Open Geospatial Consortium Inc.

Date: 2009-02-07

Reference number of this OGC® project document**: 07-118r1**

Version: 0.0.3

Category: OGC™ Interoperability Program Report

Editors: R.Smillie, A.Cucumel SPACEBEL s.a.

User Management Interfaces for Earth Observation Services

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

Contents

Figures

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:

Contact	Organisation	Email
Rowena Smillie	Spacebel	Rowena.Smillie@spacebel.be
Alexandre Cucumel	Spacebel	Alexandre.Cucumel@spacebel.be
Wouter Van de Weghe	Oracle	wouter.van.de.weghe@oracle.com

iv. Revision history

Date	Version	Editor	Sections modified	Description
15 September 2007	0.0.1 Draft	R.Smillie	All	Initialised Draft Document.
23 April 2008	0.0.2	R.Smillie		Updated in line with EO DAIL implementation project
07 February	0.0.3	R.Smillie		Updated in line with EO DAIL implementation project SOAP version changed to 1.1 Authentication request does not use WS-Security Message examples added Encryption and signature descriptions improved

v. Foreword

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

1. The Extensible Markup Language (XML), World Wide Web Consortium, http://www.w3.org/TR/1998/REC-xml-19980210
2. Simple Object Access Protocol (SOAP) Version 1.1 W3C Note 08 May 2000 , http://www.w3.org/TR/2000/NOTE-SOAP-20000508/
3. WSDL, Web Services Description Language (WSDL) 1.1, http://www.w3.org/TR/wsdl

4. SAML, Security Assertion Markup Language 1.1, OASIS http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security

5. WS-Security Web Services Security 1.1, OASIS http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss

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

[NR1] W3C Recommendation January 1999, Namespaces In XML, http://www.w3.org/TR/2000/REC-xml-names

[NR2] W3C Recommendation 6 October 2000, Extensible Markup Language (XML) 1.0 (Second Edition), http://www.w3.org/TR/REC-xml

[NR3] W3C Recommendation 2 May 2001: XML Schema Part 0: Primer, http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/

[NR4] W3C Recommendation 2 May 2001: XML Schema Part 1: Structures, http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

[NR5] W3C Recommendation 2 May 2001: XML Schema Part 2: Datatypes, http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/

[NR6] W3C Simple Object Access Protocol (SOAP) Version 1.1 W3C Note 08 May 2000 , http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

[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

[NR9] WS-Security, SOAP Message Security V1.1 http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf

[NR10] SAML, Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V1.1 http://www.oasis-open.org/committees/download.php/3406/oasis-sstc-saml-core-1.1.pdf

[NR11] Web Services Security SAML Token Profile 1.1 http://www.oasis-open.org/committees/download.php/16768/wss-v1.1-spec-os-SAMLTokenProfile.pdf

[NR12] Secure Hash Standards (SHA-1) National Institute of Standards and Technology http://csrc.nist.gov/cryptval/shs.htm

[NR13] HMA Architectural Design Technical Note version 1.6, 13/06/2007 http://services.eoportal.org/portal/system/HelpUI.jsp

[NR14] Glossary for the OASIS Security Assertion Markup Language (SAML) V2.0 http://docs.oasis-open.org/security/saml/v2.0/saml-glossary-2.0-os.pdf

[NR15] Java Cryptography Architecture API Specification & Reference

http://java.sun.com/j2se/1.5.0/docs/guide/security/CryptoSpec.html

[NR16] OGC 04-016r5, OWS Common Implementation Specification 2004/12/17

[NR17] XML encryption http://www.w3.org/TR/xmlenc-core/

[NR18] XML signature http://www.w3.org/TR/xmldsig-core/

[NR19] Apache XML Security http://santuario.apache.org/Java/index.html

3.2 Other references

[OR1] HMA Operational Scenarios Technical Note HMA-TN-ASU-SY-0001 http://services.eoportal.org/portal/system/HelpUI.jsp

[OR2] HMA-TN-SIE-EN-003 GMES Minimum User Profile 1.0c

[OR3] EO-DAIL User Management Specifications

DAIL-RS-ASU-EN-0002 1.1

[OR4] SOAP Version 1.2 http://www.w3.org/TR/soap12-part1/

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

Service Provider (Ground segment) components receiving Web service requests 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 within the circle of trust to the identity provider for authentication but for the interface context this is transparent as the federated identity request would be identical to the initial request.)

2) Each subsequent service request by the client (Web service consumer) is to include the SAML token in the SOAP header in the way described 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 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.

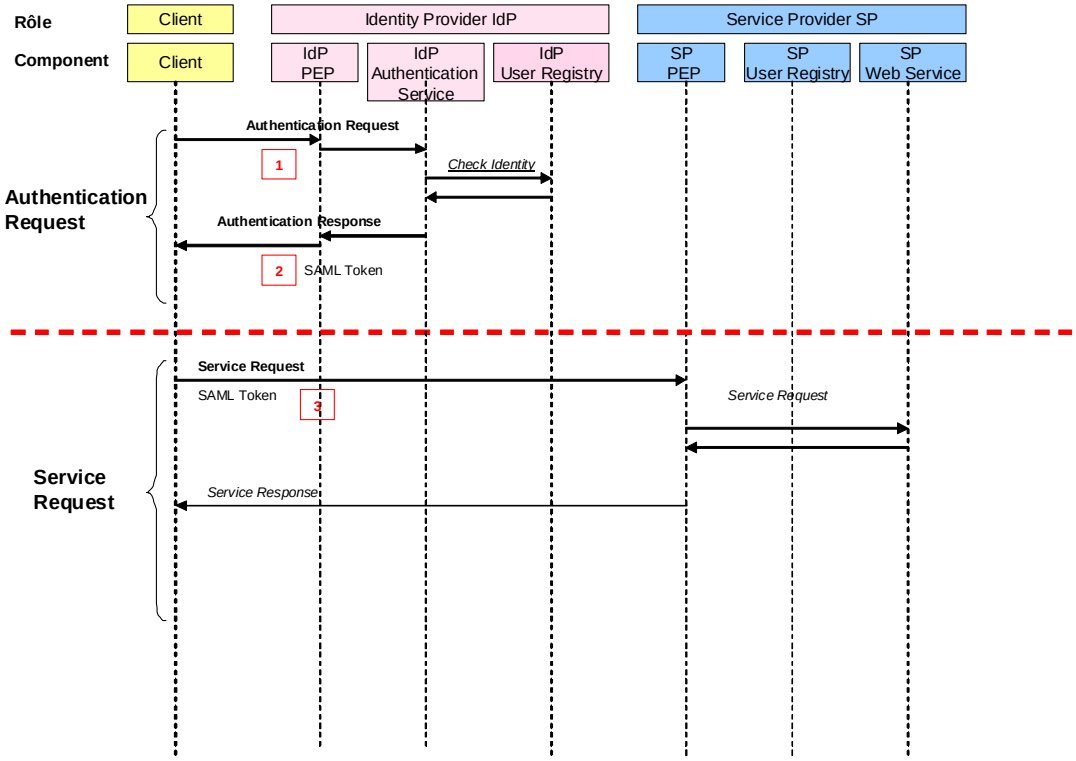

Figure 1 User Management Use Cases

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 may be intercepted by the IdP PEP.

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 (and therefore modify) the content.

6.5.1 Encryption

Encryption and decryption of the SAML token is performed by the authentication service during an authentication request and response. It is performed by the PEP during the authorization request and response. 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.

An example request is given below:

Example Authentication Request:

```
<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>Alexandre CUCUMEL</q0:username>
<q0:password>abc1234</q0:password>
</q0:Authenticate>
</soapenv:Body>
</soapenv:Envelope>
```

Example Encrypted Authentication Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
   <soapenv:Body>
      <ns:AuthenticateResponse xmlns:ns="http://earth.esa.int/um/eop">
         <ns:return>
            <saml:Assertion
xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
xmlns="urn:oasis:names:tc:SAML:1.0:assertion" IssueInstant="2009-01-
07T09:23:32Z" Issuer="http://www.spacebel.be" MajorVersion="1"
MinorVersion="1">
               <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>
                        <xenc:EncryptionMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
                        <xenc:CipherData>
```

```
   <xenc:CipherValue>Sefd50aLQcVONxPObMleJsGt+51CpN7QkdO9rtLolw6F87ssgWwlvt/
zybu/rIO9nM16+xqCZWH2IVoe3+WimsYzih8XECa8n/2S75gXsVAZt6sOBUC2WYSbusehsF9d7c7
sEqaZVTlpkKNqalBhJ2d1eZPTzQHPDPWIGnjSYW0=</xenc:CipherValue>
                                    </xenc:CipherData>
                               </xenc:EncryptedKey>
                        </ds:KeyInfo>
                        <xenc:CipherData>

