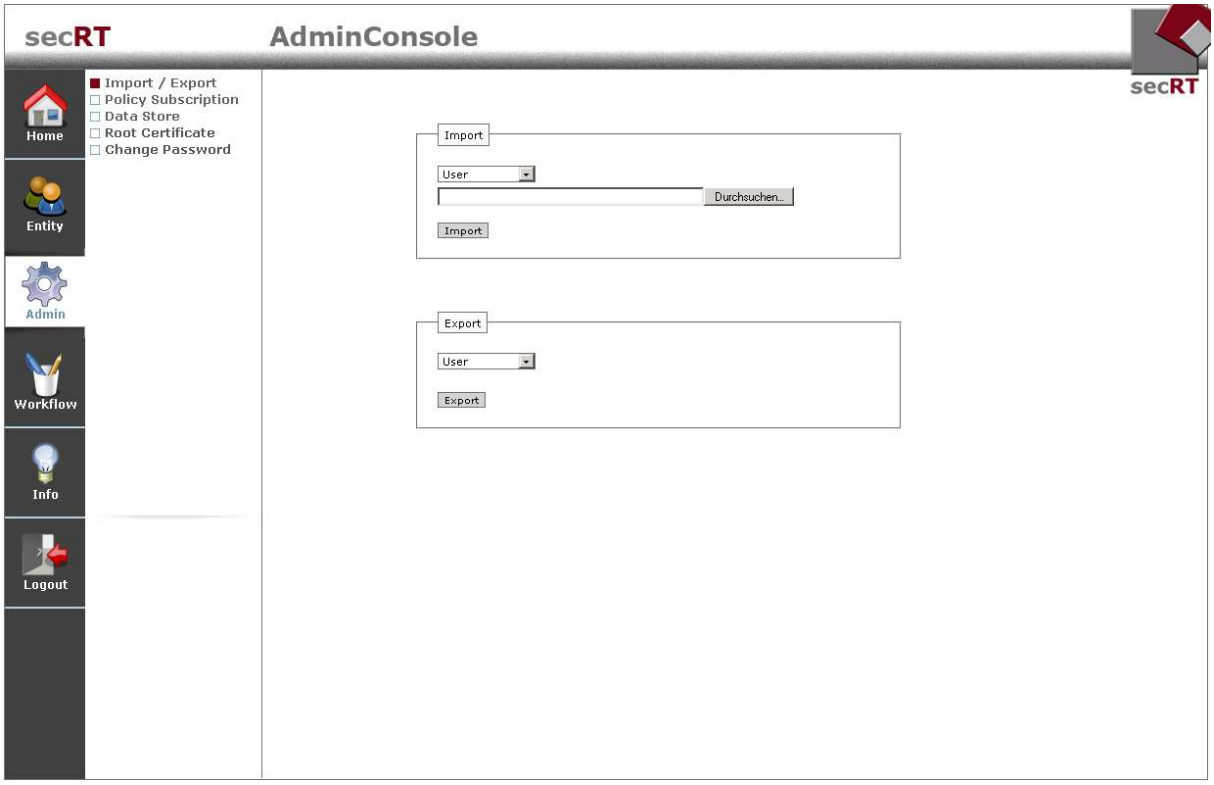
Click **Delete** to delete the selected role. The roles are deleted after confirming the pop-up prompt.

4.4 Admin

Under menu point Admin, you can execute miscellaneous administrative operations, which are described in this chapter.

4.4.1 Import / Export

At Import / Export, you may export and import the entities user, roles, user role-assignments and workflows.



Select the entity type from the selection list and click the Export button to export an entity. To import an entity, select the import file and the type from the selection list and click Import. Here new entities are inserted and existing ones are updated.

The entries in the files are separated by a line break. The fields of an entry are separated by a comma.

The description of the individual formats follows. The format of the workflow file is not described, as only the import of exported files is supported.

4.4.1.1 User

Username	Username
Password	Initialization password
Email Address	Email address

Surname	Surname
First Name	First name
Activation Status	True for active users, false for inactive users
Certificate	Base64-encoded DER Certificate
Private Key	Base64-encoded PKCS#12-Keystore protected with the Initialization Password and bears a key with the Username as Alias.

4.4.1.2 Roles

Role Name	Name of the role
Description	Description text of the role
Group Name	Name of the group to which the role belongs

