User Management Interfaces for Earth Observation Services

Copyright notice
Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved. To obtain additional rights of use, visit http://www.opendata.org/legal/.

Warning

This document is not an OGC Implementation Specification. This IPR is not an official position of the OGC. Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type: OGC™ Interoperability Program Report
Document subtype: Candidate Implementation Specification
Document stage: Draft
Document language: English

This document does not represent a commitment to implement any portion of this specification in any company’s products.
NOTICE

Permission to use, copy, and distribute this document in any medium for any purpose and without fee or royalty is hereby granted, provided that you include the above list of copyright holders and the entire text of this NOTICE.

We request that authorship attribution be provided in any software, documents, or other items or products that you create pursuant to the implementation of the contents of this document, or any portion thereof.

No right to create modifications or derivatives of OGC documents is granted pursuant to this license. However, if additional requirements (as documented in the Copyright FAQ at http://www.opengeospatial.org/legal/ipr_faq.htm) are satisfied, the right to create modifications or derivatives is sometimes granted by the OGC to individuals complying with those requirements.

THIS DOCUMENT IS PROVIDED “AS IS,” AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE DOCUMENT ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT OR THE PERFORMANCE OR IMPLEMENTATION OF THE CONTENTS THEREOF.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to this document or its contents without specific, written prior permission. Title to copyright in this document will at all times remain with copyright holders.

RESTRICTED RIGHTS LEGEND. Use, duplication, or disclosure by government is subject to restrictions as set forth in subdivision ©(1)(ii) of the Right in Technical Data and Computer Software Clause at DFARS 252.227.7013

OpenGIS®, OGC™ OpenGeospatial™ and OpenLS ® are trademarks or registered trademarks of Open Geospatial Consortium, Inc. in the United States and in other countries.
Contents
1 SCOPE ................................................................................................................................... 8
2 CONFORMANCE ....................................................................................................................... 8
3 REFERENCES ............................................................................................................................ 8
  3.1 NORMATIVE REFERENCES ............................................................................................... 8
  3.2 OTHER REFERENCES ......................................................................................................... 9
4 TERMS AND DEFINITIONS ......................................................................................................... 10
5 SYMBOLS AND ABBREVIATIONS ............................................................................................ 12
  5.1 SYMBOLS (AND ABBREVIATED TERMS) .......................................................................... 12
  5.2 DOCUMENT TERMS AND DEFINITIONS ........................................................................... 13
6 SYSTEM CONTEXT ...................................................................................................................... 13
  6.1 APPLICATION DOMAIN .................................................................................................... 13
  6.2 PROTOCOL BINDING ........................................................................................................ 13
  6.3 LIBRARIES ....................................................................................................................... 13
  6.4 BASIC USE CASES ............................................................................................................ 13
  6.5 SECURITY MODEL .............................................................................................................. 15
    6.5.1 Encryption ................................................................................................................... 15
    6.5.2 Message Digest ............................................................................................................ 17
    6.5.3 Authentication Use Cases ........................................................................................... 20
      6.5.3.1 Default Case - Local Entity is IdP ........................................................................ 20
      6.5.3.2 External Entity is IdP ............................................................................................ 21
    6.5.4 Authorisation Request .................................................................................................. 22
    6.5.5 OASIS SAML ................................................................................................................ 23
    6.5.6 OASIS Ws-Security ..................................................................................................... 24
      6.5.6.1 Encryption .............................................................................................................. 24
      6.5.6.2 Signature ................................................................................................................ 25
7 INTERFACE .................................................................................................................................. 25
  7.1 AUTHENTICATE .................................................................................................................. 25
    7.1.1 Request ........................................................................................................................ 25
    7.1.2 XML encoding ............................................................................................................. 26
    7.1.3 Response ...................................................................................................................... 26
      7.1.3.1 Example Authentication Response Before Encryption ..................................... 28
    7.1.4 Failed Authentication Request ................................................................................... 30
    7.1.5 WSDL .......................................................................................................................... 30
  7.2 SERVICE REQUEST .............................................................................................................. 32
    7.2.1 Request ........................................................................................................................ 32
    7.2.2 XML encoding ............................................................................................................. 32
    7.2.3 Failed Request .............................................................................................................. 36
  7.3 SERVICE RESPONSE .......................................................................................................... 36
    7.3.1 Synchronous ................................................................................................................. 36
    7.3.2 Use Case: User logs in at client and makes Synchronous Service Request to HM Service 36
    7.3.3 Asynchronous .............................................................................................................. 38

Figures
Figure 1 User Management Use Cases ..................................................................................... 14
Figure 2 Federating (Local) Entity is request designated IdP (Default Case) .......................... 21
Figure 3 External Entity is request designated IdP .................................................................. 22
Figure 4 Digital Signature ......................................................................................................... 25
Figure 5: Example Authenticate Request .............................................................................. 26
Date: June 30, 20

Figure 6: Example Response

Figure 7: Authentication Service WSDL

Figure 8: Service Request Example

Figure 9 DAIL Sequence Diagram Showing Client Request with Synchronous Response

Figure 10 Sequence Diagram Showing Asynchronous Request
i. Preface

This document explains how user and identity management information is included in the protocol specifications for EO (Earth Observation) services for catalogue access (OGC 06-131), ordering (OGC 06-141) and programming (OGC 07-018) in the EO DAIL and HMA operational interfaces.