   <xenc:CipherValue>wMDLujhjBwTYSgumuuBWVNDhx+ar5bb5NVSFd5s9HOvYT5CJ+XwMPPy
TwCaohj6po4y+3rvH6+XLQU3XpueQR3jWd4xBZnKxQSfNTD6edVZp4T1dJfW3CW/EmfHcE494iGb
g33HVi7AU3nq3ZpztdYkIYewVQrOKpsiZCqeXorFQEHA7/zNA8lGHKSFKPfx5dhvxk7M1B73W+30
kkzIhtfHEK/pirdHfUzAstCF5ECAjbJjgju1pLtkl5fHHnlWgVZvfFwKUYj0DfwzgvRtGae/Kcve
swSUan3p/3oVQJ2nLY3AEcXzGZIypSLBP9yhjyocvr9SIVzy6UIKAbOXepL0Sgk5orv5xm6n2QpA
9AGdJIWVR/nByyMo3bNAlAs40xj9h18QEiyjvV3IwZnCeeZOHfmXWi7sHplCajsm+N9/Qx5e/cJl
BsLzJauYqCZAAgF9OSTafT4gMlV+lynyiTQRLJpDGbf46dCGcu7UGt2HGGTlhqf1je3AKR9yvx3C
WDcIYYzZU2ygt5sSwILOloXdBWpUEtYU0r6wPZBBcIiWSR0ItcksZK6QNsZFBjForj7ppaAERDUl
Mco3BTMhQueduGlGuD4DXT2MHPrZRmLMqWyaD7oqTd9i4t076s190FVxfjl27lqwvMS7hv6vrXPY
fnSS2bKEXnzSbeSN0HxyKpaXVXvIBUk1xfLf2+QmUzQG53uTrTMwBOtVMEY/LwftpUff6NlJm+Hf
avL2JIcjA1RlFpUmRbkhNzrsPT5+ZFlwfsIKDz/BpNwCiF2vdgiC3YxgHqZRPExfcgt/RTGEyjkk
bdCa0dpY0Sk4S2VNvG79izygBw9U2lMS+DNI4dVK/ySNbnJUlmwnxxQcYMzcqKfkPsLjddUKNLP2
WUmrF80tadmGmBxAX/az1dH/9hEmg4mcEQs5c3VSACbUXS5vTT7R9LjRUcdovAiOXr3nHWUODn7e
KT2rvavB9t8xpoDMi6taguWvtUJvdzGGNMb9MrXhQxUA1s6MCccopDj2LmEhFXDuGuHNz03qOmYq
ehvdLT7qVeU109I3IsZAHkyXad0vrheZmO7D1Z0BMorpR8iYdGA1stB0ovywVFZK6VEV0YsliQpF
MHXaYMHNZHSX2/sn6Uso8ff4DvbMM5uk9ddFiWU1hD/VbGG6zgxiVWAUr6oJbhCz9/r13bwm98lc
8yOYAakwHpEYQBFN9BWGZAcNvFbkUtP/EuEMu7rxcOzR7Bf/ZBTKNB93x9OjgW+BYSZ3nmaDCJro
MFqYAXX9HLR9vfUu8bSZPLhI1kcTTlI55Jzx0zM1LiXL6+Xnqd9Z44UyRMx7sduHRt90ojgMTQ2I
jtwB4bGyq9oGnxg2Li50fQseyToiJhG4EE8Y7479OQhyecOHBfmnzdKwB30P1r/oDFDvXFBrxWTw
gXxYUTDAcHB9NtNPA09VjLmrMblqRc9FsPbWLCxV2TPzInJhO2SPwsU5P+Ng5RqwH2tpice8Ob0S
G0Iug/dR1P+uYGo4jQ0zv5cDPGQ0KTdQJyGkZc4fMYrEtrOrXffeN15HT1mlIN5/YTj/MMBIj5DX
V4ETnjBYXe+of7MAg4xiS4j0vLLYvfUk+NEF7NZuGCuazYvzavQF5lAhTXhjM1ajI1BRKmFBFm3l
DEsVvHBtliAxi46+e4wtJ9UNmq99rLK2OeBAuZ+wIPH1rXbM6kIH8iY6+SjP+2R2mHoVKmkGcesF
dkrwG4yrz2nBadggRsnakglABRGfFZrOUmH0wTBfiVapCIPMLlBzXopo54YAUgyTEV1uS34d7ZMP
wavaHiVyfldJD9gl4H9o/HFJ/HkF1bHFrvPFvoO7pmLyTsbRLmgMocxPb3qRaQX+LX7i0WieT9o3
wuR0ylPyl1/1VsYUAlpt4IwuZwlfUVcm4NVKTOyYPDBJk716/OSmRwltAF0+mt1LhUvV/0EC6Tb8
6TYwAc+UUD3paXBYjdydTAfD/25ymfhgOA/UpO0h39QnR8kyRyH0agt8XPt6ku94o/+gI0arbOa+
c65YTHTeOzhQn7N1VieLAmg4tT8SNoZs+7S78JVTjnK4qDoweJVgVVW3rKLu/w5VKSfqs/iFHF0M
LQcD3IEgY7bVEBgrXbIYkScHbw7B0vveQZBq5mcuHyY8cyKHAOfko92mAo448qfx3rM7I0NzsMSB
mwahRQubDVNVxOggmUp1nhuZuWCPamNZ78MdpOQtvDiQiVkAJUtSZonhiY1OIUDrBgxR5SNMfkT/
xddL+hEX3gb8rjTYu5guW0aQKhnZvJAWKNtRtdcDtePVB510AedL98Yaly2cYesLK59SzhFkkCok
weRQAWfP4pl542kq0qmIi+0LIds5kjYvXH/KK6w/ZFY83bESn7IOyZljHGPs5oL5qWSvOtroV4VA
EHqa6qbWirruXr85dDKiV8NoMpq2Y6wSvc5HtEmMUo8vemrDZ/F9KQDewztvrW7K6aNXOzfo7o+H
VulTxVHFDXaXNU1C8uMFqrUPOBBWzug+mEMKalgs2fC/1EyqZAOUBnf4F4XlMGtW2Eab5L8MfuG2
pdv4OEfgwaGeqVa7I/N/QVa8MuRTnPfny0GtwFL+rhfyrClMT+w1Eu/oxN4HxDFjpr+sG/A4OvBL
jXYLjjTqUCHGu0u15QnPJUwHdEk0kABriqWWAPH39OawlVTgbZWGQ+fSElG3/WhFWWu7mXTGPoCT
xW0PVDjKuvgUgLNy94WVAhAhMbE4rg7ep6ckSobU1SKO/wKj7uOEmd8L+Z9sercu2eDZqYNcL03Q
7fShCfOEn9k1rqLpryoLwkLiWs3cZrjK7TXuTn8El4fRKA4CwtOsPya8lOYSPhl3FMWsfhh9ssyX
+By4nrkAwKrsoRw9cLNk+ZVx4a1NyQ2iZxOBejrfeIhhedwt5Q+dWE+cVMS+xMxynAZjbJJEIDbg
gelUzDiwjEXHQ+UKbSWB+bmz7hZ1IaoFqAchEa7jHxOGojVK+vVBq8te+EdCALdMlkDpAp1gPrkc
atN5PQq/WwQaR8NC0+eK0iyuwzKAzSu4iB7zdA51KRvYfsIeJADO9IIdMrak0Yll9qLlizwXEaU/
nZyi6xf1I9jXcKfQinb9bO2lAoquft1MLkcVXO/uffprgJ0uicvYnS/2H4IWKoEWVfcX4z/dsiGv
wVSP2PAye4pAJLusjHsFQTQCQRX0bSkgXo1QZpT1YCUO9SAQPiG2u3bx/GVDpxIILpbtDwqtGH/f
L7EJWxLBFeW79GvkqgybE9bD2FR64e2c7SN4GYm4SrBKn045s6tNi923LNUkciE8rioDaptwOCeW
IH4PoVEa+DA7zNO33qtwExuH/3rAkiKOQsvadOz3XYXAVyj/zjJl4L4AiXGF2zAec1rxZLRJUeCg
yp2tJ52cGl9hpITbcLEaaQpnAjs7oD59avM/o4eJGvlfQhHMzLkh0l4B7CUzOwUOam5PspyF+N3k
bllM66/6sujkOKN8gktuR+0qDSCdEB8N4UTe4/bXLb4PDOTHnaeiBrfDTja8Gl/GzOgy8GVENTHy
QJFMYWpFN4b3lk5h8R0avtWrc5tGxwp8ZridoawA7j5alHrv+4r0e+q0ng+dLigTs/j89k9D60Ry
WvwX/y9r6snV4vfOVDNy4zhPpQJ8XPEXaNEOpUrmVYJM68n3aK3TGPLOsjmOos7ApB5l2V7wTHcV
FbuR+h0tkI3Ax9112/pvmzGJkLPzzY52mKD6zTyf7xVeeuK49LaoWguA==</xenc:CipherValue
>
                        </xenc:CipherData>
                    </xenc:EncryptedData>
                </saml:Assertion>
            </ns:return>
        </ns:AuthenticateResponse>
    </soapenv:Body>
```

```
</soapenv:Envelope>
```

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.

```
<Assertion xmlns="http://earth.esa.int/um/eop/saml"
xmlns:saml=" urn: oasis:names:tc:SAML:1.0:assertion
">          <saml:Assertion
xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
xmlns="urn:oasis:names:tc:SAML:1.0:assertion"
AssertionID="oracle.security.xmlsec.saml.Assertion@baa6ba"
IssueInstant="2009-02-02T16:02:45Z" Issuer="http://www.spacebel.be"
MajorVersion="1" MinorVersion="1">
             <saml:Conditions NotBefore="2009-02-02T15:02:45Z"
NotOnOrAfter="2009-02-03T16:02:45Z"/>
             <saml:AuthenticationStatement AuthenticationInstant="2009-02-
02T16:02:45Z"
AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password">
               <saml:Subject>
                 <saml:NameIdentifier>Alexandre
CUCUMEL</saml:NameIdentifier>
                 <saml:SubjectConfirmation>

<saml:ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:bearer</saml:Confirm
ationMethod>
                 </saml:SubjectConfirmation>
               </saml:Subject>
             </saml:AuthenticationStatement>
             <saml:AttributeStatement>
               <saml:Subject>
                 <saml:NameIdentifier>Alexandre
CUCUMEL</saml:NameIdentifier>
                 <saml:SubjectConfirmation>

<saml:ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:bearer</saml:Confirm
ationMethod>
                 </saml:SubjectConfirmation>
               </saml:Subject>
               <saml:Attribute AttributeName="mail">

<saml:AttributeValue>alexandre.cucumel@spacebel.be</saml:AttributeValue>
               </saml:Attribute>
               <saml:Attribute AttributeName="postaladdress">
                 <saml:AttributeValue>Belgium</saml:AttributeValue>
               </saml:Attribute>
               <saml:Attribute AttributeName="o">
                 <saml:AttributeValue>Spacebel</saml:AttributeValue>
               </saml:Attribute>
               <saml:Attribute AttributeName="sn">
                 <saml:AttributeValue>acl</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:Transforms>
                     <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>dYhSU2Whd8Dt65hLrj/HYBQPk9I=</ds:DigestValue>
                </ds:Reference>
              </ds:SignedInfo>

<ds:SignatureValue>DqhoRN9hd3tyX33dqm83wdMS5UURSEZBH5gFHtcBa/cU5RSyupongXcPm
rRrn4pwoWxpz+lmZESgSo3x76UZZlsIS8Fy34ed1EDcdFqXUbCoTdGoB1jm+4hnTndRHimIiQArr
QiOBJ9BdAYm7028kdrbR/JGzEqhm7Ej4AmRgTg=</ds:SignatureValue>
              <ds:KeyInfo>
                <ds:X509Data>

<ds:X509Certificate>MIICXjCCAccCBEhiBKYwDQYJKoZIhvcNAQEEBQAwdjELMAkGA1UEBhMC
QkUxETAPBgNVBAgTCEJlbGdpcXVlMRIwEAYDVQQHEwlCcnV4ZWxsZXMxETAPBgNVBAoTCFNwYWNl
YmVsMREwDwYDVQQLEwhCVSBTUEFDRTEaMBgGA1UEAxMRQWxleGFuZHJlIENVQ1VNRUwwHhcNMDgw
NjI1MDg0MTEwWhcNMDgwOTIzMDg0MTEwWjB2MQswCQYDVQQGEwJCRTERMA8GA1UECBMIQmVsZ2lx
dWUxEjAQBgNVBAcTCUJydXhlbGxlczERMA8GA1UEChMIU3BhY2ViZWwxETAPBgNVBAsTCEJVIFNQ
QUNFMRowGAYDVQQDExFBbGV4YW5kcmUgQ1VDVU1FTDCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkC
gYEAq4Vsyd7w1y+u53a70YVPELIjP0S8Lyi+cuwRjySyxf3YYPdF+9BX2zfr/wGai0kYmkO1NzCW
DTNn0gsd/EjLq0eReHrqkIKm4hdbcrbA09A9kA1MKo9SYQKDVM7oXU5rHqdbPdntoM4H2QzrgL4f
yIV/Zimt31UrdgZ3ySEVb/0CAwEAATANBgkqhkiG9w0BAQQFAAOBgQAxwyN+gEsWh+tXEES9xUM/
BHZ3aUsdh427IlJabJe28rR7bqlC+glYZD8JkLzHOP3lsxtuxWyg9kXKk0SUwOAPC2IjOEwhhL7W
BhNpKYacrxmY6kgVe6g/DBlIPD+XRe4pCQMARfwX22CyufDVSm3AM1nJTwEcw5OkM4pWFEdcjg==
</ds:X509Certificate>
                </ds:X509Data>
              </ds:KeyInfo>
            </ds:Signature>
          </saml:Assertion>
</Assertion>
```

The security model proposed requires that the authentication request is further decomposed into four cases as described in the following paragraphs.

6.5.3 Authentication

6.5.3.1 Federating Entity is request designated IdP

In this use case the authentication request contains an identifier for the federating entity authentication service.

Example Federated Request with IdP:

