Open Geospatial Consortium

Date: 2011-01-03

Reference number of this document: OGC 10-192

Category: Engineering Report

Editor(s): Jeff Harrison

OGC® Authentication for OGC Web Services

Warning

This document is not an OGC Standard. This document is an OGC Public Engineering Report created as a deliverable in an OGC Interoperability Initiative and is not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, any OGC Engineering Report should not be referenced as required or mandatory technology in procurements.

Document type:	OpenGIS® Engineering Report
Document subtype:	NA
Document stage:	Approved for public release
Document language:	English

License Agreement

Contents

OGC® Authentication for OGC Web Services

1 Introduction

1.1 Scope

Open geospatial services based on OGC® standards are strongly influencing development of spatial data infrastructures (SDI) around the world. These efforts have matured to a point where broad acceptance is now dependent on the capacity to secure online resources. In fact, organizations that are considering participation in SDI based on OGC web services must also consider how they can establish basic security frameworks for online resources. These requirements will continue to increase as data access transitions into collaborative data management with services like the Web Feature Service - Transactional (WFS-T) where parties collaborate on maintenance of shared geospatial data resources.

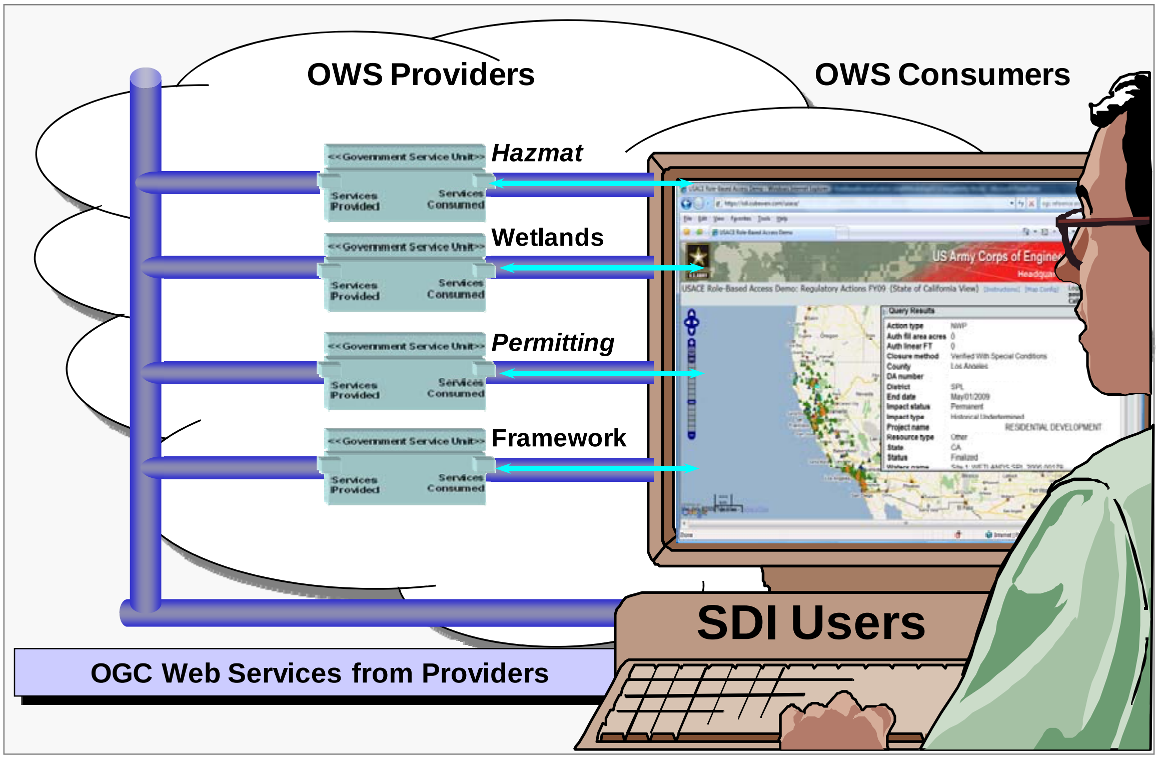


Figure 1 – Authentication for OGC Web Services is required to support SDI around the world

To help address these needs the OGC Authentication Interoperability Experiment (Auth IE) tested and documented ways of transferring basic authentication information between OGC clients and OGC services by leveraging mechanisms already in existing protocols (ex. HTTP Authentication).

Results of the Auth IE are presented in this Engineering Report document and serve as guidance to both implementers and organizations deploying solutions that involve basic authentication. It is the belief of the Auth IE participants that if such a document is made available to the community more OGC implementing products will natively support authentication.

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

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

1.2 Document contributor contact points

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

Name	Organization	Phone	Email
Ralph Baehre	LIO		ralph.baehre@ontario.ca
Nuke Goldstein	The Carbon Project	781-270-0674	ngoldstein@thecarbonproject.com
Chris Higgins	EDINA National Data Centre		chris.higgins@ed.ac.uk
Jeff Harrison	CubeWerx USA, The Carbon Project	703-491-9543	jharrison@cubewerx.com jharrison@thecarbonproject.com
Christian Kiehle	lat/lon GmbH	0049-228-184960	kiehle@lat-lon.de
Cristian Opincaru	Secure Dimensions		cristian.opincaru@secure-dimensions.de
Glenn Stowe	CubeWerx	819-771-8303	gstowe@cubewerx.com
David Wesloh	National Geospatial Intelligence Agency	314-676-0296	David.g.wesloh@nga.mil

1.3 Revision history

Date	Release	Editor	Primary clauses modified	Description
Feb. 2010	Version .1	Jeff Harrison	Initial ER	Feb. 2010
Mar. 2010	Version .2	Christian Kiehle	Added HTTP Auth. TIE	Mar. 2010
Mar. 2010	Version .2	Jeff Harrison	Edited HTTP Auth. TIE	Mar. 2010
Jun. 2010	Version .3	Jeff	Added WS	Jun. 2010

		Harrison	Security, SAML	
Nov. 2010	Version .34	Jeff Harrison	Comments from Christian Kiehle and Dave Wesloh	Nov. 2010

1.4 Future work

Improvements in this document are desirable as technology integration experiments for authentication methods continue in the OGC community.