The document was initially produced during the ESA HMA (Heterogeneous Missions Accessibility) project and refined during the FEDEO (Federated Earth Observation) Pilot. It is further refined in the ESA EODAIL Implementation project.

This document is not a new specification, however, it describes how existing specifications from W3C and OASIS can be used in combination to pass identity information to Web services some of which are based on OGC Best Practice specifications.

ii. Submitting organisations

The following organisations will submit the original document or its revisions to the OGC™ Security Working Group.

- Spacebel s.a.
- ESA – European Space Agency
- Oracle

The editors would like to acknowledge that this work is the result of collaboration and review of many organisations and would like to thank for the comments and contributions from:

- Astrium
- Spot Image
- ASI
- CNES
- DLR
- Eumetsat
- EUSC
- MDA

Note: this does not imply a complete endorsement from these organisations.
iii. Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

<table>
<thead>
<tr>
<th>Contact</th>
<th>Organisation</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rowena Smillie</td>
<td>Spacebel</td>
<td><a href="mailto:Rowena.Smillie@spacebel.be">Rowena.Smillie@spacebel.be</a></td>
</tr>
<tr>
<td>Alexandre Cucumel</td>
<td>Spacebel</td>
<td><a href="mailto:Alexandre.Cucumel@spacebel.be">Alexandre.Cucumel@spacebel.be</a></td>
</tr>
<tr>
<td>Wouter Van de Weghe</td>
<td>Oracle</td>
<td><a href="mailto:wouter.van.de.weghe@oracle.com">wouter.van.de.weghe@oracle.com</a></td>
</tr>
</tbody>
</table>

iv. Revision history

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Editor</th>
<th>Sections modified</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15 September 2007</td>
<td>0.0.1</td>
<td>R.Smillie</td>
<td>All</td>
<td>initialised draft document.</td>
</tr>
<tr>
<td>23 April 2008</td>
<td>0.0.2</td>
<td>R.Smillie</td>
<td></td>
<td>updated in line with EO DAIL implementation project</td>
</tr>
<tr>
<td>07 Feb 2009</td>
<td>0.0.3</td>
<td>R.Smillie</td>
<td></td>
<td>updated in line with EO DAIL implementation project</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>soap version changed to 1.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>authentication request does not use WS-Security</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>message examples added</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>encryption and signature descriptions improved</td>
</tr>
</tbody>
</table>
v. Foreword

This document, through its implementation profile, references several external standards and specifications as dependencies:


3. WSDL, Web Services Description Language (WSDL) 1.1, http://www.w3.org/TR/wsd1


Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open GIS Consortium, Inc. shall not be held responsible for identifying any or all such patent rights.
Introduction

This specification is complementary to a set of specifications that describe services for managing Earth Observation (EO) data products. These services include collection level, and product level catalogues, online-ordering for existing and future products, on-line access etc. and are put into context in an overall document (see HMA Architecture Technical Note [NR13]).

The intent of this specification is to describe a federated identity management interface that can be supported by many data providers (satellite operators, data distributors …), most of whom have existing (and relatively complex) facilities for the management of their data and users. The strategy is to specify a platform and provider independent interface using existing standards.

1 Scope

This proposed interface document describes the interfaces required to authenticate and authorise users in a federated system of Earth Observation services.

2 Conformance

This will be the subject of future work. In particular the extension of the CITE compliance tests for catalogue, ordering and programming to also check compliance to the current interfaces may be considered in future work.

3 References

3.1 Normative references


[NR7] WSDL, Web Services Description Language (WSDL) 1.1, http://www.w3.org/TR/wSDL

[NR8] IETF RFC 2119, Keywords for use in RFCs to Indicate Requirement Levels, http://rfc.net/rfc2119.html
<table>
<thead>
<tr>
<th>Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NR12</td>
<td>Secure Hash Standards (SHA-1) National Institute of Standards and Technology <a href="http://csrc.nist.gov/cryptval/shs.htm">cryptval/shs.htm</a></td>
</tr>
<tr>
<td>NR13</td>
<td>HMA Architectural Design Technical Note version 1.6, 13/06/2007 <a href="http://services.eoportal.org/portal/system/HelpUI.jsp">HelpUI.jsp</a></td>
</tr>
<tr>
<td>NR14</td>
<td>Glossary for the OASIS Security Assertion Markup Language (SAML) V2.0 <a href="http://docs.oasis-open.org/security/saml/v2.0/saml-glossary-2.0-os.pdf">saml-glossary-2.0-os.pdf</a></td>
</tr>
<tr>
<td>NR15</td>
<td>Java Cryptography Architecture API Specification &amp; Reference <a href="http://java.sun.com/j2se/1.5.0/docs/guide/security/CryptoSpec.html">CryptoSpec.html</a></td>
</tr>
<tr>
<td>NR16</td>
<td>OGC 04-016r5, OWS Common Implementation Specification 2004/12/17</td>
</tr>
<tr>
<td>NR17</td>
<td>XML encryption <a href="http://www.w3.org/TR/xmlenc-core/">xmlenc-core/</a></td>
</tr>
<tr>
<td>NR18</td>
<td>XML signature <a href="http://www.w3.org/TR/xmldsig-core/">xmldsig-core/</a></td>
</tr>
</tbody>
</table>