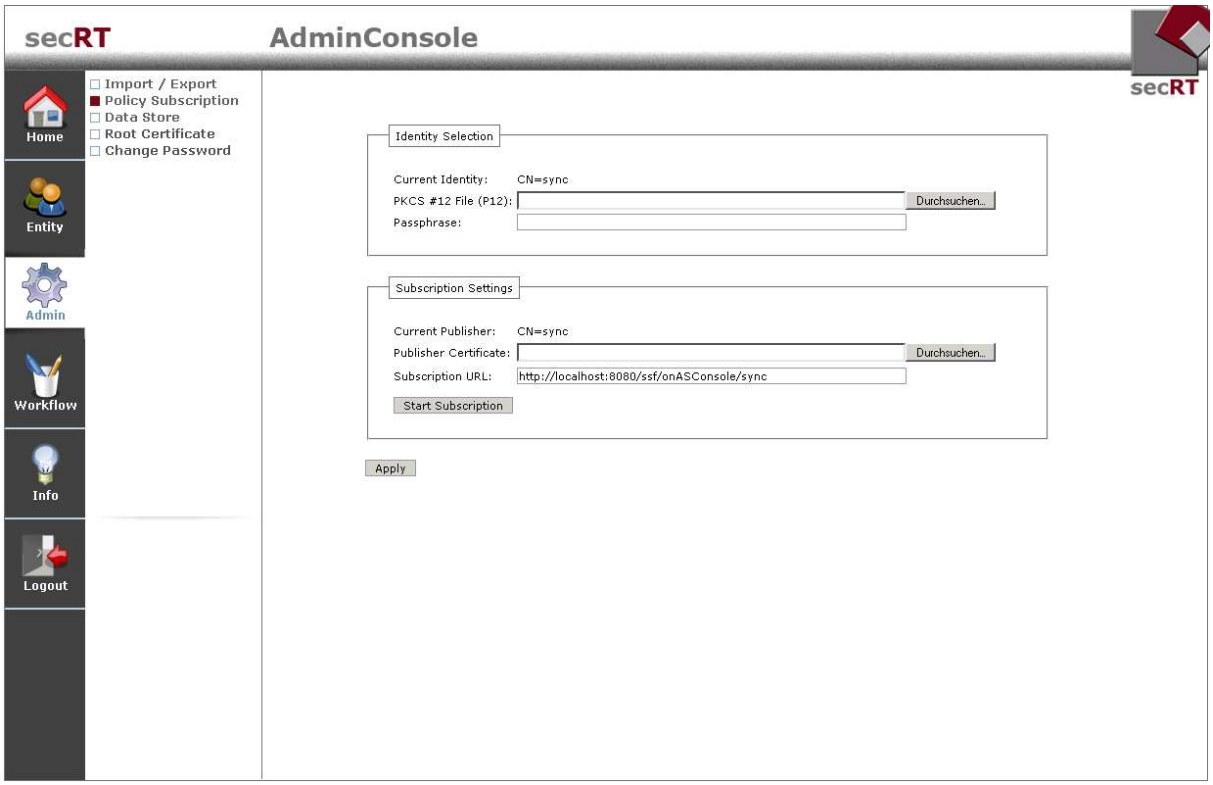
4.4.1.3 Userroles

Username	User ID of the user
Role Name	Name of the role
Timestamp	0

4.4.2 Policy Subscription

If several connectors are deployed, they will synchronize with one another. For configuration of synchronization, click the menu item Policy Subscription.

The configuration page appears:



Here you can configure encryption data for the communication required for the synchronization. Each connector needs an identity imported via a PKCS#12 container. The configuration varies depending on the connector role.

4.4.2.1 Publisher Configuration

The Publisher provides the information and bears an identity which has to be configured. The fields **Publisher Certificate** and **Subscription URL** are not relevant in this case. Additionally, each subscriber has to be available as activated and initialized user. For configuration of the publisher please act as follows:

1. Create a user
2. Initialize the user
3. Download the Keystore and the certificate
4. Load the Keystore as **Current Identity** under **Publish / Subscribe**.
5. Accept the configuration with **Apply.**

4.4.2.2 Subscribers Configuration

The subscriber has to be an initialized and active user in the Publisher's user management, as the Publisher does only communicate with those secRTs. For configuration of the subscriber, please act as follows:

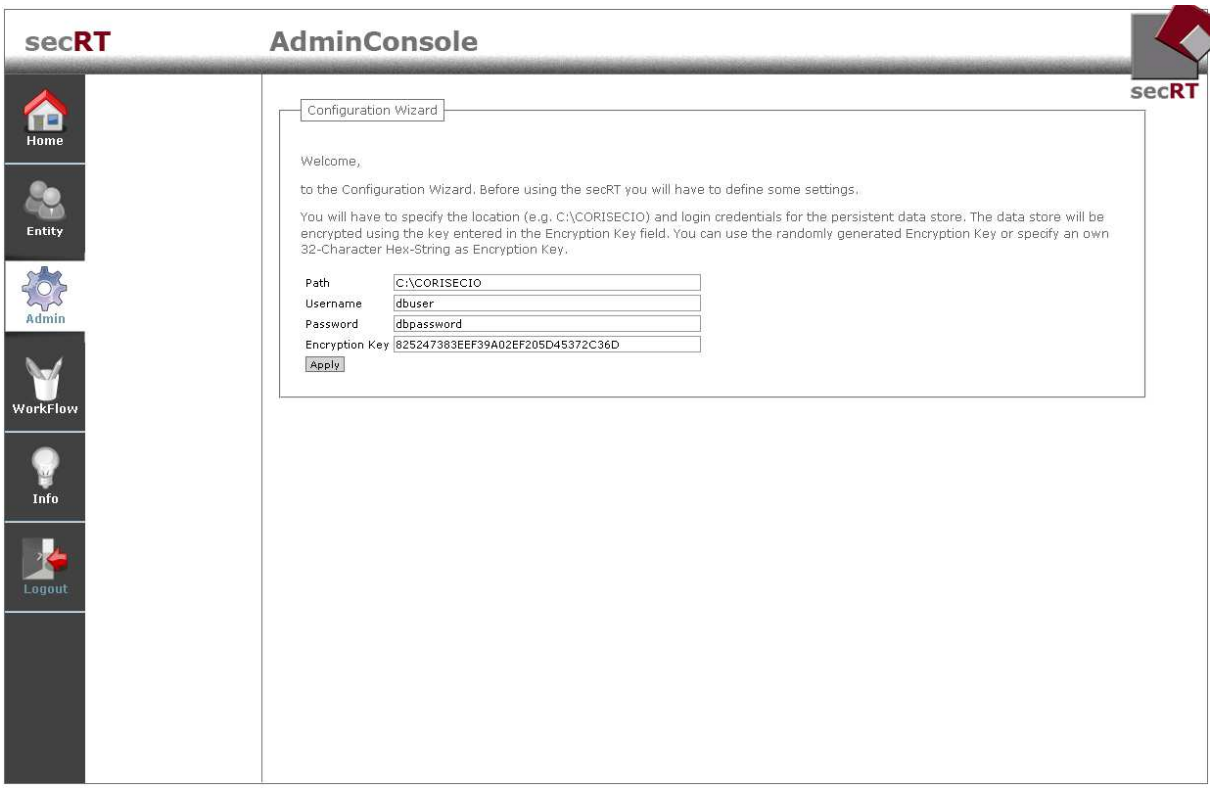
1. Open the Publisher's AdminConsole
2. Create a user and initialize it
3. Download the Keystore
4. Change over to **Policy Subscription**
5. Import the Keystore as **Current Identity**
6. Enter at **Subscription URL** the publisher URL according to the following scheme: http://[host]:[port]/[path]/onASConsole/sync
7. Import the certificate of the Publisher identity at **Publisher Certificate**.
8. Configure the subscriber with **Apply.**
9. Download the certificate of the user created in step 2.
10. Log-in to the AdminConsole of the Publisher
11. Create a user and initialize him with the certificate downloaded in step 9.