```
<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:AuthenticateFederated>
<q0:username>user NAME</q0:username> <q0:password>******</q0:password>
<q0:IdpName>esa</q0:IdpName>
</q0:AuthenticateFederated>
</soapenv:Body>
</soapenv:Envelope>
```

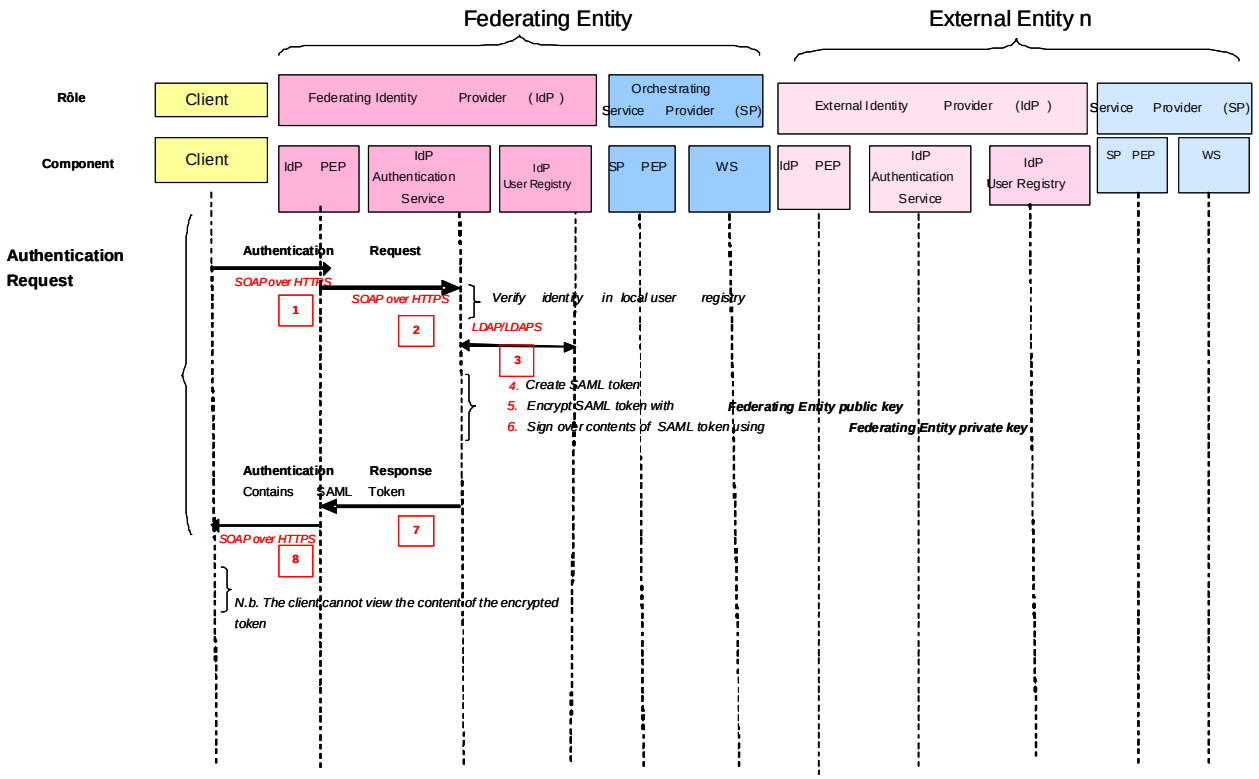

Figure 2 Federating Entity is request designated IdP

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 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 encrypts the SAML token with the Federating Entity public key.
6. (The authentication service signs over the contents of SAML token using the Federating Entity private key)
7. The authentication response containing the encrypted (and signed) SAML token is returned via the PEP to the client using SOAP over HTTPS.
8. The client is unable to decrypt the content.

6.5.3.2 External entity is request designated 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 Federated Request with IdP:

```
<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:AuthenticateFederated>
<q0:username>user NAME</q0:username> <q0:password>******</q0:password>
<q0:IdpName>spot</q0:IdpName>
</q0:AuthenticateFederated>
</soapenv:Body>
</soapenv:Envelope>
```

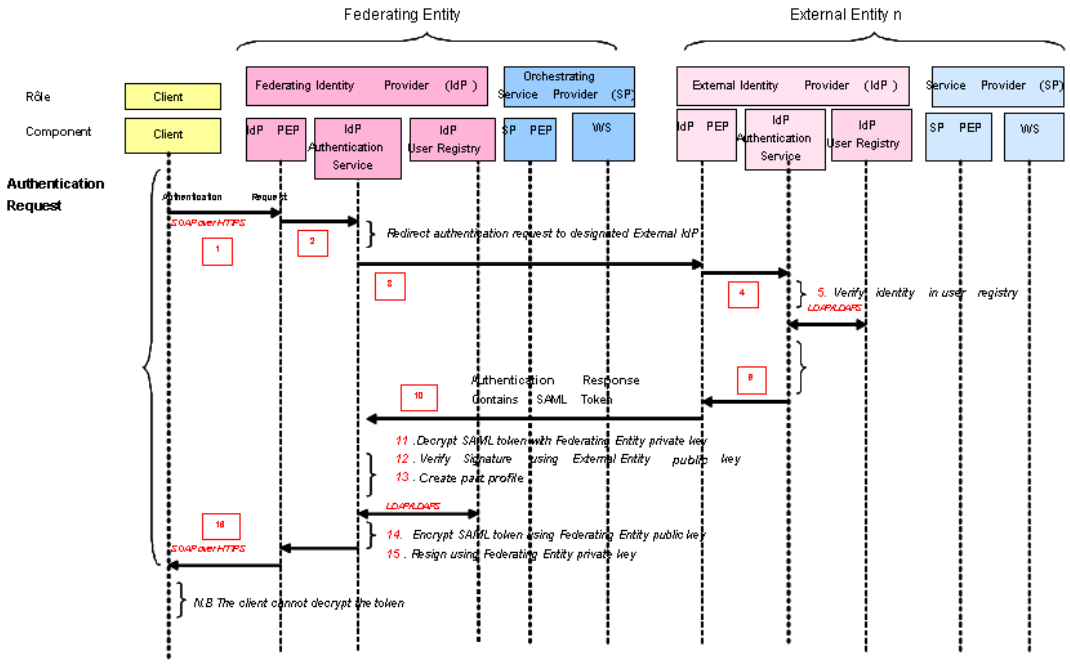

Figure 3 External Entity is request designated IdP

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 over the contents of SAML token using External Entity private key
8. The authentication service encrypts the SAML token with Federating Entity public key.
9. The authentication response containing the SAML token in the SOAP body is returned to the external entity PEP.
10. The authentication response containing the SAML token in the SOAP body is returned to the federating entity authentication service.
11. The federating entity authentication service decrypts the SAML token using the federating entity private key.
12. The federating entity authentication service verifies the signature using the external entity public key.
13. The federating entity authentication service creates the part profile for the user in the federating entity user registry over LDAP/LDAPS.
14. The federating entity authentication service encrypts the SAML token using the federating entity public key.
15. (The federating entity authentication service resigns the token using the federating entity private key).
16. The client receives but cannot decrypt the token.

6.5.3.3 No request designated IdP - Federating entity resolved as IdP

In this use case there is no IdP given in the authentication request. The authentication service assignes the local federating entity as the IdP and authenticates using the local user registry.

Example Federated Request with no IdP:

```
<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:AuthenticateFederated>
<q0:username>user NAME</q0:username>
<q0:password>******</q0:password>
</q0:AuthenticateFederated>
</soapenv:Body>
</soapenv:Envelope>
```

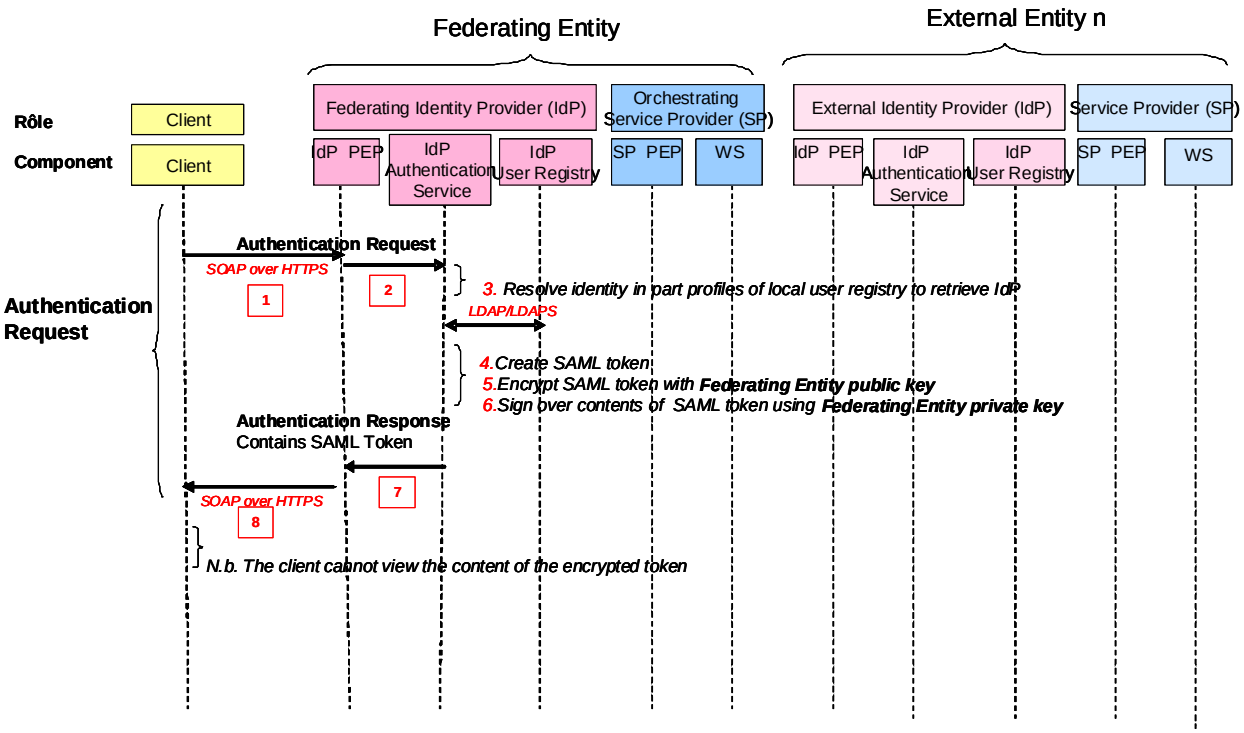

Figure 4: No request designated IdP – Federating Entity IdP

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 finds the identity in the local user registry over LDAP/LDAPS and from the part profile resolves the IdP as the local federating entity.
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 encrypts the SAML token with the Federating Entity public key.
6. (The authentication service signs over the contents of SAML token using the Federating Entity private key)
7. The authentication response containing the encrypted (and signed) SAML token is returned via the PEP to the client using SOAP over HTTPS.
8. The client is unable to decrypt the content.

6.5.3.4 No request designated Idp - External entity resolved as IdP

N.B. This scenario is feasible but is not supported in the DAIL authentication service due to security concerns over user passwords being forwarded to an incorrect IdP.

In this use case there is no IdP given in the authentication request. The authentication service resolves the Idp from the part profile stored in the local user registry which in this case contains an identifier for an external entity authentication service.

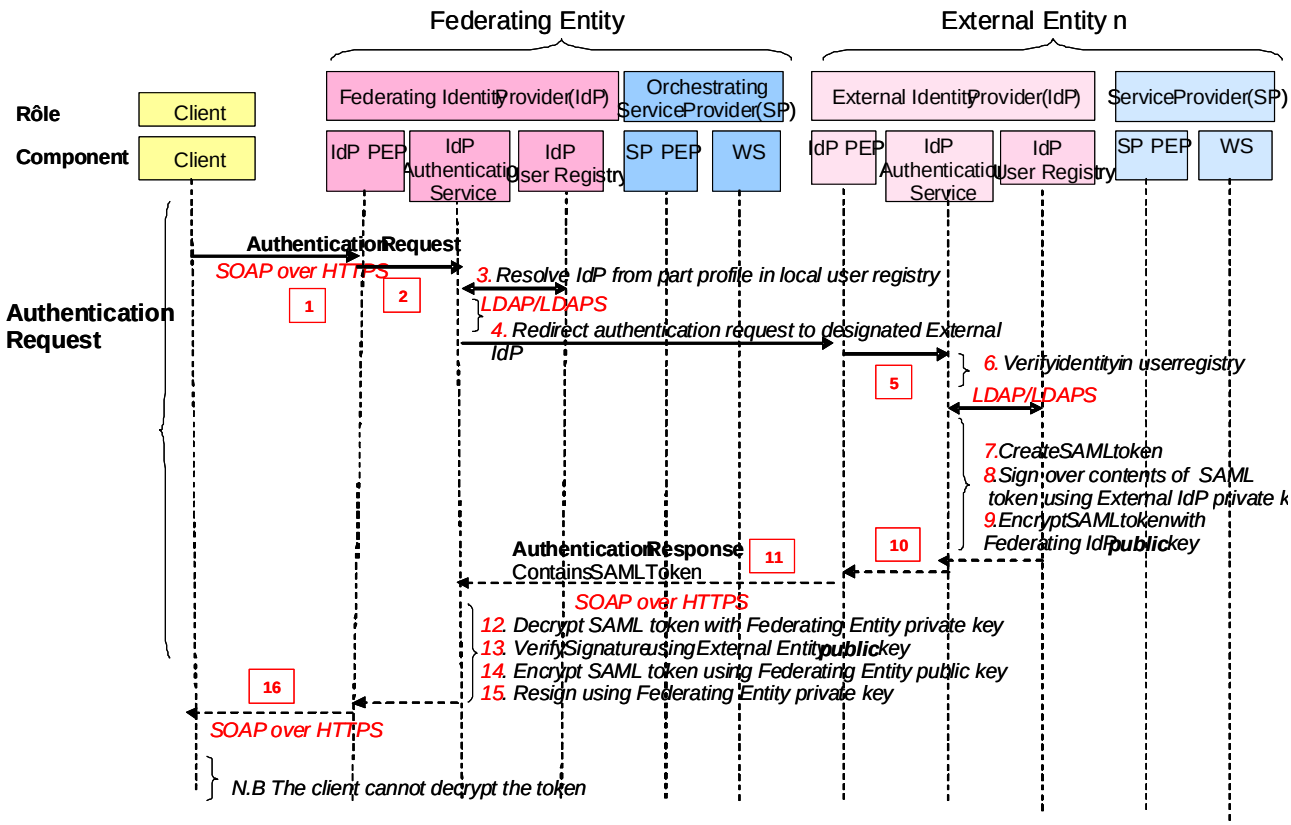

Figure 5 No request designated IdP – External IdP

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 of the federating entity resolves the IdP as external entity n from the part profile in the local user registry.
4. The authentication service forwards the authentication request to the PEP of the designated External IdP.
5. The PEP of the external entity forwards the message to the authentication service of the external entity.
6. The authentication service verifies the user in the external entity user registry.
7. The authentication service creates the SAML token using the minimum profile attributes retrieved from the user profile in the user registry.
8. The authentication service signs over the contents of SAML token using External Entity private key

9. The authentication service encrypts the SAML token with Federating Entity public key.
10. The authentication response containing the SAML token in the SOAP body is returned to the external entity PEP.
11. The authentication response containing the SAML token in the SOAP body is returned to the federating entity authentication service.
12. The federating entity authentication service decrypts the SAML token using the federating entity private key.
13. The federating entity authentication service verifies the signature using the external entity public key.
14. The federating entity authentication service encrypts the SAML token using the federating entity public key.
15. (The federating entity authentication service resigns the token using the federating entity private key.)
16. The client cannot decrypt the token.

6.5.4 Authorisation Request

The authorisation request may contain a SAML token in the WS-Security element of the SOAP header. This SAML token is obtained from a previous authentication request and is used to control access to services.

The scenario elaborated below shows the complex case where control of the request passes through an orchestration ws which subsequently propagates requests to various external protected web services.

N.B. It is not mandatory that the authorisation request is preceeded by an authentication request. The authorisation request is not required to contain a SAML token. However, access to services is controlled by the policies applied in the PEP and so only unprotected services will be accessible in this case.

6.5.4.1 Synchronous Response

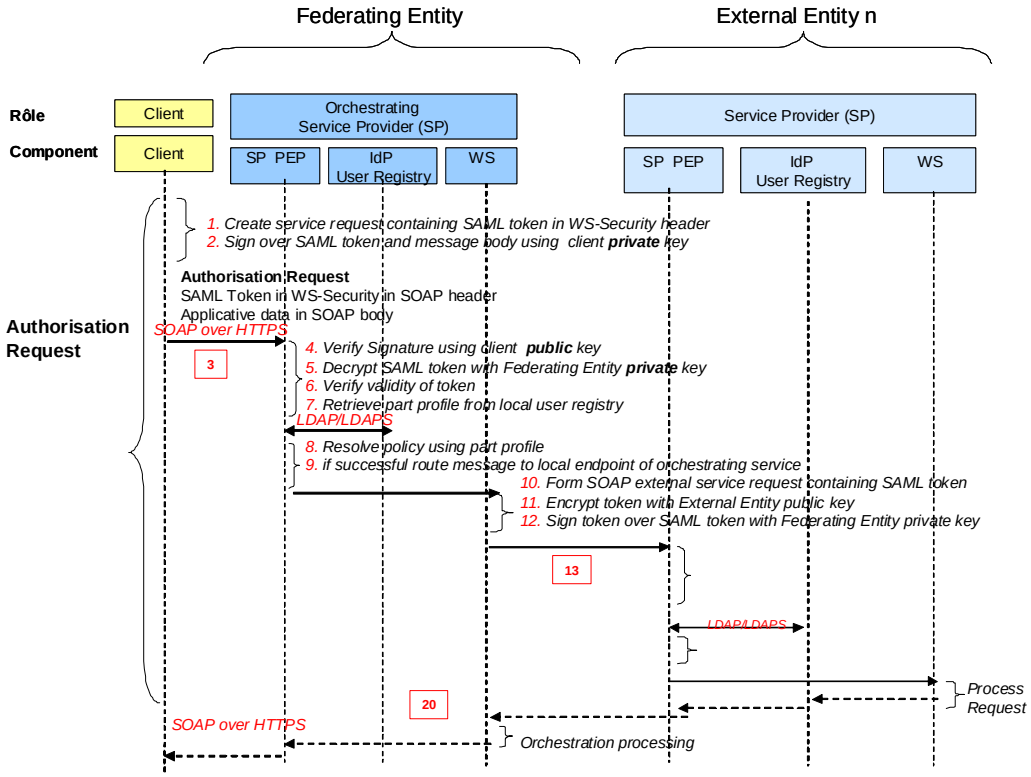

Figure 6 Authorisation Request

1. Create the service request containing the SAML token in WS-Security element of the SOAP header.
2. Sign over the SAML token and message body using the client private key.
3. The request is sent to the federating entity PEP using SOAP over HTTPS.
4. The federating entity PEP verifies the signature using the client public key.
5. The federating entity PEP decrypts the SAML token using the federating entity private key.
6. The federating entity PEP verifies the validity of the token.
7. The federating entity retrieves the part profile from the local user registry over LDAP/LDAPS.
8. The federating entity PEP enforces the policy protecting the requested service.
9. If access is permitted the request is routed to the local endpoint of the orchestrating service.
10. The federating entity orchestrating WS forms the SOAP request(s) containing the SAML token in the WS-Security element of the SOAP header.
11. The federating entity orchestrating WS encrypts the token with the external entity public key.

12. The federating entity orchestrating WS signs with the federating entity private key.
13. An signed authorisation service request containing the encrypted token is sent to the external entity PEP.
14. The external entity PEP verifies the signature using the federating entity public key.
15. The external entity PEP decrypts the SAML token using the external entity private key.
16. The external entity PEP verifies the validity of the token.
17. The external entity PEP retrieves the part profile from the external entity user registry over LDAP/LDAPS.
18. The external entity PEP enforces the policy applied to the requested service.
19. If permitted the request is sent to the external entity service endpoint requested where the request is processed.
20. The synchronous response is returned as SOAP over HTTPS.

6.5.4.2 Asynchronous Response

The asynchronous case is treated in a similar way to the synchronous case.

1. The external entity WS forms the SOAP response containing the WS-Security element in the SOAP header. This message is signed using the private key of the external entity. N.b. There is no SAML token.
2. The response is sent using SOAP over HTTPS to the federating identity PEP.
3. The federating entity PEP verifies the signature using the external entity public key.
4. The federating entity PEP directs verifies the signature and forwards the message to a federating entity dispatcher service.
5. The federating entity dispatcher service processes the response. This is not further described as it is implementation dependent and the persistence and correlation management required is outside the scope of this document.

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]):

Minimum Profile	DAIL Part Profile	Mandatory data (not exported)	Description	inetOrgPerson mapping	Extended Class
hmaId			Unambiguous HMA identity	uid	
c			Country of origin	homePostalAddress	
o			Organisation	o	
hmaProjectName			Names of projects with which user is affiliated.		hmaProjectName
hmaAccount			The HMA account number		hmaAccount
hmaServiceName			Associated services		hmaServiceName
	userProfile		Commercial/GMES/scientific		userProfile
	email		Email address	mail	
		password		password	
		state	Enabled/disabled. This information allows an administrator to		state

			disable a specific user.		
	homePostalAddress		Home address	homePostalAddress	
	IdP		Identity provider		IdP

Table 1: Attributes in SAML Token

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.

The SAML token content should be encrypted within <xenc> element of WS-Security.

The WS-Security specification uses the XML Encryption standard. Two elements <xenc:ReferenceList> and <xenc:EncryptedKey>can be used within the <wsse:Security> header block. When a client encrypts portion(s) of a SOAP message using XML Encryption it MUST prepend a sub-element to the <wsse:Security> header block. The sub-element MUST contain the information necessary for the recipient to identify the portions of the message that it is able to decrypt.

SAML assertion references can occur as encrypted content within the <xenc:EncryptedData> elements referenced by Id from the <xenc:DataReference> elements of <xenc:ReferenceList>. An <xenc:ReferenceList> element which identifies the encrypted content should occur as a toplevel element in a <wsse:Security> header.

Example:

```
<S11:Envelope xmlns:S11="..." xmlns:wsse="..." xmlns:wsu="..."
xmlns:ds="..." xmlns:xenc="...">
<S11:Header>
<wsse:Security>
<xenc:ReferenceList xmlns:xenc="...">
<xenc:DataReference URI="#EncryptedSTR1"/>
</xenc:ReferenceList>
</wsse:Security>
</S11:Header>
<S11:Body>
```