### 3.2 Other references

<table>
<thead>
<tr>
<th>Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OR1</td>
<td>HMA Operational Scenarios Technical Note HMA-TN-ASU-SY-0001 <a href="http://services.eoportal.org/portal/system/HelpUI.jsp">HelpUI.jsp</a></td>
</tr>
<tr>
<td>OR2</td>
<td>HMA-TN-SIE-EN-003 GMES Minimum User Profile 1.0c</td>
</tr>
<tr>
<td>OR3</td>
<td>EO-DAIL User Management Specifications DAIL-RS-ASU-EN-0002 1.1</td>
</tr>
<tr>
<td>OR4</td>
<td>SOAP Version 1.2 <a href="http://www.w3.org/TR/soap12-part1/">soap12-part1/</a></td>
</tr>
</tbody>
</table>
4 Terms and definitions

For the purposes of this document, the following terms and definitions apply:

4.1. Authentication [NR14]
To confirm a system entity’s asserted principal identity with a specified, or understood, level of confidence.

4.2. circle of trust
A federation of service providers and identity providers within which service providers accept the authentication asserted by the identity provider.

4.3. client
software component that can invoke an operation from a server

4.4. external entity
This is the entity external to the DAIL owning the protected web service. For the EO DAIL project it is the various ground segments that perform this activity. The external entity can be both an identity provider and service provider. There can be many external entities.

4.5. federated identity [NR14]
A principal's identity is said to be federated between a set of Providers when there is an agreement between the providers on a set of identifiers and/or attributes to use to refer to the Principal.

4.6. federating entity
This is the entity performing the federation of the identities. For the EO DAIL project it is the EO DAIL that performs this activity. The authentication request always passes through the federating entity. The federating entity can be both identity provider and service provider. There is only one federating entity.

4.7. identifier
a character string that may be composed of numbers and characters that is exchanged between the client and the server with respect to a specific identity of a resource

4.8. identity provider [NR14]
A kind of service provider that creates, maintains, and manages identity information for principals and provides principal authentication to other service providers within a federation, such as with web browser profiles.
4.9. interface
named set of operations that characterise the behaviour of an entity [ISO 19119]

4.10. operation
specification of a transformation or query that an object may be called to execute [ISO 19119]

4.11. parameter
variable whose name and value are included in an operation request or response

4.12. PEP
Policy enforcement point.

4.13. principal [NR14]
A system entity whose identity can be authenticated.

4.14. request
invocation of an operation by a client

4.15. response
result of an operation, returned from a server to a client

4.16. server
service instance
a particular instance of a service [ISO 19119]

4.17. service
distinct part of the functionality that is provided by an entity through interfaces [ISO 19119]
capability which a service provider entity makes available to a service user entity at the interface between those entities [ISO 19104 terms repository]

4.18. service interface
shared boundary between an automated system or human being and another automated system or human being [ISO 19101]

4.19. service provider [NR14]
A role donned by a system entity where the system entity provides services to principals or other system entities.
4.20. 
transfer protocol
common set of rules for defining interactions between distributed systems [ISO 19118]

5 Symbols and abbreviations

5.1 Symbols (and abbreviated terms)
Some frequently used abbreviated terms:

BPEL Business Process Execution Language
DAIL Data Access Integration Layer
EO Earth Observation
HMA Heterogeneous Missions Accessibility
HTTP HyperText Transport Protocol
IdP Identity Provider
ISO International Organisation for Standardisation
OGC Open GIS Consortium
SAML Security Assertion Markup Language
SOAP Simple Object Access Protocol
SP Service Provider
URI Uniform Resource Identifier
URL Uniform Resource Locator
URN Uniform Resource Name
WSDL Web Service Definition Language
W3C World Wide Web Consortium
XML eXtensible Markup Language
5.2 Document terms and definitions

This document uses the specification terms defined in Subclause 5.3 of [NR16].

6 System context

This section documents special requirements and describes the context of use.

6.1 Application domain

Web service requests are received by service providers (Ground segments). These service Providers should be able to identify who issued the request and react accordingly. The following approach is proposed:

1) An authentication Web service (accepting a user identifier password and optionally his identity provider) returns a SAML token which authenticates the user to the client (i.e. Web service consumer). (This authentication web service may federate the identity to another identity provider for authentication. At the interface context this is transparent, the federated identity request being identical to the initial request.)

2) Each subsequent service request by the client (Web service consumer) should include the SAML token in the SOAP header as described later in this document.

3) Each service provider accepts service requests only via a "policy enforcement point". The PEP decides based on the content of the message body, the contents of the message header (including authentication token) and the context (i.e. applicable policies) whether to accept or to refuse the service request or reroute it.