4.4.2.3 Synchronization

For Synchronization, click **Start Subscription**. Thereafter, the entities users, roles, user role-assignment and workflows are actualized in succession. During this process, new entities are added and existing entities are actualized. The entities will not be deleted in the Subscriber. When the active workflow is updated, it will be reactivated afterwards.

4.4.3 Data Store

Before the connector can be used, the database connection has to be configured by clicking **Data Store** in the menu. When logging in for the first time at the AdminConsole of a connector without Security Repository setup, you will be directed to this page automatically.



At **Path** enter the absolute path to the required data base directory. This path can be chosen freely and indicates where the database files will be stored. Please note that you must have full access rights to this directory.

When choosing a path in a connector, like e.g. C:/apache-tomcat-5.5.29/webapps/connector/adapter/derby, the data will be overwritten for a new version of the connector. In this case, you should perform a backup in advance or store the data outside the connector.

Enter the name of the database user under **Username** and his **Password** under password. Such kind of user is created this way. The field **Encryption Key** is pre-set with a random Key. Adjust it when necessary.

Please kindly note that the entered information should be stored separately as copy to make sure that you still will have access to your data after accidental deletion. Usually updates are provided as Web Archive; please kindly note that the database should be outside the web application directory, to be able to reuse it after an update.

In case of a re-import of the file repository.bsr the Application Server has to be started anew.

Accept your changes with **Apply**. If all information is correct, the repository will be configured. In case of an error, you will receive a notification and you will have to check the Log file.

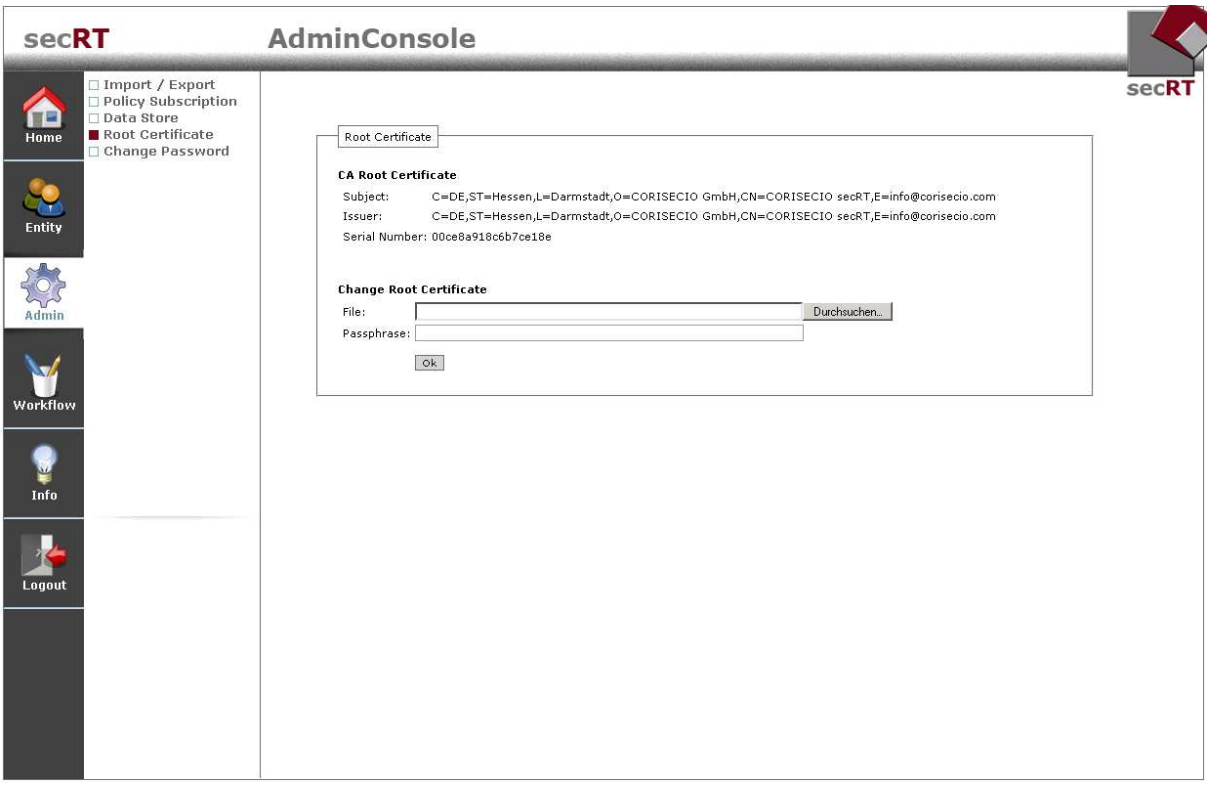
If the configuration has been done successfully once, a renewed configuration of the Data Store is only possible by deleting the configuration file (repository.bsr) manually.

Please note that when using two or more connectors, they cannot be connected to the same Security Repository.

4.4.4 Root Certificate

You may change the issuer certificate for the integrated CA. All newly issued certificates will be stored in the current root certificate.

Click **Root Certificate** in the menu. The current issuer certificate is displayed.



In the section **Change Root Certificate**, you can load an issuer certificate contained in a PKCS#12 container to replace the existing one. Therefore, enter the password for the PKCS#12.