```
<xenc:EncryptedData wsu:Id=”EncryptedSTR1”>
<xenc:CipherData>
<xenc:CipherValue>...</xenc:CipherValue>
</xenc:CipherData>
</xenc:EncryptedData>
</S11:Body>
</S11:Envelope>
```

Figure 7 Encryption in WS-Security

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.

d. Include a timestamp in the signed message to prevent replay attacks.

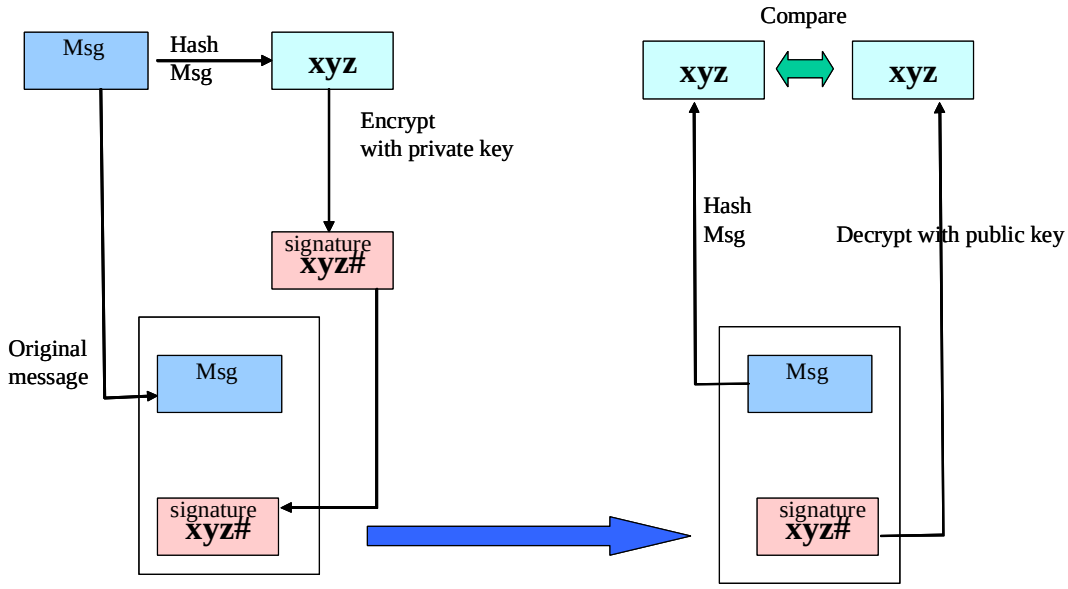

Figure 8: Digital Signature

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:

- Sender : Hash and signs (encrypts the hash code)
- Receiver : Hash and verify hash (decrypts the hash)

This ensures that the message was sent by a known client and that the message arrived intact.

An example of the <ds:Signature> element of WS-Security is given below.

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<wsse:Security
xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd">
<Assertion xmlns="http://earth.esa.int/um/eop/saml"
xmlns:saml=" urn: oasis:names:tc:SAML:1.0:assertion
">          <saml:Assertion
xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
xmlns="urn:oasis:names:tc:SAML:1.0:assertion"
AssertionID="oracle.security.xmlsec.saml.Assertion@baa6ba"
IssueInstant="2009-02-02T16:02:45Z" Issuer="http://www.spacebel.be"
MajorVersion="1" MinorVersion="1">
             <saml:Conditions NotBefore="2009-02-02T15:02:45Z"
NotOnOrAfter="2009-02-03T16:02:45Z"/>
             <saml:AuthenticationStatement AuthenticationInstant="2009-02-
02T16:02:45Z"
AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password">
               <saml:Subject>
                 <saml:NameIdentifier>Alexandre
CUCUMEL</saml:NameIdentifier>
                 <saml:SubjectConfirmation>

<saml:ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:bearer</saml:Confirm
ationMethod>
                 </saml:SubjectConfirmation>
               </saml:Subject>
             </saml:AuthenticationStatement>
             <saml:AttributeStatement>
               <saml:Subject>
                 <saml:NameIdentifier>Alexandre
CUCUMEL</saml:NameIdentifier>
                 <saml:SubjectConfirmation>

<saml:ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:bearer</saml:Confirm
ationMethod>
                 </saml:SubjectConfirmation>
               </saml:Subject>
               <saml:Attribute AttributeName="mail">

<saml:AttributeValue>alexandre.cucumel@spacebel.be</saml:AttributeValue>
               </saml:Attribute>
               <saml:Attribute AttributeName="postaladdress">
                 <saml:AttributeValue>Belgium</saml:AttributeValue>
               </saml:Attribute>
               <saml:Attribute AttributeName="o">
                 <saml:AttributeValue>Spacebel</saml:AttributeValue>
               </saml:Attribute>
               <saml:Attribute AttributeName="sn">
                 <saml:AttributeValue>acl</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:Transforms>
                     <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>dYhSU2Whd8Dt65hLrj/HYBQPk9I=</ds:DigestValue>
                 </ds:Reference>
               </ds:SignedInfo>

<ds:SignatureValue>DqhoRN9hd3tyX33dqm83wdMS5UURSEZBH5gFHtcBa/cU5RSyupongXcPm
rRrn4pwoWxpz+lmZESgSo3x76UZZlsIS8Fy34ed1EDcdFqXUbCoTdGoB1jm+4hnTndRHimIiQArr
QiOBJ9BdAYm7028kdrbR/JGzEqhm7Ej4AmRgTg=</ds:SignatureValue>
               <ds:KeyInfo>
                 <ds:X509Data>

<ds:X509Certificate>MIICXjCCAccCBEhiBKYwDQYJKoZIhvcNAQEEBQAwdjELMAkGA1UEBhMC
QkUxETAPBgNVBAgTCEJlbGdpcXVlMRIwEAYDVQQHEwlCcnV4ZWxsZXMxETAPBgNVBAoTCFNwYWNl
YmVsMREwDwYDVQQLEwhCVSBTUEFDRTEaMBgGA1UEAxMRQWxleGFuZHJlIENVQ1VNRUwwHhcNMDgw
NjI1MDg0MTEwWhcNMDgwOTIzMDg0MTEwWjB2MQswCQYDVQQGEwJCRTERMA8GA1UECBMIQmVsZ2lx
dWUxEjAQBgNVBAcTCUJydXhlbGxlczERMA8GA1UEChMIU3BhY2ViZWwxETAPBgNVBAsTCEJVIFNQ
QUNFMRowGAYDVQQDExFBbGV4YW5kcmUgQ1VDVU1FTDCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkC
gYEAq4Vsyd7w1y+u53a70YVPELIjP0S8Lyi+cuwRjySyxf3YYPdF+9BX2zfr/wGai0kYmkO1NzCW
DTNn0gsd/EjLq0eReHrqkIKm4hdbcrbA09A9kA1MKo9SYQKDVM7oXU5rHqdbPdntoM4H2QzrgL4f
yIV/Zimt31UrdgZ3ySEVb/0CAwEAATANBgkqhkiG9w0BAQQFAAOBgQAxwyN+gEsWh+tXEES9xUM/
BHZ3aUsdh427IlJabJe28rR7bqlC+glYZD8JkLzHOP3lsxtuxWyg9kXKk0SUwOAPC2IjOEwhhL7W
BhNpKYacrxmY6kgVe6g/DBlIPD+XRe4pCQMARfwX22CyufDVSm3AM1nJTwEcw5OkM4pWFEdcjg==
</ds:X509Certificate>
                 </ds:X509Data>
               </ds:KeyInfo>
             </ds:Signature>
           </saml:Assertion>
</Assertion>
</wsse:Security>
</soapenv:Header>
<soapenv:Body>
     <GetOptions xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://earth.esa.int/hma/ordering"
xsi:schemaLocation="http://earth.esa.int/hma/ordering ..\Order.xsd"
service="OS" version="1.2.0">
<collectionId>urn:HMA:EUM1.ESA.EECF.ENVISAT_ASA_IMx_xS</collectionId>
     </GetOptions>
</soapenv:Body>
</soapenv:Envelope>
```

Figure 9: Example Signature

7 Interface

7.1 Authenticate

The Authenticate operation allows clients to retrieve authentication metadata from a 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.

```
<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>User NAME</q0:username>
 <q0:password>******</q0:password>
</q0:Authenticate>
</soapenv:Body>
</soapenv:Envelope>
```

Figure 10: Example Authenticate Request

7.1.3 Response

The following XML shows an unencrypted example response

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
   <soapenv:Body>
      <ns:AuthenticateResponse xmlns:ns="http://earth.esa.int/um/eop">
         <ns:return>
            <Assertion xmlns="http://earth.esa.int/um/eop/saml"
xmlns:saml="Urn: oasis:names:tc:SAML:1.0:assertion">
               <saml:Assertion
xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
xmlns="urn:oasis:names:tc:SAML:1.0:assertion"
AssertionID="oracle.security.xmlsec.saml.Assertion@baa6ba"
IssueInstant="2009-02-02T16:02:45Z" Issuer="http://www.spacebel.be"
MajorVersion="1" MinorVersion="1">
                  <saml:Conditions NotBefore="2009-02-02T15:02:45Z"
NotOnOrAfter="2009-02-03T16:02:45Z"/>
                  <saml:AuthenticationStatement
AuthenticationInstant="2009-02-02T16:02:45Z"
AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password">
                     <saml:Subject>
                        <saml:NameIdentifier>Alexandre
CUCUMEL</saml:NameIdentifier>
                        <saml:SubjectConfirmation>

   <saml:ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:bearer</saml:Conf
irmationMethod>
                        </saml:SubjectConfirmation>
                     </saml:Subject>
```

```
                    </saml:AuthenticationStatement>
                    <saml:AttributeStatement>
                           <saml:Subject>
                                  <saml:NameIdentifier>Alexandre
CUCUMEL</saml:NameIdentifier>
                                  <saml:SubjectConfirmation>

   <saml:ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:bearer</saml:Conf
irmationMethod>
                                  </saml:SubjectConfirmation>
                           </saml:Subject>
                           <saml:Attribute AttributeName="mail">

   <saml:AttributeValue>alexandre.cucumel@spacebel.be</saml:AttributeValue>
                           </saml:Attribute>
                           <saml:Attribute AttributeName="postaladdress">

   <saml:AttributeValue>Belgium</saml:AttributeValue>
                           </saml:Attribute>
                           <saml:Attribute AttributeName="o">

   <saml:AttributeValue>Spacebel</saml:AttributeValue>
                           </saml:Attribute>
                           <saml:Attribute AttributeName="sn">

   <saml:AttributeValue>acl</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:Transforms>
                                                <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>dYhSU2Whd8Dt65hLrj/HYBQPk9I=</ds:DigestValue>
                                  </ds:Reference>
                           </ds:SignedInfo>

   <ds:SignatureValue>DqhoRN9hd3tyX33dqm83wdMS5UURSEZBH5gFHtcBa/cU5RSyupongX
cPmrRrn4pwoWxpz+lmZESgSo3x76UZZlsIS8Fy34ed1EDcdFqXUbCoTdGoB1jm+4hnTndRHimIiQ
ArrQiOBJ9BdAYm7028kdrbR/JGzEqhm7Ej4AmRgTg=</ds:SignatureValue>
                           <ds:KeyInfo>
                                  <ds:X509Data>

   <ds:X509Certificate>MIICXjCCAccCBEhiBKYwDQYJKoZIhvcNAQEEBQAwdjELMAkGA1UEB
hMCQkUxETAPBgNVBAgTCEJlbGdpcXVlMRIwEAYDVQQHEwlCcnV4ZWxsZXMxETAPBgNVBAoTCFNwY
WNlYmVsMREwDwYDVQQLEwhCVSBTUEFDRTEaMBgGA1UEAxMRQWxleGFuZHJlIENVQ1VNRUwwHhcNM
DgwNjI1MDg0MTEwWhcNMDgwOTIzMDg0MTEwWjB2MQswCQYDVQQGEwJCRTERMA8GA1UECBMIQmVsZ
2lxdWUxEjAQBgNVBAcTCUJydXhlbGxlczERMA8GA1UEChMIU3BhY2ViZWwxETAPBgNVBAsTCEJVI
FNQQUNFMRowGAYDVQQDExFBbGV4YW5kcmUgQ1VDVU1FTDCBnzANBgkqhkiG9w0BAQEFAAOBjQAwg
YkCgYEAq4Vsyd7w1y+u53a70YVPELIjP0S8Lyi+cuwRjySyxf3YYPdF+9BX2zfr/wGai0kYmkO1N
zCWDTNn0gsd/EjLq0eReHrqkIKm4hdbcrbA09A9kA1MKo9SYQKDVM7oXU5rHqdbPdntoM4H2Qzrg
L4fyIV/Zimt31UrdgZ3ySEVb/0CAwEAATANBgkqhkiG9w0BAQQFAAOBgQAxwyN+gEsWh+tXEES9x
UM/BHZ3aUsdh427IlJabJe28rR7bqlC+glYZD8JkLzHOP3lsxtuxWyg9kXKk0SUwOAPC2IjOEwhh
L7WBhNpKYacrxmY6kgVe6g/DBlIPD+XRe4pCQMARfwX22CyufDVSm3AM1nJTwEcw5OkM4pWFEdcj
g==</ds:X509Certificate>
```