6.2 Protocol binding

To provide an overall coherent architecture within this context operations shall support the embedding of requests and responses in SOAP messages. Only SOAP messaging (via HTTP/POST or HTTPS/POST) with document/literal style shall be used. Messages should conform to SOAP 1.1 [NR6]. The message payload shall be in the body of the SOAP envelope. All authentication tokens shall be in the WS-Security element in the header of the SOAP envelope.

6.3 Libraries

The Santaurio Apache XML security Java library [NR19] has been used to implement the examples given from the DAIL implementation project.

6.4 Basic use cases

The use cases covered by this specification are shown in the following sequence diagram:

- Authentication: An authentication request is first made to the identity provider (IdP).
- Authorisation: A service request sent to the service provider (SP). This service request is a call of any of the operations defined in the catalogue (OGC 06-131), ordering (OGC 06-141) or programming (OGC 07-018) specifications but is not limited to these. The service requests can be synchronous or asynchronous via ws-addressing. This is transparent for the current specification.

A mission ground segment may be either an identity provider (IdP), a service provider (SP) or both IdP and SP.
This specification covers identity federation whereby the receiving IdP (federating entity), if not the IdP for the request, resolves the IdP and passes the authentication request to the correct IdP.

Authorisation requests (service requests) may address more than one ground segment, to perform so-called multi-mission requests, these requests are orchestrated by a BPEL workflow.

The policy enforcement on the SP is non invasive meaning that it is independent of the SP implementation.

**Figure 1 User Management Use Cases**

The high level use case for authentication and authorisation is shown in the above figure. Following sections of this document further elaborate the detail of the authentication and authorisation.

1. The authentication request is sent by the client to the authentication service which in the DAIL is directly exposed as a web service and does not pass through the PEP. However, if required a request could equally be intercepted by the IdP PEP and routed.

2. The client receives the authentication response containing the SAML token

3. The client then sends a service request i.e. an authorisation request. This request contains the SAML token.
6.5 **Security Model**

The model is based on WS-Security SAML token profile [NR11]. The authentication request contains the name and password identifying the user plus an optional definition of the designated identity provider.

User credentials are sent in SOAP over an encrypted channel i.e. HTTPS. The signed and encrypted SAML token is returned as SOAP over HTTPS. The client is unable to decrypt the content.

6.5.1 **Encryption**

Encryption of the SAML token is performed by the authentication service during an authentication request and response. Decryption is performed by the PEP during the authorization request. The encryption algorithm used is the AES-128 as defined in [NR15].

The encryption process is as follows:

- The authentication service first creates the symmetric key using The AES-128 encryption algorithm.
- This symmetric key is then itself encrypted with the public key of the IdP (i.e. GS) using the RSA encryption algorithm to create a secret key.
- The SAML token is then encrypted with the generated secret key using the AES-128 encryption algorithm.
- The message is then built.

**Example Authentication Request:**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
xmlns:q0="http://earth.esa.int/um/eop"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <q0:authenticate>
      <q0:username>TestUser</q0:username>
      <q0:password>TestUser42</q0:password>
    </q0:authenticate>
  </soapenv:Body>
</soapenv:Envelope>
```

**Example Encrypted Authentication Response:**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
xmlns:ns="http://earth.esa.int/um/eop">
  <ns:authenticateResponse>
    <Assertion xmlns="http://earth.esa.int/um/eop/saml">
      <xenc:EncryptedData xmlns:xenc="http://www.w3.org/2001/04/xmlenc#" Type="http://www.w3.org/2001/04/xmlenc#Content">
        <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"/>
        <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
          <xenc:EncryptedKey>
```

15
It should be noted that the “assertion” contained in the AuthenticateResponse is **not** a SAML assertion but is an element containing an encrypted SAML token. This is specified in dail-enc-schema.xsd (see Annex). The tag enables identification of the element as an encrypted SAML assertion.

### 6.5.2 Message Digest

The secure hash SHA-1 digital signature message digest algorithm is proposed and is supported by [NR15]. The SAML token is signed before it is encrypted.

Example signed token before encryption.

```xml
<assertion xmlns="urn:oasis:names:tc:SAML:1.0:assertion"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:oasis:names:tc:SAML:1.0:schema:assertion">
  <version>1.0</version>
  <subject>
    <nameIdentifier>dail</nameIdentifier>
    <subjectConfirmation method="urn:oasis:names:tc:SAML:1.0:cm:assertionMethod"/>
  </subject>
  <conditions>
    <notBefore>2009-06-25T13:34:55Z</notBefore>
    <notOnOrAfter>2009-06-25T13:39:55Z</notOnOrAfter>
  </conditions>
    authenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password"/>
  <subject>
    <nameIdentifier>dail42</nameIdentifier>
    <subjectConfirmation/>
  </subject>
</assertion>
```
<saml:ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:bearer</saml:ConfirmationMethod>

</saml:SubjectConfirmation>
</saml:Subject>

<saml:Attribute AttributeName="hmaId" AttributeNamespace="http://earth.esa.int/um/eop/saml">
<saml:AttributeValue>DAIL42</saml:AttributeValue>
</saml:Attribute>