4.4.5 Change Password

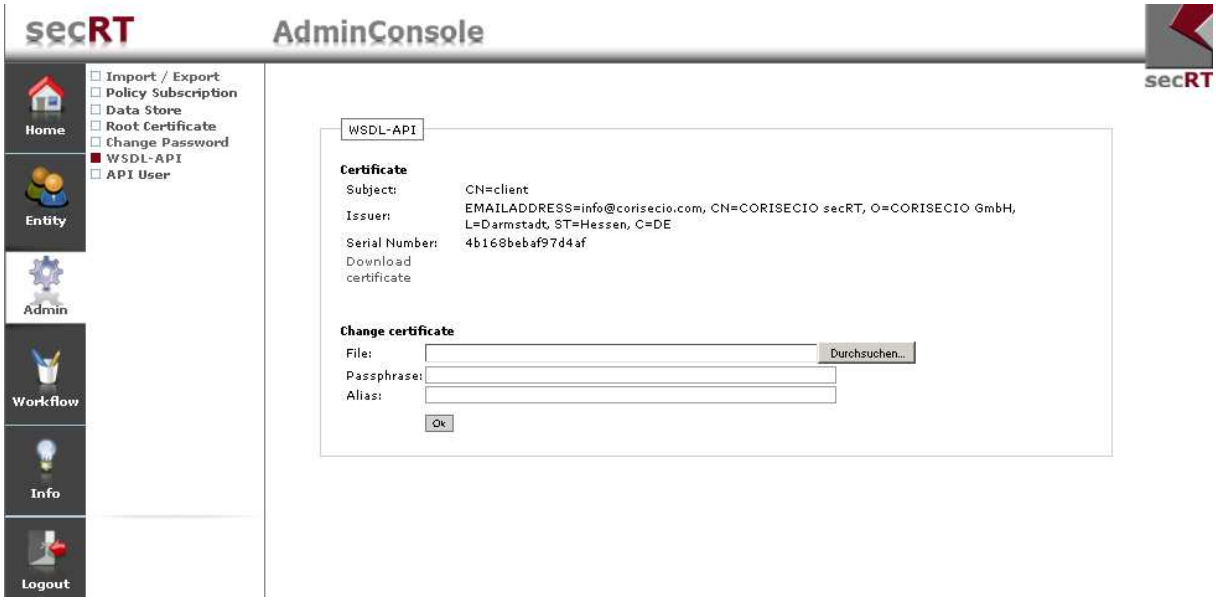
With **Change Password** you can change the access data to the AdminConsole. Please note that the password change is only effective locally. You may reset the password by re-loading the Security Rules.

4.4.6 WSDL-API

The communication settings for the WSDL-API can be configured under WSDL-API in the menu.

The sole configuration possibility is the keypair, which is used to encrypt the requests and to decrypt on server side, respectively sign. In order to enable a successful communication, a keypair has to be configured! Therefore, you can load a PKCS#12 keypair.

Enter the password for the key store and an alias as an option for access to the key. In case that you do not enter an alias, the first key from the key store is used. Click **Ok** to load the PKCS#12 file.



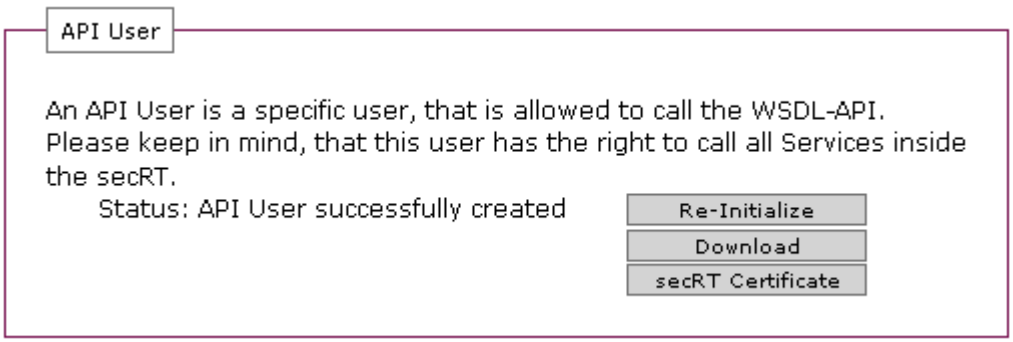
You can download the certificate – if configured – via the **Download Certificate** link.

Please note also section 4.4.7 for interface usage.

4.4.7 API User

To be able to use the WSDL-API, it needs an authorized user. With a click on **Create & Initialize**, a user with the right to use the WSDL-API can be created.

Initialization of the API user is conducted analogous to initialization of other users (see 4.3.1.5). The API user, however, has to have a private key, which means that a key set will have to be imported in case that you choose the import option. After initialization, the following page is displayed:



The API-user Keystore can be loaded with a click of the **Download** button. A password can be entered. The key store is loaded as a PKCS#12 file.

To enable use of the WSDL-API, it also needs the secRT certificate, which can be loaded through the **secRT Certificate** button.

A re-initialization of the API users is possible, as well. Therefore, click **Re-Initialize**.

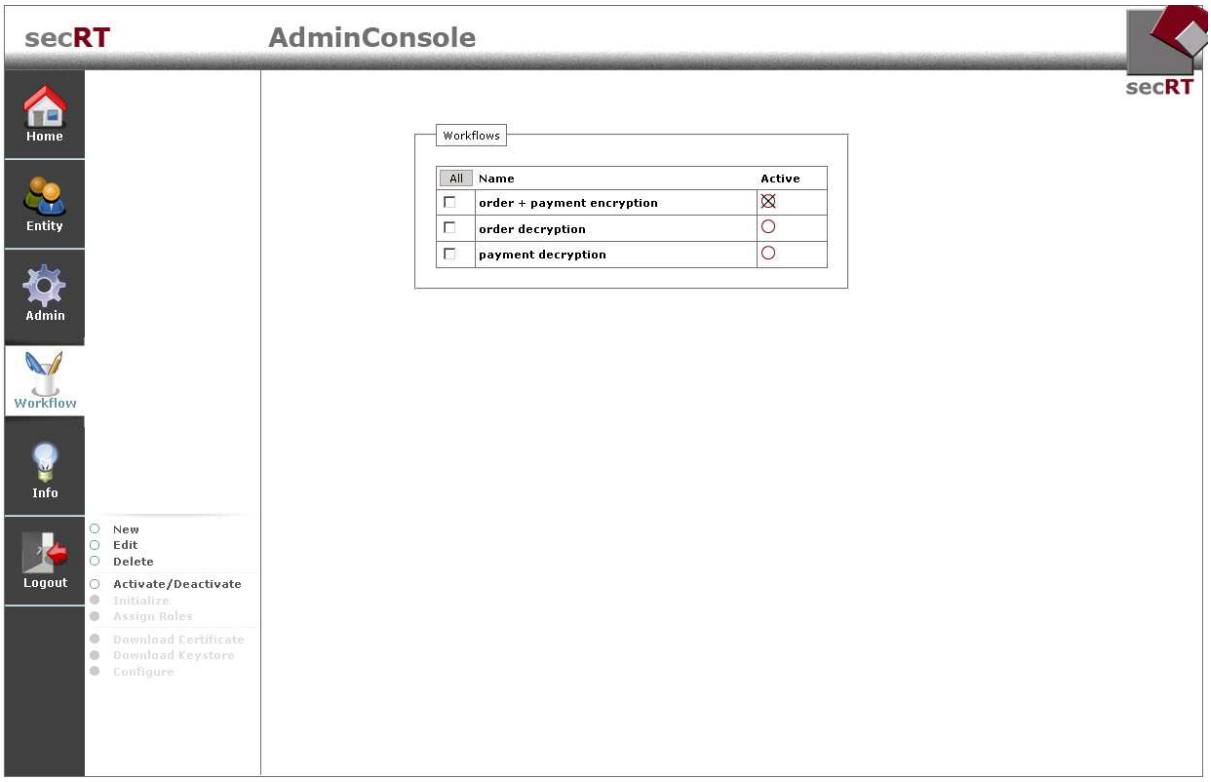
SOAP messages that are sent to the WSDL-API have to be signed with the private key of the API user at first. Thereafter, they have to be encrypted with the secRT certificate. The replies from the secRT are signed with the keys configured for the WSDL-API (see 4.4.6) and encrypted with the certificate of the API user.