```
                                    </ds:X509Data>
                                </ds:KeyInfo>
                            </ds:Signature>
                        </saml:Assertion>
                    </Assertion>
                </ns:return>
            </ns:AuthenticateResponse>
        </soapenv:Body>
</soapenv:Envelope>
```

Figure 11: Example Authenticate Response

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

7.2 AuthenticateFederated

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

7.2.1 Request

Protocol: SOAP over HTTPS

7.2.2 XML encoding

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

```
<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:AuthenticateFederated>
<q0:username>User NAME</q0:username>
 <q0:password>******</q0:password>
<q0:IdpName>spot</q0:IdpName>
</q0:AuthenticateFederated>
</soapenv:Body>
</soapenv:Envelope>
```

Figure 12: Example AuthenticateFederated Request

7.2.3 Response

The following XML shows an unencrypted example response. This is the structure of the both:

- the federated response message accepted by the DAIL authentication service and coming from an Idp (GS) and encrypted with the DAIL public key.
- The federated response message returned by the DAIL authentication service to a client.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
   <soapenv:Body>
      <ns:AuthenticateFederatedResponse
xmlns:ns="http://earth.esa.int/um/eop">
         <ns:return>
            <Assertion xmlns="http://earth.esa.int/um/eop/saml"
xmlns:saml="Urn: oasis:names:tc:SAML:1.0:assertion">
               <saml:Assertion
xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
xmlns="urn:oasis:names:tc:SAML:1.0:assertion"
AssertionID="oracle.security.xmlsec.saml.Assertion@baa6ba"
IssueInstant="2009-02-02T16:02:45Z" Issuer="http://www.spacebel.be"
MajorVersion="1" MinorVersion="1">
                  <saml:Conditions NotBefore="2009-02-02T15:02:45Z"
NotOnOrAfter="2009-02-03T16:02:45Z"/>
                  <saml:AuthenticationStatement
AuthenticationInstant="2009-02-02T16:02:45Z"
AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password">
                     <saml:Subject>
                        <saml:NameIdentifier>Alexandre
CUCUMEL</saml:NameIdentifier>
                        <saml:SubjectConfirmation>

   <saml:ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:bearer</saml:Conf
irmationMethod>
                        </saml:SubjectConfirmation>
                     </saml:Subject>
                  </saml:AuthenticationStatement>
                  <saml:AttributeStatement>
                     <saml:Subject>
                        <saml:NameIdentifier>Alexandre
CUCUMEL</saml:NameIdentifier>
                        <saml:SubjectConfirmation>

   <saml:ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:bearer</saml:Conf
irmationMethod>
                        </saml:SubjectConfirmation>
                     </saml:Subject>
                     <saml:Attribute AttributeName="mail">

   <saml:AttributeValue>alexandre.cucumel@spacebel.be</saml:AttributeValue>
                     </saml:Attribute>
                     <saml:Attribute AttributeName="postaladdress">

   <saml:AttributeValue>Belgium</saml:AttributeValue>
                     </saml:Attribute>
                     <saml:Attribute AttributeName="o">

   <saml:AttributeValue>Spacebel</saml:AttributeValue>
                     </saml:Attribute>
                     <saml:Attribute AttributeName="sn">

   <saml:AttributeValue>acl</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:Transforms>
                                                <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>dYhSU2Whd8Dt65hLrj/HYBQPk9I=</ds:DigestValue>
                                    </ds:Reference>
                              </ds:SignedInfo>

   <ds:SignatureValue>DqhoRN9hd3tyX33dqm83wdMS5UURSEZBH5gFHtcBa/cU5RSyupongX
cPmrRrn4pwoWxpz+lmZESgSo3x76UZZlsIS8Fy34ed1EDcdFqXUbCoTdGoB1jm+4hnTndRHimIiQ
ArrQiOBJ9BdAYm7028kdrbR/JGzEqhm7Ej4AmRgTg=</ds:SignatureValue>
                              <ds:KeyInfo>
                                    <ds:X509Data>

   <ds:X509Certificate>MIICXjCCAccCBEhiBKYwDQYJKoZIhvcNAQEEBQAwdjELMAkGA1UEB
hMCQkUxETAPBgNVBAgTCEJlbGdpcXVlMRIwEAYDVQQHEwlCcnV4ZWxsZXMxETAPBgNVBAoTCFNwY
WNlYmVsMREwDwYDVQQLEwhCVSBTUEFDRTEaMBgGA1UEAxMRQWxleGFuZHJlIENVQ1VNRUwwHhcNM
DgwNjI1MDg0MTEwWhcNMDgwOTIzMDg0MTEwWjB2MQswCQYDVQQGEwJCRTERMA8GA1UECBMIQmVsZ
2lxdWUxEjAQBgNVBAcTCUJydXhlbGxlczERMA8GA1UEChMIU3BhY2ViZWwxETAPBgNVBAsTCEJVI
FNQQUNFMRowGAYDVQQDExFBbGV4YW5kcmUgQ1VDVU1FTDCBnzANBgkqhkiG9w0BAQEFAAOBjQAwg
YkCgYEAq4Vsyd7w1y+u53a70YVPELIjP0S8Lyi+cuwRjySyxf3YYPdF+9BX2zfr/wGai0kYmkO1N
zCWDTNn0gsd/EjLq0eReHrqkIKm4hdbcrbA09A9kA1MKo9SYQKDVM7oXU5rHqdbPdntoM4H2Qzrg
L4fyIV/Zimt31UrdgZ3ySEVb/0CAwEAATANBgkqhkiG9w0BAQQFAAOBgQAxwyN+gEsWh+tXEES9x
UM/BHZ3aUsdh427IlJabJe28rR7bqlC+glYZD8JkLzHOP3lsxtuxWyg9kXKk0SUwOAPC2IjOEwhh
L7WBhNpKYacrxmY6kgVe6g/DBlIPD+XRe4pCQMARfwX22CyufDVSm3AM1nJTwEcw5OkM4pWFEdcj
g==</ds:X509Certificate>
                                    </ds:X509Data>
                              </ds:KeyInfo>
                        </ds:Signature>
                     </saml:Assertion>
                  </Assertion>
               </ns:return>
         </ns:AuthenticateFederatedResponse>
   </soapenv:Body>
</soapenv:Envelope>
```

Figure 13: Example AuthenticateFederated Response

7.2.4 Failed AuthenticateFederated Request

See Authenticate request.

7.2.5 WSDL

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

```
<?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:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:ns1="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"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
targetNamespace="http://earth.esa.int/um/eop">
<wsdl:documentation>AuthenticationService</wsdl:documentation>
<wsdl:types>
<xs:schema attributeFormDefault="qualified" elementFormDefault="qualified"
targetNamespace="http://earth.esa.int/um/eop">
<xs:element name="Authenticate">
<xs:complexType>
<xs:sequence>
  <xs:element minOccurs="0" name="username" nillable="true" type="xs:string"
/>
  <xs:element minOccurs="0" name="password" nillable="true" type="xs:string"
/>
  </xs:sequence>
  </xs:complexType>
  </xs:element>
<xs:element name="AuthenticateResponse">
<xs:complexType>
<xs:sequence>
  <xs:element minOccurs="0" name="return" nillable="true" type="xs:anyType"
/>
  </xs:sequence>
  </xs:complexType>
  </xs:element>
<xs:element name="AuthenticateFederated">
<xs:complexType>
<xs:sequence>
  <xs:element minOccurs="0" name="username" nillable="true" type="xs:string"
/>
  <xs:element minOccurs="0" name="password" nillable="true" type="xs:string"
/>
  <xs:element minOccurs="0" name="IdpName" nillable="true" type="xs:string"
/>
  </xs:sequence>
  </xs:complexType>
  </xs:element>
<xs:element name="AuthenticateFederatedResponse">
<xs:complexType>
<xs:sequence>
  <xs:element minOccurs="0" name="return" nillable="true" type="xs:anyType"
/>
  </xs:sequence>
  </xs:complexType>
  </xs:element>
<xs:element name="decryptAndCheckSignature">
<xs:complexType>
<xs:sequence>
  <xs:element minOccurs="0" name="input" nillable="true" type="xs:anyType"
/>
  </xs:sequence>
  </xs:complexType>
  </xs:element>
<xs:element name="decryptAndCheckSignatureResponse">
<xs:complexType>
<xs:sequence>
  <xs:element minOccurs="0" name="return" nillable="true" type="xs:anyType"
/>
  </xs:sequence>
  </xs:complexType>
  </xs:element>
  </xs:schema>
  </wsdl:types>
```

```
<wsdl:message name="AuthenticateRequest">
  <wsdl:part name="parameters" element="ns:Authenticate" />
  </wsdl:message>
<wsdl:message name="AuthenticateResponse">
  <wsdl:part name="parameters" element="ns:AuthenticateResponse" />
  </wsdl:message>
<wsdl:message name="decryptAndCheckSignatureRequest">
  <wsdl:part name="parameters" element="ns:decryptAndCheckSignature" />
  </wsdl:message>
<wsdl:message name="decryptAndCheckSignatureResponse">
  <wsdl:part name="parameters" element="ns:decryptAndCheckSignatureResponse"
/>
  </wsdl:message>
<wsdl:message name="AuthenticateFederatedRequest">
  <wsdl:part name="parameters" element="ns:AuthenticateFederated" />
  </wsdl:message>
<wsdl:message name="AuthenticateFederatedResponse">
  <wsdl:part name="parameters" element="ns:AuthenticateFederatedResponse" />
  </wsdl:message>
<wsdl:portType name="AuthenticationServicePortType">
<wsdl:operation name="Authenticate">
  <wsdl:input message="ns:AuthenticateRequest"
wsaw:Action="urn:Authenticate" />
  <wsdl:output message="ns:AuthenticateResponse"
wsaw:Action="urn:AuthenticateResponse" />
  </wsdl:operation>
<wsdl:operation name="decryptAndCheckSignature">
  <wsdl:input message="ns:decryptAndCheckSignatureRequest"
wsaw:Action="urn:decryptAndCheckSignature" />
  <wsdl:output message="ns:decryptAndCheckSignatureResponse"
wsaw:Action="urn:decryptAndCheckSignatureResponse" />
  </wsdl:operation>
<wsdl:operation name="AuthenticateFederated">
  <wsdl:input message="ns:AuthenticateFederatedRequest"
wsaw:Action="urn:AuthenticateFederated" />
  <wsdl:output message="ns:AuthenticateFederatedResponse"
wsaw:Action="urn:AuthenticateFederatedResponse" />
  </wsdl:operation>
  </wsdl:portType>
<wsdl:binding name="AuthenticationServiceSoap11Binding"
type="ns:AuthenticationServicePortType">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http"
style="document" />
<wsdl:operation name="Authenticate">
  <soap:operation soapAction="urn:Authenticate" style="document" />
<wsdl:input>
  <soap:body use="literal" />
  </wsdl:input>
<wsdl:output>
  <soap:body use="literal" />
  </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="decryptAndCheckSignature">
  <soap:operation soapAction="urn:decryptAndCheckSignature" style="document"
/>
<wsdl:input>
  <soap:body use="literal" />
  </wsdl:input>
<wsdl:output>
  <soap:body use="literal" />
  </wsdl:output>
  </wsdl:operation>
<wsdl:operation name="AuthenticateFederated">
  <soap:operation soapAction="urn:AuthenticateFederated" style="document" />
<wsdl:input>
  <soap:body use="literal" />
  </wsdl:input>
```

```
<wsdl:output>
  <soap:body use="literal" />
  </wsdl:output>
  </wsdl:operation>
  </wsdl:binding>
<wsdl:service name="AuthenticationService">
<wsdl:port name="AuthenticationServiceHttpSoap11Endpoint"
binding="ns:AuthenticationServiceSoap11Binding">
  <soap:address
location="http://dail.esa.int/AxisService/services/AuthenticationService.Aut
henticationServiceHttpSoap11Endpoint" />
  </wsdl:port>
  </wsdl:service>
  </wsdl:definitions>
```

Figure 14: Authentication Service WSDL

7.3 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..