<saml:Attribute AttributeName="c" AttributeNamespace="http://earth.esa.int/um/eop/saml">
<saml:AttributeValue>Italy</saml:AttributeValue>
</saml:Attribute>

<saml:Attribute AttributeName="o" AttributeNamespace="http://earth.esa.int/um/eop/saml">
<saml:AttributeValue>ESA</saml:AttributeValue>
</saml:Attribute>

<saml:Attribute AttributeName="hmaProjectName" AttributeNamespace="http://earth.esa.int/um/eop/saml">
<saml:AttributeValue>HMA imp</saml:AttributeValue>
</saml:Attribute>

<saml:Attribute AttributeName="hmaAccount" AttributeNamespace="http://earth.esa.int/um/eop/saml">
<saml:AttributeValue>daisp</saml:AttributeValue>
</saml:Attribute>

<saml:Attribute AttributeName="hmaServiceName" AttributeNamespace="http://earth.esa.int/um/eop/saml">
<saml:AttributeValue>catalogue</saml:AttributeValue>
</saml:Attribute>

</saml:AttributeStatement>

<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"/>
The security model proposed requires that the authentication request is further decomposed into two cases as described in the following section.
6.5.3 Authentication Use Cases

6.5.3.1 Default Case - Local Entity is IdP

In this use case the authentication request contains an identifier specifying the local entity as IdP. This is the default case when no IdP is provided in the request.

Example:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:q0="http://earth.esa.int/um/eop"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <q0:authenticate>
      <q0:username>TestUser</q0:username>
      <q0:password>TestUser42</q0:password>
    </q0:authenticate>
  </soapenv:Body>
</soapenv:Envelope>
```

Or

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:q0="http://earth.esa.int/um/eop"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <q0:authenticate>
      <q0:username>TestUser</q0:username>
      <q0:password>TestUser42</q0:password>
      <q0:servername>
      </q0:servername>
    </q0:authenticate>
  </soapenv:Body>
</soapenv:Envelope>
```
Figure 2 Federating (Local) Entity is request designated IdP (Default Case)

1. The authentication request is sent to the authentication service using SOAP over HTTPS. (May pass through the policy enforcement point (PEP) of the federating entity.)

2. (The PEP of the federating entity receives the request and forwards it to the authentication service of the federating entity.)

3. The authentication service verifies the identity in the **local** user registry over LDAP/LDAPS.

4. The authentication service creates a SAML token using the minimum profile attributes retrieved from the user registry. The SAML token is created containing assertion of the authentication and assertion regarding the value of the subset of attributes from the minimum user profile (see description in section 6.5.5).

5. The authentication service signs the SAML token using the Federating (local) Entity private key.

6. The authentication service encrypts the SAML token with the Federating (local) Entity public key.

7. The authentication response containing the encrypted and signed SAML token is returned to the client using SOAP over HTTPS.

8. The client is unable to decrypt the content.

6.5.3.2 External Entity is IdP
In this use case the authentication request contains an identifier for the external entity authentication service. The relation table between identifiers and external entities authentication service url shall be stored on the server and configured at service deployment time. It must be done in this way for security as the system must deny access to untrusted authentication server.