4.5 Workflow

Via Workflow Manager, resp. Workflow Editor, the secRT provides the option to arrange the security functions available via adapter in a process logic and to configure them (workflows). This achieves realization of security functions for the Workflow Engine without programming effort. To call up the Workflow Manager, click on the respective menu item.

4.5.1 Overview

It is possible to manage several workflows. Therefore, the overview page Workflow Manager can be used. The structure is as follows:



In this example three workflows are already configured. They are listed with their names.

Only one workflow can be active at a time, i.e. a relevant service can be carried out.

The actions on the overview page in detail are:

4.5.1.1 New

A new workflow is created with a click on the **New** button. Enter the name of the workflow to be created in the text field and click the **Ok b**utton.

The name has to be unique and has to consist of 1-60 characters.

4.5.1.2 Edit

With the Edit button, the selected workflow can be edited, i.e. the exact process sequence is configured. This is conducted through the Workflow Editor (see **Fehler! Verweisquelle konnte nicht gefunden werden.**).

4.5.1.3 Delete

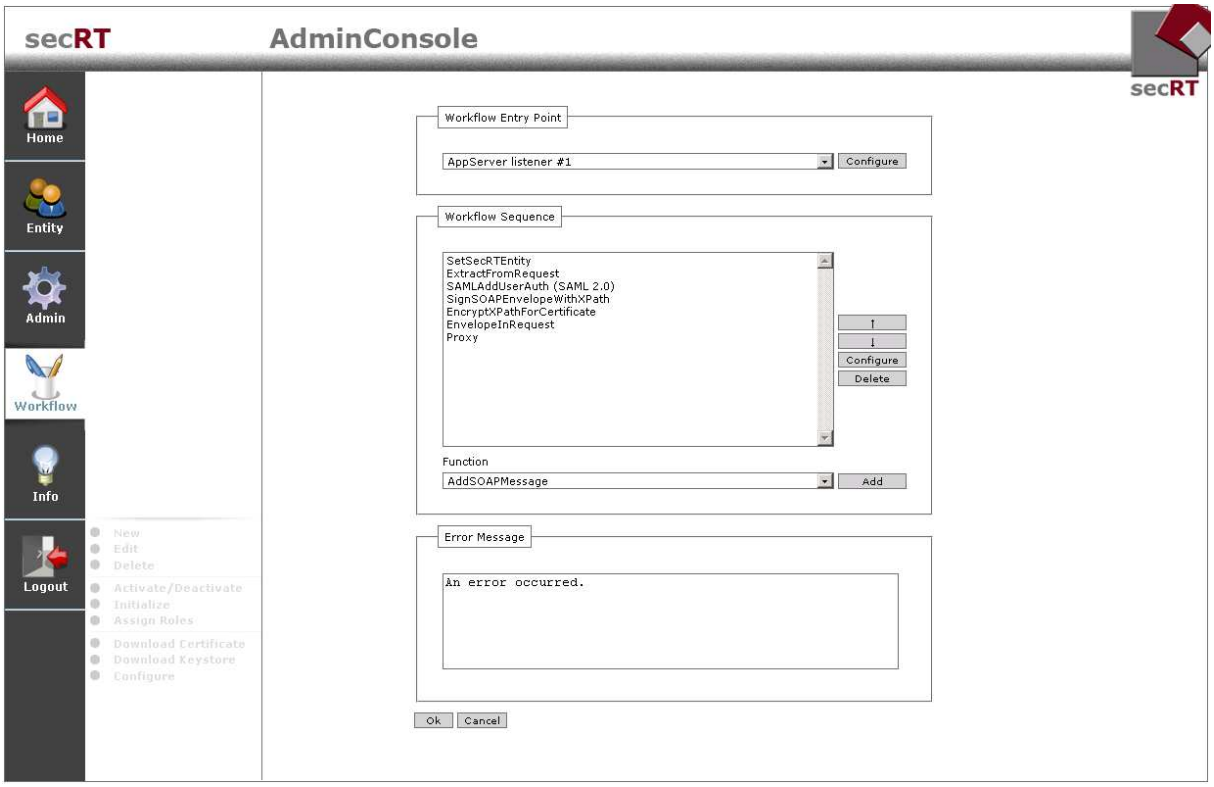
Via this button, the selected workflow is deleted. It is not possible to delete the active workflow. At trying, a respective error message comes up.

4.5.1.4 Activate / Deactivate

Via this button, the selected workflow is activated. The functions configured for the workflow are initialized and executed. The workflow previously activated is set inactive, the respective processes are stopped. If an already activated workflow is reactivated, this is like a restart of the services (probably with modified configuration).

4.5.2 Workflow Editor

Via the Workflow Editor, you can define linear workflows. You reach this page by clicking the edit button for a workflow.



The main page is divided in the sections **Workflow Entry Point** (Listener configuration see 4.5.2.1), **Workflow Sequence** (configuration of the function list see 4.5.2.2) and **Error Message** (Error page, see **Fehler! Verweisquelle konnte nicht gefunden werden.**). The image shows a workflow already configured.

4.5.2.1 Configuring of functions resp. of the listener

For configuration of a service, you need a Listener receiving the service requests, as well as functions executing the processing of the requests.