7.3.1 Request

Protocol: SOAP plus WS-Security over HTTPS.

7.3.2 XML encoding

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

```
<soapenv:Envelope:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
:xsd="http://www.w3.org/2001/XMLSchema"
:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header>
<wsse:Security="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-
1.0.xsd":soap="http://schemas.xmlsoap.org/soap/envelope/":saml="urn:oasis:na
mes:tc:SAML:1.0:assertion":wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
        <Assertion xmlns="http://earth.esa.int/um/eop/saml">
          <xenc:EncryptedData:xenc="http://www.w3.org/2001/04/xmlenc#"
="http://www.w3.org/2001/04/xmlenc#Content">
            <xenc:EncryptionMethod="http://www.w3.org/2001/04/xmlenc#aes128-
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>ooKS1gS5Gg1NlEWtHgTq/r45llq5UFIk/R2c0aDceXDdAQ2QY9Tmdqxzkm
yVxe0vRPiJg5Sutg0q/zuswp+TmOIem2AaBR3txNKdFD991jZx+3+ehzaur/vRr8BLiPxjhJ8S3H
EkxXnwK2iSnnKkyn2m7co7yCYjs+//lUfWpKU=</xenc:CipherValue>
                </xenc:CipherData>
```

```
            </xenc:EncryptedKey>
          </ds:KeyInfo>
          <xenc:CipherData>

<xenc:CipherValue>O/GO7uJ3AiNgD2dxkKGzs9wG9OI1ZfGFH8HFkLIoHst3S1i+sDrJQANq2Y
O0MhIcF0FPAWjvrios8ZCrVmeEH37wiYhkHJDj02077vJ7U9/+4Ce6EX1+eF7apk9Cb0Vm/GnCMv
R4wQm5NJIlvto7PuNWjWjNq8gg3eEG69NVW9C7KsTzeK3g+OQKHdm5We2uZKk7/61O4ED/9nHzdJ
pP10q7xwqQwZTwR4v84ljpOSKnNoF6AeZ1KsQ62kZEVIm4ZBL9Cftlyv1nZk09KLGJgMUlF+JMlS
cLmGVjMrTrZF98AH8vwqWChNMrhpxuGgJFYtV9MXtVf9aL0ggB9f/G/kKnrHdKjzEZITUrR4bNE1
sJcwEUjmyBdCbL9xsB9IY0bsu8Y3ssU+dEPQJDCSiuWMfTCihnyCWw7TjIf/JMUMvktHFq9qKecp
awlpSRbXBxifmYp8jKx6fK9Qb1yAAh6190rdDlqVzvYpRHzogxTQS5cQUnjBY5bv1ljXNxlvExsq
ldIU9HId7dQEi5CP4+xt4TcLeV2+nYOTD5p2ZiY3b+wc75+NIFLMBNflMdkO86rCSjv7uApxXu8e
p9rGaHIoip0vgjh9TAa/gSmsD0Y/FBSMgF5/GQDmxaOhxrd6swcls3Prqfx/K8vNL9ab0Y7dl848
ZWlPxkIPmtfv3+HtUP4JlgeMa9t7g2Vb3+aM6V5jien8JAGUnO/a4h+g3ek/eb8belwh9Iubxkah
whs2mKREL1W7bD2aWay2Yabh9zclWb7NpB3C7z/+HCbycFYrChGeh1QUlZU/ubcu2H82rxP2tpV+
aTDjSCraPPX37759xCPLk7QTUR+pDUMoG3IFgzGSjkBdlXPzWEGPSqSv008CIB0GKASjwxiywbK0
5w/zCv1JzhipIEnQgc94j0oFrz4peoYDnSqEu8DtFjjbzIoPC/B5LndVwSuVwSw5M3M4IRPP6hbx
KL66/o6qEKF8QY+igTgEzrDsg1+bhib4gHRyW6+ivBpAZWvmVQ29dCUxpsie2C6WMwnaERSIc3z5
DOFEFC74ngo5kOHANrAOp2Q/tO9lbnqCWHi3R0Sz4NMOO2Men/aU1GZ9+pAHyP0fQBHofQQnzJyT
0S7rBuY0i90AZrkZ32Dgpy6dWBxNRqMh2uLuhof+u8huBQZDK9Tdxe+GNzc+QZboYpU6vX57cS4C
7uRZJYVNlQQNmkhxJCaiSWR/cqfn5C2wj3hte4pxjVSsYm9tciZwuhsOSjz2BNjoM+oli+fdsFLh
08EQoQhsrfMFCjPbNaAz3NcSN3zZn96uxQDs+VU1trxjFHxWSbFcPVqRB5CtmvfyGtDluzm7CO0Y
lDXLQmtL+opF3rAXLgMVhWnbrB6n6hTzSQ7jBMgveY1/DoUFOPrc5DBKu9C4+72PyPDRTeglx2XV
4VtHL5xBXGaCfnxmML1CLlS8EpYvXWSM98vVaPCYboHTgKqNAr9HEz9Oa1CArDVL6+o/MjxLVsJx
SsAF+A6nHbTqQuMDjcgZH+5ePp40Eny2kQuwa62ur3d7VszIktMxEH1pRQHKNo6M9PCIRg1txkOv
95ikYoYNOTPo/cDuwNLhb10XCsrBH77s9woyJzZKVez5ekgQigESSDEzgiYaphwQhNrT+L58jYkb
Xh9wgzKAVsZuLLtdSF12BuU9TjpZhnIVA83YIC9s0bGtb+8zJM+PpT08txEE6EAv5wHCLHYhOteo
6pBPn+fuXzuw1nUdo1bsygk4mBKB/ExXcpkT8/FA2yxAy2fmYhinP6bhFdxstFP0jtgXobNcPTs6
tJxGdShLJeJqtGau7MdVv+UcFaUsuvB/6lB7kTuwsyxkHKFht6FGht6JajzKCNVn4dQXfMz7gosR
U69E14FAXQLy4alcw+RLlooTAEPddyGKEy7EzPE8a1YRAgaRMlZmTdfbe5TQSJ/fQ1HF2R6xhyK2
ThANaU8FiCXTKip1pQnuY53bAffF+gcJvLqKZfJruldt3Dvr03rWaBmIWcGK2ThANaU8FiCXTKip
1pQnuY53bAffF+mYhTq1JO2oI8iHsNYq3tshxigniMEhK2ThANaU8FiCXTKip1pQnuY53bAffF+f
XsuaLLQdGnKH2QYgj9A/iQlhVJfVTK2ThANaU8FiCXTKip1pQnuY53bAffF+VWk3iJ/uV6aV6adb
9RvaJgHPyDgg+CK2ThANaU8FiCXTKip1pQnuY53bAffF+ZgC1hLwEjCaaWcvidCf7AAZOCfDtj7K
2ThANaU8FiCXTKip1pQnuY53bAffF+qlVWqfrf1dH0Ut0DNk8pugI/iV+0iVK2ThANaU8FiCXTKi
p1pQnuY53bAffF+ysF8RnjR/51FU+fl6rl8nVqMf8UKKYK2ThANaU8FiCXTKip1pQnuY53bAffF+
Cia1MGGoe5VqurHiYx/vnQtLS1Ro5EK2ThANaU8FiCXTKip1pQnuY53bAffF+o9cdJkmpmg7EbPF
r+lvogWqFAmtTkGK2ThANaU8FiCXTKip1pQnuY53bAffF+t0o+U2t3urWrjnmOXHfMQm02shw/Tg
K2ThANaU8FiCXTKip1pQnuY53bAffF+R4mbe0KGYfFnm+1SwJrnWZXamXCSCuK2ThANaU8FiCXTK
ip1pQnuY53bAffF+V+ZcZl4ju8EOtSgOpC8tYakkKSWAV9K2ThANaU8FiCXTKip1pQnuY53bAffF
+2aIgTlfmQp2/UIoSaXW79MyUe/SLxnK2ThANaU8FiCXTKip1pQnuY53bAffF+FCNbKSz+CCj0cP
0zhlsE0v8o8cB/2zK2ThANaU8FiCXTKip1pQnuY53bAffF+fZxdnsN3wYEnCe1cprGwiWQZ01BSe
iK2ThANaU8FiCXTKip1pQnuY53bAffF+UMJwNJBvzm6qgeUqgteAark4Euz2veK2ThANaU8FiCXT
Kip1pQnuY53bAffF+a5bq7hY97DTdc8xt1ZxFxflM3p1iK+K2ThANaU8FiCXTKip1pQnuY53bAff
F+jOhrdRHQwQq67dh8JVKQwqQXB67dyhK2ThANaU8FiCXTKip1pQnuY53bAffF+W2QJZxKNSR5Lr
TyCYQ/VgRlUJr2CdzK2ThANaU8FiCXTKip1pQnuY53bAffF+QJTtT/tf+uw1dH/Ov6IutLpJZtqX
nSK2ThANaU8FiCXTKip1pQnuY53bAffF+TPqSWHJ32FeIYFCABcZ0Lt7nYjYU77K2ThANaU8FiCX
TKip1pQnuY53bAffF+6j1sOtygHwM+FTC9qCb3Nq1RLu4ZnSK2ThANaU8FiCXTKip1pQnuY53bAf
fF+luotouLGBlwiFpMKW5CmlDF9CXfxz1K2ThANaU8FiCXTKip1pQnuY53bAffF+Q2TWj/oDXnwb
DVkTlUoDSOfbVi9mX9K2ThANaU8FiCXTKip1pQnuY53bAffF+rCMxJIGsPuDatzNoRcLQxLmO12X
VV8K2ThANaU8FiCXTKip1pQnuY53bAffF+VhNoeKc4Ng6wDzEtzFv/M01+WAtPLcK2ThANaU8FiC
XTKip1pQnuY53bAffF+zWn5yYxzUP+TxT465upqcsYKGaFdTJK2ThANaU8FiCXTKip1pQnuY53bA
ffF+PASM+s4asTyHXtB0X2kgL7r2Ykecu2K2ThANaU8FiCXTKip1pQnuY53bAffF+qcjNGjETysR
m9IZDeXQR8xEjBw5/DvK2ThANaU8FiCXTKip1pQnuY53bAffF+4sMjgDscmlUan765FJf2SDKe7E
4rtCK2ThANaU8FiCXTKip1pQnuY53bAffF+gRngppF/cN94F7Tclp1UzfRiiyr9+ZK2ThANaU8Fi
CXTKip1pQnuY53bAffF+CsuxTzLxl9u6/IDnCxcWrIfVIFwAf83oGyF6GcQU83IWUrxZjJjgtRoS
XRWRhyQj8XCxzslge2rj9mxhOFHuRtoCs6qw/OleLhIEnWgoFgJOfx/yQEsfIbJfDGo/aQC9UEsM
Fwxkq0NfPupeSWx43orxlUXoitSd/CxY1rn6S9facWAP8XAjlZ+IyxfAnjaqeIu8FPwMzoR7UuEe
npFglaFacIUsb2JEsbf3OI4Cw3/CG/8ztfYnHGfNjpGPZwAombtPNgFyC6hipMHLhCV+Qx9YsEEE
ZWnPXG+xEcyHqsqR6h2hqo5uBZrsadLvO2jK3g7+nTLvYlxNuis/3ERGW4AfQ9ybdvLh2hae/5Qw
iKlmZPwQWxR7dDQr8NXZh+D7I7msXoLPFm3FMg00cx/lWrLpYEye/aT3L/WAWp1tnjItPV7V4rIU
FUPlwWQURh4gdrnv1Of2lWoFlvBgZwn1GYZO1OTYDZouRorgrnlmgwNijx7QcSY8Dqv/4iom4G+1
hHyNdMGEN7fwaHaBKqXN72M/1SelWybhMDTiTSJSe1WWiftNcu3SXqBK9Aa5k/l5BNozYkuPZXnZ
kM2S+g8oayepHYssqtXqYo4o+lIWgc/SI8VURIWw4U/ekDKyir45fUpJsIlckdJ+HsRLrUAupgIm
/HvFZ69B2VaU8d9vJ7dXNcSN/tNDPXZVnQfzv6V6wOx2hfICwDmrrQofXlNO4Jw9AugNDwVVutRJ
AMtgmnnDb74bxHpbgMTzLrFg4++TbyOP5VAd5W+QdbHRIMsaLwC87HIgPSjjqGCsgjg8HST4GRfw
dBTy7mlh2ws6ISptyHNJseJfUkDQ4ww/d6ejEfLS6UbAeB/hYJqLWudOdQGjfGAKbtOw9vnPeXvh
```

```
KCRs4zSe9r6iTKvg7QCzwAWiFrHgRN3UcmyGD/Or1fWr1p1umFblnQXUQvyBd2yKjDpbiDisKCm5
PJigpQW9tHO/dPcQu4w9GKJGn58eYh0b8
Rz5Hyu0UMbwrumcgthnz7Ql0GkEDyetQwcjhjmPih3P2QIGHNtf3PPQKPXWisK4Ayy1PaVMCm6Vq
X9d7bFNESOJRC48uOsrV8VFbIZ/UTpSEz+N+xtRH9W4L9ymVdFhd2cBXkKs2Z6nCPQX1cH28u5Xv
FR0ZyaBrm8t32XwQtQg6jpUTg/Y2r+GEU70wcLeURE4SqcQqctFIdbFW/UYpLuzmtWQRdAmVqbuR
8lTe8AkRO8ZI79ttshJzSdG7orjq5Xp+2A35d4OjJeiyygOtgUtrF1BD0vX3zKkJzBDBZ4jyD7YV
QencNjNzR0TzaTANW8rADboYz/N12Xv6SPiM3JYQSY/Qqx/bGU6EzYqvnH2MYEA2B5FutEAnajHN
R3FOans7tL3XeNniT4DxZxlm/1Ez6DrtO+76lB9YWB6WSSnPb8g/O7gfbXn8ddj05bMPq+RqBxkl
IBpj1PSdpAeF1USI8BB4sGrLB3xqm+EHDgIAV3Big4bQssCh22KGEMDdZqMUNKHwU/peTqhL5G41
YLjQ5ADpcK8VSkT1I2PkVjsft0K3R1JdWbKA1XwVjEYaZdRFkciF3vt194hV4GXTnxGpbqm+qhhD
VpkEr7aJNPGWAnbA==</xenc:CipherValue>
           </xenc:CipherData>
         </xenc:EncryptedData>
       </Assertion>
</wsse:Security>
</soapenv:Header>
<soapenv:Body>
    <GetOptions xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
="http://earth.esa.int/hma/ordering"
:schemaLocation="http://earth.esa.int/hma/ordering ..\Order.xsd" ="OS"
version="1.2.0">
<collectionId>urn:HMA:EUM1.ESA.EECF.ENVISAT_ASA_IMx_xS</collectionId>
    </GetOptions>
</soapenv:Body>
</soapenv:Envelope>
```