Example Request with IdP:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:q0="http://earth.esa.int/um/eop"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <q0:authenticate>
      <q0:username>TestUser</q0:username>
      <q0:password>TestUser42</q0:password>
      <q0:servername>spot</q0:servername>
    </q0:authenticate>
  </soapenv:Body>
</soapenv:Envelope>
```
1. The authentication request is sent to the policy enforcement point (PEP) of the federating entity using SOAP over HTTPS.

2. The PEP of the federating entity receives the request and forwards it to the authentication service of the federating entity.

3. The authentication service redirects the authentication request to the PEP of the designated External IdP. The authentication service URL is extract from the table previously described.

4. The PEP of the external entity forwards the message to the authentication service of the external entity.

5. The authentication service verifies the user in the external entity user registry.

6. The authentication service creates the SAML token using the minimum profile attributes retrieved from the user profile in the user registry.

7. The authentication service signs the SAML token using the External Entity private key

8. The authentication service encrypts the SAML token with the Federating Entity public key.

9. The authentication response containing the SAML token in the SOAP body is returned.

10. The client receives but cannot decrypt the token.

6.5.4 Authorisation Request

The authorisation request may contain an encrypted SAML token in the WS-Security element of the SOAP header. This SAML token is obtained from an authentication request as previously described and is used to control access to services.
N.B. It is not mandatory that the authorisation request is preceded by an authentication request as the SAML token is not mandatory in the service request. However, access to services is controlled by the policies applied in the PEP.

6.5.5 OASIS SAML

SAML (Security Assertion Markup Language) [NR11] is the OASIS Security Services Technical Committee XML standard for exchanging authentication and authorisation data between security domains, i.e. exchange between an identity provider (producer of assertions) and a service provider (consumer of assertions).

SAML is required to implement federated identity and identifies two roles; the identity provider (IdP) and the service provider. These communicate through SAML assertions. A SAML assertion is an XML document containing information about how the user was authenticated and can contain other user attributes. SAML bindings are defined for HTTP Post and SOAP.

SAML includes mechanisms that allow providers to communicate privacy policy/settings from one to the other. For instance, a Principal's consent to some operation being performed can be obtained at one provider and this fact communicated to another provider through the SAML assertions and protocols.

A SAML assertion is a package of information that supplies one or more statements made by a SAML authority.

- Authentication: The specified subject was authenticated by a particular means at a particular time. A typical authentication statement asserts Subject S authenticated at time t using authentication method m.

- Attribute: The specified subject is associated with the supplied attributes. A typical attribute statement asserts Subject S is associated with attributes X,Y,Z having values v1,v2,v3. Relying parties use attributes to make access control decisions

WS-Security SAML Token Profile [NR11] defines how SAML assertions are processed in SOAP messages.

SAML 1.1 is proposed to encode the user authentication token.

The following subset of attributes necessary to implement the basic EO DAIL policy steps are proposed to be included in the SAML token (see GMES Minimum User Profile [OR2]):

<table>
<thead>
<tr>
<th>Minimum Profile</th>
<th>DAIL Part Profile</th>
<th>Mandatory data (not exported)</th>
<th>Description</th>
<th>inetOrgPerson mapping</th>
<th>Extended Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>hmaId</td>
<td></td>
<td>Unambiguous HMA identity</td>
<td>uid</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c</td>
<td></td>
<td>Country of origin</td>
<td>homePostalAd dress</td>
<td></td>
<td></td>
</tr>
<tr>
<td>o</td>
<td></td>
<td>Organisation</td>
<td>o</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hmaProjectNam</td>
<td></td>
<td>Names of projects with which user is</td>
<td>hmaProjectName</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 1: Attributes in SAML Token

<table>
<thead>
<tr>
<th>e</th>
<th>affiliated.</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>hmaAccount</td>
<td>The HMA account number</td>
<td>hmaAccount</td>
</tr>
<tr>
<td>hmaServiceName</td>
<td>Associated services</td>
<td>hmaServiceName</td>
</tr>
<tr>
<td>userProfile</td>
<td>Commercial/GME S/scientific</td>
<td>userProfile</td>
</tr>
<tr>
<td>email</td>
<td>Email address</td>
<td>mail</td>
</tr>
<tr>
<td>password</td>
<td>password</td>
<td></td>
</tr>
<tr>
<td>state</td>
<td>Enabled/disabled. This information allows an administrator to disable a specific user.</td>
<td>state</td>
</tr>
<tr>
<td>homePostalAddress</td>
<td>Home address</td>
<td>homePostalAddress</td>
</tr>
<tr>
<td>IdP</td>
<td>Identity provider</td>
<td>IdP</td>
</tr>
</tbody>
</table>

It should be noted that certain information such as password and the enabled/disabled state of a user is required to be held in the minimum user profile in the registry but shall not form a part of the data exported in the SAML token.

6.5.6 OASIS Ws-Security

Web Services Security [NR9] from OASIS is a communications protocol providing for security of web services. WS-Security 1.0 was released on April 19 2004 and version 1.1 on February 17 2006.

WS-Security is proposed to encode the SAML assertions in the SOAP header. WS-Security SAML Token Profile defines how SAML assertions are processed in SOAP messages and so it is proposed for this interface.

6.5.6.1 Encryption

Encryption is required to prevent the message content being read by someone other than the intended recipient. N.b. It does not prevent the message being modified, for this a digital signature is required.

The recipient, in this case the service providers “publish” their certificates allowing “anyone” to encrypt a message to them using the published public key. Only the recipient holding the corresponding private key can decrypt such a message.
6.5.6.2 Signature

WS-Security permits digital signatures to be used to prove that the message has not been changed since sending. A recipient can be sure that it is the user who has signed the message. The XML signature <ds:Signature> element of WS-Security can be used for signature.

a. Sender: Hash and signs (encrypts the hash code)
b. Receiver: Hash and verify hash (decrypts the hash)
c. Ensures that the message was sent by a known client and that the message arrived intact.

![Digital Signature Diagram]

**Figure 4 Digital Signature**

Message encryption is not sufficient to guarantee that the message comes from a trusted client as this depends on how many people know the “encryption code”. It does not prevent someone from changing the message content.

SAML used with XML signature <ds:Signature> element of WS-Security allows signing the messages as well:

1. Sender: Hash and signs (encrypts the hash code)
2. Receiver: Hash and verify hash (decrypts the hash)

7 Interface

7.1 Authenticate

The Authenticate operation allows clients to retrieve authentication metadata from a nominated IdP server. The response to an Authenticate request should be an XML document containing authentication metadata about the authentication and requestor.

7.1.1 Request

Protocol: SOAP over HTTPS
7.1.2 XML encoding

The following XML-Schema fragment defines the XML encoding of the message body of the Authenticate operation.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
 xmlns:s0="http://earth.esa.int/um/eop"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <s0:authenticate>
      <s0:username>TestUser</s0:username>
      <s0:password>TestUser42</s0:password>
      <s0:serverName>spot</s0:serverName>
    </s0:authenticate>
  </soapenv:Body>
</soapenv:Envelope>
```