The Listener is set up via the dropdown menu in the section **Workflow Entry Point**; functions may be added or deleted via the **Add** resp. **Delete** button.

The function added to a workflow via + is selected via the selection list under **Workflow Sequence**.

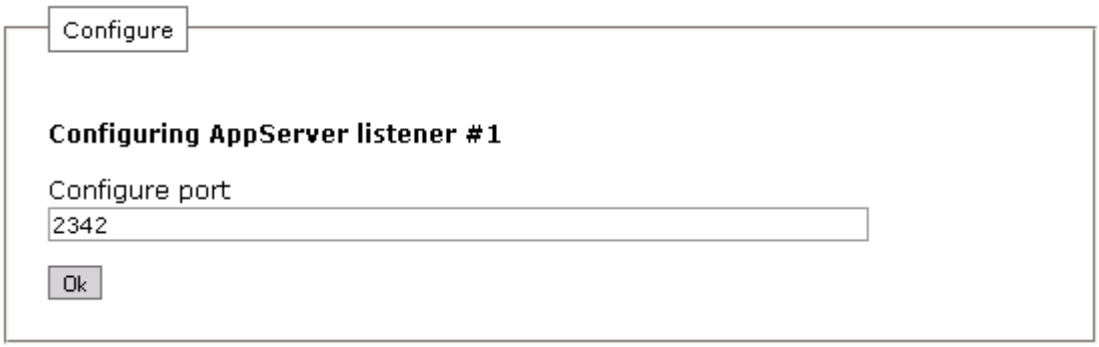
To delete a function you have to select it in the list at first.

This also applies to modification of the functions sequence. These may be modified via ↑ resp. ↓.

The Workflow Editor automatically reads the available Listener and functions at start of the web application. They are the result of the available adapters.

Functions resp. Listeners usually require specific configuration parameters. These can be reached via the **Configure** button. It exists twice, for Listener and functions respectively. To configure a function, it has to be selected from the function list at first.

After click on Configure, a page appears where the required parameters may be entered, just like e.g. upon configuration of the **AppServer Listener #1**:



Enter the required parameter values on the page and save the entries via **Save**. After saving the data, you should return to the Workflow Editor page.

Note:
An input validation on the configuration pages of the Listeners resp. functions is not done. If you would like to abolish entries, you can use the Back button of your browser.

Please note that with click on Save the workflow configuration change is not made persistent yet. This is only done via click on **Ok** on the Editor main page (also see 4.5.2.4).

Changes can be abolished via **Cancel**. You will return to the overview page of the Workflow Manager.

Note:
The relevance of the functions' and listener's configuration parameters is described in the Reference Guides.

Configuration of Keystores resp. certificates

Certain functions resp. Listeners require the configuration of special objects like e.g. keystores or certificates.

The keystore configuration is conducted via a user selection list. The user selection list consists of users being initialized and activated, as well as of an empty entry provided that the configuration is neither required nor needed. A key set is always assigned to initialized users.

No further action for configuration of a keystore is required but selecting of a user key.

Configuration of value lists

Some configuration parameters of a function resp. a Listener allow entering a list of several values.

Example: Proxy function.

Add the proxy function on the Workflow Editor main page and click the respective **Configure** button.

While the proxy function provides plain text fields for value entry, it also offers the option to configure lists (e.g. `RewritingTypes`).