Figure 15: Service Request Example

7.3.3 Failed Request

An example is given below:

```
<?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>
```

7.4 ServiceResponse

The 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 authentication .

7.5 Sequence Diagrams

The interactions of each of the components are shown in the following sequence diagrams.

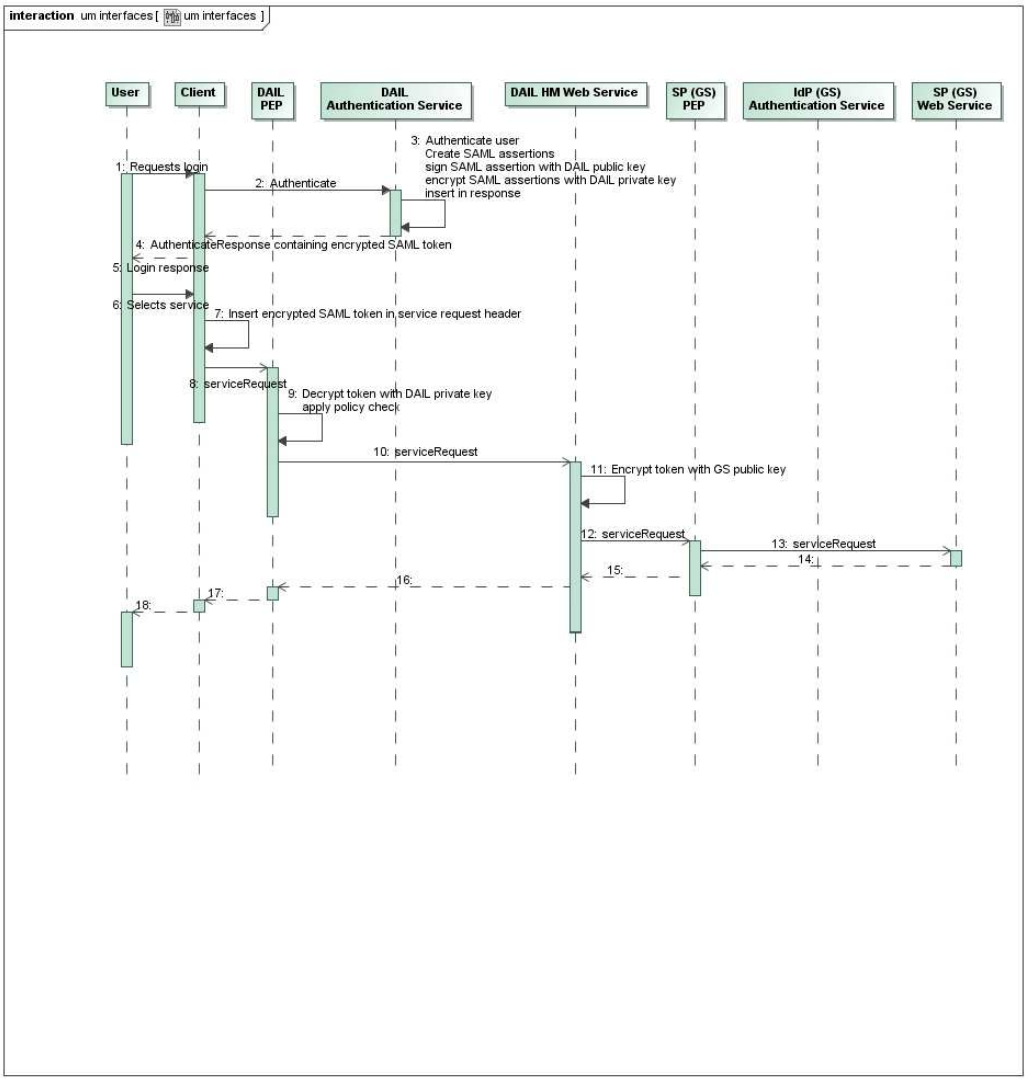

Figure 16 Sequence Diagram Showing Synchronous Request

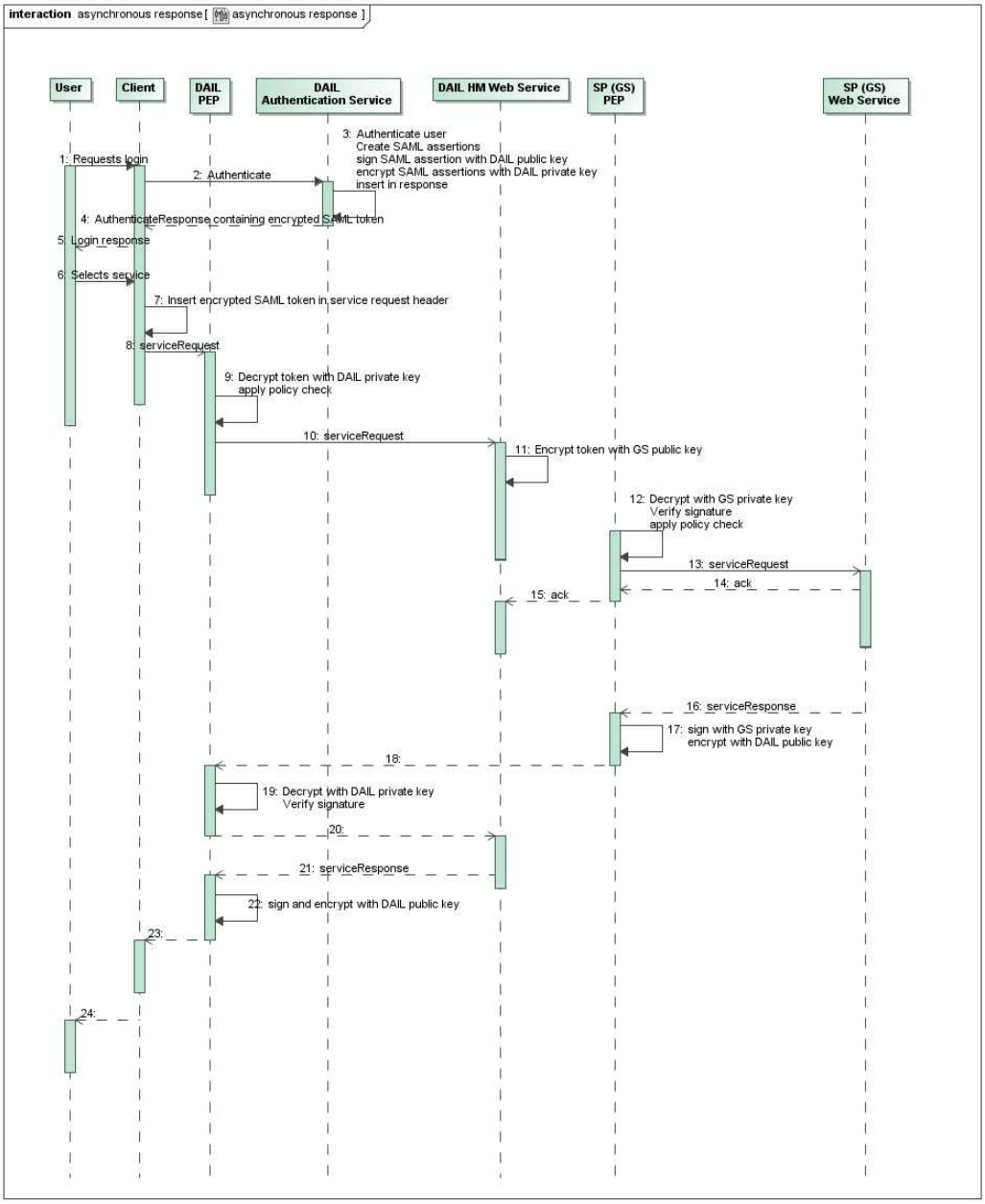

Figure 17 Sequence Diagram showing asynchronous request