Figure 5: Example Authenticate Request

7.1.3 Response

The following XML shows an encrypted example response. The authenticate response message is always encrypted with the DAIL public key i.e. in both the use cases the client receives the same response:

- the federated response message to the federating entity authentication service and coming from an external Idp.
- The federated response message returned by the federating entity authentication service to a client.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
 xmlns:ns="http://earth.esa.int/um/eop"
 xmlns:nss="http://www.w3.org/2001/XMLSchema" Type="http://www.w3.org/2001/04/xmlenc#">
  <ns:Assertion xmlns:nss="http://www.w3.org/2000/09/xmldsig#">
    <nss:EncryptedData xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
      <nss:CipherData>
        <nss:CipherValue>cbE8vIFomyDuxR8N4EdwS9UKpSoUbMrWSVprW7iypMwFZLeHR9Rxd4iw5
      dU4K7ffyNdR9Tr99d8YIdpFLzCvYas63g5x4/XnyA1E2AU8ZBBpM2dtr3g4YMywFPrw176
      mHM+MEVRZdAm0NFhqXhcS92m3Yam14mK=</nss:CipherValue>
      </nss:CipherData>
    </nss:EncryptedData>
  </ns:Assertion>
</soapenv:Envelope>
```
Figure 6: Example Authentic Response
7.1.3.1 Example Authentication Response Before Encryption

An example is given here for completeness of the fragment before encryption:

```
<assertion xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
            xmlns:xmlsec="urn:oasis:names:tc:SAML:1.0:assertion"
            AssertionID="oracle.security.xmlsec.saml.Assertion1955a65"
            MajorVersion="1" MinorVersion="1">


  <assertionAuthenticationStatement AuthenticationInstant="2009-06-
                                                                   25T13:34:55Z"
                                      AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password">

    <subject>

      <nameIdentifier>dail</nameIdentifier>

      <subjectConfirmation>
        <confirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:bearer</confirmationMethod>
      </subjectConfirmation>

    </subject>

  </assertionAuthenticationStatement>

  <attributeStatement>

    <subject>

      <nameIdentifier>DAIL42</nameIdentifier>

      <subjectConfirmation>
        <confirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:bearer</confirmationMethod>
      </subjectConfirmation>

    </subject>

  </attributeStatement>

  <attribute NameId="hmad" Nombre="http://earth.esa.int/um/eop/saml">

    <attributeValue>DAIL42</attributeValue>

  </attribute>

  <attribute NameId="c" Nombre="http://earth.esa.int/um/eop/saml">

    <attributeValue>Italy</attributeValue>

  </attribute>

</assertion>
```
<saml:Attribute AttributeName="o"
AttributeNamespace="http://earth.esa.int/um/eop/saml">
  
  <saml:AttributeValue>ESA</saml:AttributeValue>
</saml:Attribute>

<saml:Attribute AttributeName="hmaProjectName"
AttributeNamespace="http://earth.esa.int/um/eop/saml">
  
  <saml:AttributeValue>HMA</saml:AttributeValue>
</saml:Attribute>

<saml:Attribute AttributeName="hmaAccount"
AttributeNamespace="http://earth.esa.int/um/eop/saml">
  
  <saml:AttributeValue>dailsp</saml:AttributeValue>
</saml:Attribute>

<saml:Attribute AttributeName="hmaServiceName"
AttributeNamespace="http://earth.esa.int/um/eop/saml">
  
  <saml:AttributeValue>catalogue</saml:AttributeValue>
</saml:Attribute>

</saml:AttributeStatement>

<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="" />
    
    <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
    
    <ds:Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments"/>
    
  </ds:Transforms>

  <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>

  <ds:DigestValue>nLkuqyqDgsxQnPzVDDckxaA0</ds:DigestValue>

</ds:Reference>
</ds:SignedInfo>
Date: June 30, 2009

User Management Interfaces for EO

7.1.4 Failed Authentication Request

Security considerations require that full error information is not returned to the user. An example is given below:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:soapenv:Body = "http://schemas.xmlsoap.org/soap/envelope/"
  soapenv:Fault = "http://schemas.xmlsoap.org/soap/envelope/"
  <faultcode>soapenv:Server"</faultcode>
  <faultstring>Exception occurred while trying to invoke service method Authenticate</faultstring>
  <faultstring>Fault = "http://schemas.xmlsoap.org/soap/envelope/"
  </soapenv:Fault>
</soapenv:Envelope>
```

7.1.5 WSDL

The WSDL is given below for the authentication web service used by the identity provider.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:wsdl = "http://schemas.xmlsoap.org/wsdl/
  xmlns:mime = "http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:soap12 = "http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:http = "http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:nsl = "http://org.apache.axis2/xsd"
  xmlns:wsaw = "http://www.w3.org/2006/05/addressing/wsdl"
  xmlns:ns = "http://earth.esa.int/um/eop"
  xmlns:xs = "http://www.w3.org/2001/XMLSchema"
```
Date: June 30, 2009

User Management Interfaces for EO

```
<wsdl:documentation>
<wsdl:operation name="authenticate">
<wsdl:input message="ns:authenticateRequest"/>
<wsdl:output message="ns:authenticateResponse"/>
</wsdl:operation>
</wsdl:documentation>
```