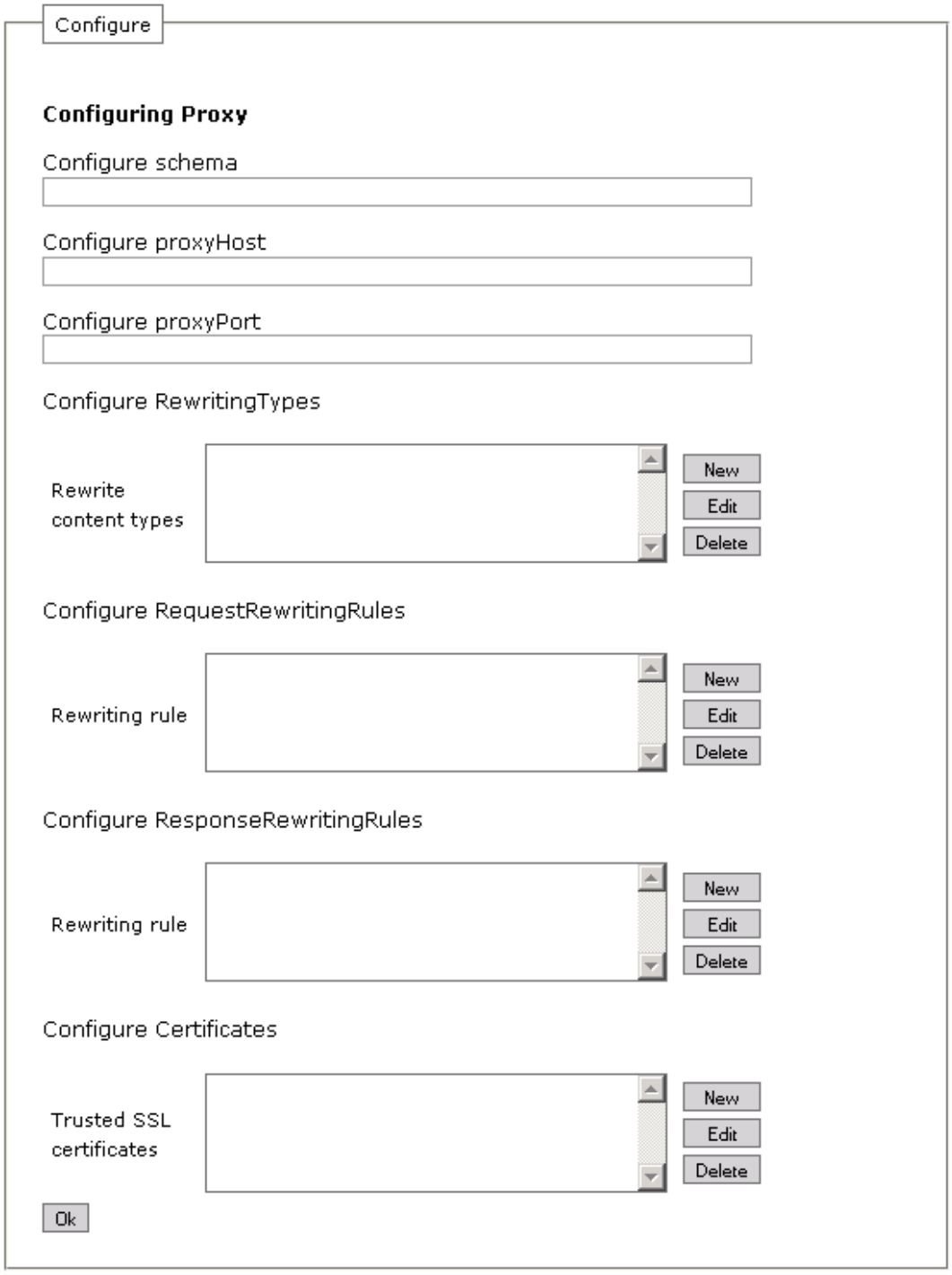


Image 1 Configuration page of the proxy function (details)

Like keystores and certificates, lists are not editable directly, too.

At click on **New**, a new page appears in the browser allowing entry configurations. Example:

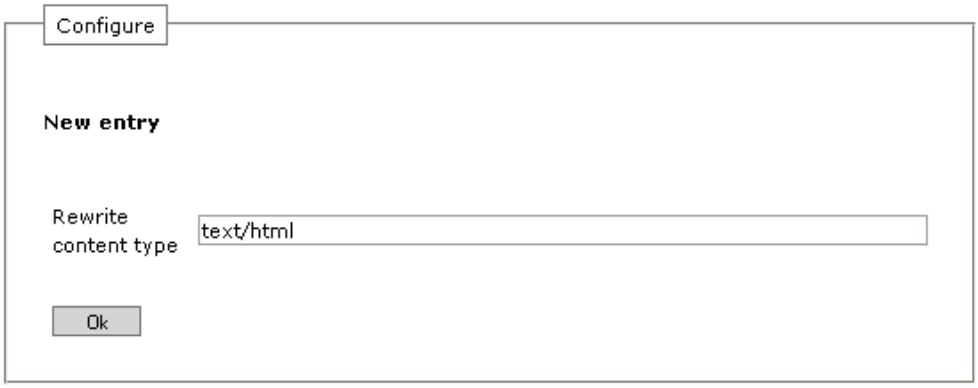


Image 2 Configuration page for a list entry

After clicking **Save**, the new list entry will be saved and you will return to the previous page. This way, you can configure as many entries as you like.

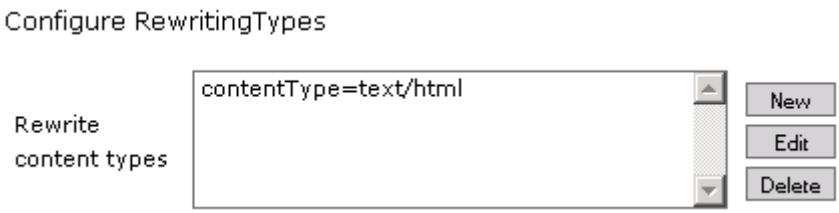


Image 3 Value list with several entries

Via the buttons **New**, **Edit** and **Delete**, you can create new entries resp. edit or delete existing ones.

4.5.2.2 Sequence Control

The processing sequence of the functions through the Workflow Engine is executed according to the sequence defined by you in the Workflow Editor. It can be changed with the buttons ↑ resp. ↓. Processing of an incoming request may be canceled prematurely because of arising errors or influencing conditions. This follows the result values of a function stated in the reference guide. There are three possible values:

1. **OK**: the function has been terminated as expected, the processing continues with the next function on the list.
2. **Abort**: the function was terminated as expected, but the processing of the request stops.
3. **Error**: the function was terminated with an error, the processing stops and the request is replied with an error page (see 4.5.2.3).

Not every function can return all result values.

Which result values of a certain function are possible, can be seen in the Reference Guide.

4.5.2.3 Error Page

If an unexpected error occurs during execution of a function, this can be communicated to the user with display of an error message. This could be seen when using the configured service via his web browser e.g.

The error page is configured via the main page of the Workflow Editor under **Error Message**. Input of HTML code is possible here.

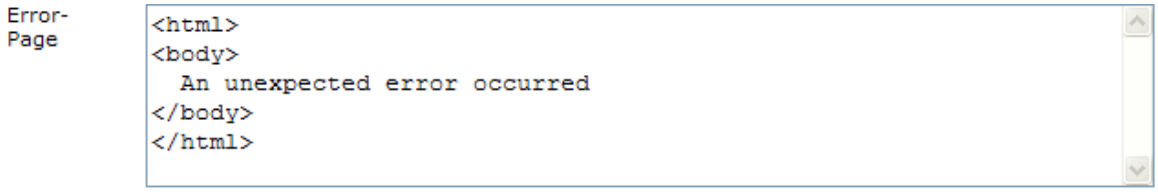


Image 4 Configured error page

4.5.2.4 Saving and using the configuration

The configured service is saved via **Ok** on the Editor main page. Then you will return to the Workflow Manager overview page.

A change of the actively executed workflow will only be effective if this workflow is activated again.

Please consider that the Workflow Editor is not verifying the configured parameters. Therefore, please test your configuration.

4.5.2.5 Testing the configuration

Enter the following URL in your browser: http:// [host]:[port]. Replace [port] with the TCP/IP Port you have stated at the Listener configuration.

4.6 Info

Under menu item Info, information about the secRT is displayed.

4.6.1 Product Info

This page provides information about the product.

4.6.2 Services

On this page the started bundles and the registered OSGi services are shown.

4.6.3 Help

On this page you will find a help text.

4.7 Logout

Via logout you will leave the administration interface of the AdminConsole. Meanwhile, the web application continues. Do not forget to leave the web application via **Logout** to prevent unauthorized use of the AdminConsole.

4.8 Logging

The log files of the security connector are located in the AdminConsole's directory structure in the log folder. To open the log file you will have to open the folder log. In this folder, there is a file named connector.[YYYY]-[MM]-[TT].log. Here, [YYYY] means the year, [MM] the month and [TT] the day the log file was created. To open the log file you will have to close the application.