```xml
<wsdl:documentation>
<wsdl:operation name="authenticate">
<wsdl:input message="ns:authenticateRequest"/>
<wsdl:output message="ns:authenticateResponse"/>
</wsdl:operation>
</wsdl:documentation>
```
7.2 ServiceRequest

Through the implementation of this interface to the ServiceRequest (i.e. the service operations such as the catalogue GetRecords, the programming GetFeasibility etc.) authenticated clients will send requests to a server controlling access to the final service. The request is made using WS-Security containing the SAML token previously returned in the AuthenticationResponse.

N.b. The service requests from a client to the DAIL or from the DAIL to a service provider are the same.

7.2.1 Request

Protocol: SOAP plus WS-Security over HTTP/HTTPS.

7.2.2 XML encoding

The following XML-Schema fragment defines the XML encoding of an example ServiceRequest operation.


<ogc:Literal>2009-06-26T00:00:00.000</ogc:Literal>


<ogc:Literal>2009-06-26T23:59:59.000</ogc:Literal>

<ogc:PropertyName>$acquisitionPlatform/@objectType</ogc:PropertyName>


<ogc:PropertyName>$acquisitionPlatform/@id</ogc:PropertyName>

<ogc:PropertyName>$acquisitionPlatform/@sourceObject</ogc:PropertyName>

<ogc:PropertyName>$acquisitionPlatAsso/@associationType</ogc:PropertyName>


<ogc:PropertyName>$acquisitionPlatAsso/@targetObject</ogc:PropertyName>

<ogc:PropertyName>$acquisitionPlatform/@id</ogc:PropertyName>

<ogc:And>

<ogc:Filter>
  <csw:Constraint>
    <csw:Query>
      <csw:GetRecords>
        <env:Body>
        </env:Enveloped>
      </csw:GetRecords>
    </csw:Query>
  </csw:Constraint>
</ogc:Filter>

Figure 8: Service Request Example
7.2.3 Failed Request

An example is given below:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Body>
        <soapenv:Fault>
            <faultcode>AuthorisationFailed</faultcode>
            <faultstring>Country of origin not authorised</faultstring>
        </soapenv:Fault>
    </soapenv:Body>
</soapenv:Envelope>
```

7.3 ServiceResponse

7.3.1 Synchronous

The service response is as defined in the corresponding catalogue, ordering and programming ICDs.

7.3.2 Use Case: User logs in at client and makes Synchronous Service Request to HM Service

In this case the sequence is as follows:

1. A user already registered in the local (DAIL) user registry logs in at the client (DAIL Client or an external client).
2. The client authenticates the user through the DAIL authentication service.
3. The DAIL authentication service:
   - validates the user,
   - creates the SAML assertions from the information held in the user registry
   - signs the SAML assertion with the DAIL private key.
   - encrypts the signed SAML assertion with the DAIL public key.
4. The DAIL authentication service returns the authentication response containing the encrypted SAML token.
5. The user is given confirmation of login.
6. The user selects a service.
7. The client constructs the service request and inserts the encrypted SAML assertion in the request. (N.b. The client should manage the token validity as it is possible that the token has expired and will therefore not pass the PEP checks. To ensure the token has not expired an authentication request could be be requested at each service request).
8. The service request is sent.
9. The DAIL PEP decrypts the token using the DAIL private key and applies policy checks.
10. The request is forwarded to the DAIL HM service.

11. GS service requests are constructed and the SAML token in the header is encrypted with the GS public key.

12. The service request is sent to the GS PEP.

Figure 9 DAIL Sequence Diagram Showing Client Request with synchronous Response
7.3.3 Asynchronous

The asynchronous service response is as defined in the corresponding catalogue, ordering and programming ICDs. This response may be protected by the same encryption and signature as defined for the service request and authentication. The sequence is as follows:

1. The SP prepares the response to the endpoint mentioned in the WS_Addressing.

2. The service provider creates a token authenticating himself i.e. GS and signs it with his private key. This is then encrypted with the public key of the DAIL and inserted into the asynchronous response in the same way as previously described for a service request.

3. The asynchronous response is returned to the address provided in the ws_addressing of the request. This will normally be the address of a PEP.

Figure 10 Sequence Diagram showing asynchronous request

Annex - Schemas

authentication.xsd

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<x:schema xmlns:x="http://www.w3.org/2001/XMLSchema"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns="http://earth.esa.int/um/eop" xmlns:saml="http://earth.esa.int/um/eop/saml"
targetNamespace="http://earth.esa.int/um/eop" elementFormDefault="qualified">
  <x:import namespace="http://earth.esa.int/um/eop/saml"
    schemaLocation="/dail-enc-schema.xsd"/>
  <x:element name="authenticateResponse"/>
```
1. The XML schema file `dail-enc-schema.xsd` defines a complex type that includes elements for authentication and user management.

2. The `oasis-200401-wss-wssecurity-secext-1.0.xsd` file contains additional schema definitions for securing web services.
http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd

**oasis-sstc-saml-schema-assertion-1.1.xsd**