5 WSDL-API

In order to use the WSDL-API, it needs a PKCS#12 keystore with the administrator´s private key and the X.509 Certificate of the security broker. At first, the SOAP messages have to be signed with administrator´s private key and then they have to be encrypted with the security broker´s certificate.

5.1 Recall of WSDL of Services

For a deployed secRT, the WSDL of the available services can be recalled under http://[HOST]:[PORT]/[CONTEXTPATH]/wsdl.

5.2 Communication

A Service Binding is found in the displayed WSDL, where you will find the URL for the service recalls. It is named: http://[Host]:[Port]/[Contextpath]/broker/service. Communication is protected by means of XML signature (http://www.w3.org/TR/xmldsig-core/) and by XML encryption (http://www.w3.org/TR/xmlenc-core/).

Please note:

- The signature has to be created through the body of the SOAP message before encryption is made.
- The certificate of the accessing administrator (in `test.p12`) has to be included in the key info.
- The body element is encrypted with the public key of the addressee (broker.crt).

5.2.1 Example

5.2.1.1 Request:

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope soap:actor="http://schemas.xmlsoap.org/soap/actor/next" soap: mustUnderstand="1" xmlns:SOAP-SEC="http://schemas.xmlsoap.org/soap/ security/2000-12" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd= "http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/ XMLSchema-instance">
<soap:Header id="Header"><SOAP-SEC:Signature soap:actor="" soap:must Understand="0">
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo>

<ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
<ds:Reference URI="#Body">
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>1J6Zfx93EUD70VF7S5hvjetBpeQ=</ds:DigestValue>
</ds:Reference>
<ds:Reference URI="#Header">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>DF4afbQu65pv7AkM7Ky50s6/B0k=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>
L4H90lKqAEoqlVuw1dRAlHySDv+0hWUdBBQb5mOfp+p6XyEEV+3jCj9za5JO/ewkp
YsJGrcGfNc+
hcVzLrl-
hfdI/fe3bcdNmR/8Y35dzU+so9XbF/Tn/6AVHuhBGWMAX54MXN4i0JjV6ZzstLk96pB
6y
kwmr7HNOLg8TmTTF+/U=
</ds:SignatureValue>
<ds:KeyInfo>
<ds:X509Data>
<ds:X509Certificate>
…
</ds:X509Certificate>
</ds:X509Data>
<ds:KeyValue>
<ds:RSAKeyValue>
<ds:Modulus>
…
</ds:Modulus>
<ds:Exponent>AQAB</ds:Exponent>
</ds:RSAKeyValue>
</ds:KeyValue>
</ds:KeyInfo>
</ds:Signature></SOAP-SEC:Signature></soap:Header><soap:Body id="Body"><xenc:EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Content" xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"><xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes256-cbc" xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"/><ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">

<xenc:EncryptedKey
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"><xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"/><xenc:CipherData
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"><xenc:CipherValue
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">0qS2il1MDBthy8qCt2zmjBQh25a
809Ae0FexhhYTK0uBMhFOez052Sc4QfOywguLdqvCltl6KTO6
7cIfqZyeu2UbahWjwc3I2itSenre1dGvdGu14Y8q0rYmcQfWgtBG/kgmqnXjcfdMYqw
W5gPbyzw5
2Ad1dQsXGlkInxdYNgA=</xenc:CipherValue></xenc:CipherData></xenc:EncryptedKey></ds:KeyInfo><xenc:CipherData
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"><xenc:CipherValue
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
…
</xenc:CipherValue></xenc:CipherData></xenc:EncryptedData></soap:Body></soap:Envelope>

5.2.1.2 Reponse

<soap:Envelope soap:actor="http://schemas.xmlsoap.org/soap/actor/next" soap:mustUnderstand="1" xmlns:SOAP-SEC="http://schemas.xmlsoap.org/soap/ security/2000-12" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns: xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/ XMLSchema-instance">
  <soap:Header id="Header">
    <SOAP-SEC:Signature soap:actor="" soap:mustUnderstand="0">
      <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:SignedInfo>
          <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
          <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
          <ds:Reference URI="#Body">
            <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig# sha1"/>
            <ds:DigestValue>3JKOf2FwtwtQV1s+jdGXlQjHYjk=</ds:DigestValue>
          </ds:Reference>
          <ds:Reference URI="#Header">
            <ds:Transforms>
              <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig# enveloped-signature"/>
            </ds:Transforms>
            <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig# sha1"/>
            <ds:DigestValue>DF4afbQu65pv7AkM7Ky50s6/B0k=</ds:DigestValue>
          </ds:Reference>

```
        </ds:SignedInfo>

<ds:SignatureValue>MTq2fwCnVPRJ9J6TxIafvTvCi4Jfxqxm2UDb6xur10KlomUDVrz
lbREgUqO0eq0cDhKyBElLai+r
N9Fvdcnwwxd+U1Ff6artbHmK5dDfdOchCrBLSGICRax9hqsKhdf+SjMagi3QW8exAq
iHMKZyLKD2
J6q/aRJQ/0TGCBFyD64=</ds:SignatureValue>
        <ds:KeyInfo>
          <ds:X509Data>
            <ds:X509Certificate>
                …
            </ds:X509Certificate>
          </ds:X509Data>
          <ds:KeyValue>
            <ds:RSAKeyValue>
              <ds:Modulus>…</ds:Modulus>
              <ds:Exponent>AQAB</ds:Exponent>
            </ds:RSAKeyValue>
          </ds:KeyValue>
        </ds:KeyInfo>
      </ds:Signature>
    </SOAP-SEC:Signature>
  </soap:Header>
  <soap:Body id="Body">
    <xenc:EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Content" xmlns:
xenc="http://www.w3.org/2001/04/xmlenc#">
      <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#
aes256-cbc"/>
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <xenc:EncryptedKey>
          <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#
rsa-1_5"/>
          <xenc:CipherData>
            <xenc:CipherValue>…</xenc:CipherValue>
          </xenc:CipherData>
        </xenc:EncryptedKey>
      </ds:KeyInfo>
      <xenc:CipherData>
        <xenc:CipherValue>
            …
        </xenc:CipherValue>
      </xenc:CipherData>
    </xenc:EncryptedData>
  </soap:Body>
</soap:Envelope>
```