1.5 Forward

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

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

2 References

The following documents are referenced in this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

OGC 06-121r3, OpenGIS® Web Services Common Standard

OGC 06-042, OpenGIS® Web Map Service Implementation Specification, Version 1.3.0

W3C, Hypertext Transfer Protocol - HTTP/1.0

3 Terms and definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Specification [OGC 06-121r3] shall apply. In addition, the definitions in W3C, Hypertext Transfer Protocol - HTTP/1.0 shall apply.

4 Conventions

4.1 Document terms and definitions

The following specification terms and definitions are used in this document:

a) shall – verb form used to indicate a requirement to be strictly followed to conform to this specification, from which no deviation is permitted

b) should – verb form used to indicate desirable ability or use, without mentioning or excluding other possibilities

c) may – verb form used to indicate an action permissible within the limits of this specification

d) can – verb form used for statements of possibility

e) informative – a part of a document that is provided for explanation, but is not required

f) normative – a part of a standards document that is required

g) annex – an auxiliary part of a document

5 Authentication for OGC Web Services Overview

This Authentication for OGC Web Services Engineering Report addresses standard ways of transferring basic authentication information between OGC clients and OGC services by leveraging mechanisms already in existing protocols.

As described in the OGC Reference Model, OGC services are defined using open non-proprietary Internet standards such as HTTP, URL, MIME, XML, WSDL or SOAP. The usual interaction between clients and services is request-response, where the client first makes a request which is transferred by either HTTP or SOAP to the service and then expects a response back. The pattern is depicted below.

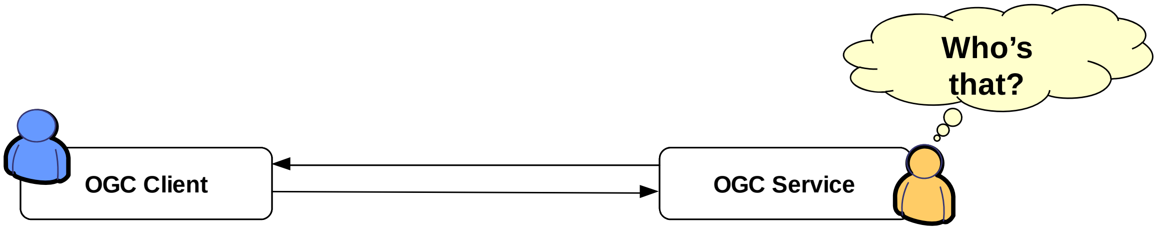


Figure 2 - Request-response interaction between OGC clients and OGC services

There are plenty of scenarios where the service needs to know the identity of the requesting party before processing the request. Such scenarios include typical authorization use-cases (different permissions are associated with different requesters) and auditing.

However, the OGC service specifications do not address the issue of transferring identity information together with the request. There have been many approaches in the past years to address this issue both within the OGC through the activities of the GeoDRM DWG, GeoRM SWG, Security DWG, the OWS-3, OWS-4, OWS-5, OWS-6 initiatives, as well as through different projects not organized by the OGC such as the US NSDI Role-based Access Control Project.

Although this issue is a “hot topic” for more than 5 years, there is no “best practice” of how to transfer basic authentication information between OGC clients and services. The consequence of this is that only a limited number of COTS products natively support basic security functions such as authentication and access control whenever security is involved.

There are various ways in which identity information can be transferred from the OGC client to the OGC service by leveraging the underlying protocols. Both HTTP and SOAP offer native support for embedding security information and there are several main-stream authentication protocols that leverage these features. Most importantly, by embedding the identity information in the transfer protocol the OGC service specifications are not touched at all, so the existing level of interoperability is not altered in any way.

6 Scope of the Authentication Interoperability Experiment

6.1 Issues In-Scope for the Auth IE

It is the scope of this Interoperability Experiment to test different standard ways of transferring identity information by means of embedding this information in the transport protocol. Results of these tests are provided in an engineering report documenting best practices for transferring identity information from OGC client components to OGC service components by leveraging the mechanisms available in the transport protocol.

The following are within the scope of this IE:

- To develop client and service components that demonstrate how identity can be transferred in an interoperable way, by means of the transport protocol;
- To investigate the requirements which are set on the OGC clients and services to enable support for these mechanisms;
- To document the findings in an official engineering report that shall be candidate for Best Practices document;
- To document the use-cases and scenarios, which are applicable to each authentication method and thus provide some guideline to organizations seeking to implement and deploy solutions that include authentication.
- Conduct Internet-based demonstrations of functioning client and service components as Technology Integration Experiments by volunteer participants.

As such, the following authentication methods were investigated in this interoperability experiment:

- HTTP Authentication
- WS Security with SOAP
- SAML
  - Web Browser
  - Desktop Client
  - Shibboleth with WAYF
  - Other Use Cases

6.2 Issues Out-of-Scope for the Auth IE

The following issues are explicitly left out of the scope of this Interoperability Experiment:

- Metadata describing authentication capabilities of services;
- Modifications of the existing OGC service specifications (to accommodate authentication issues);
- Authorization, audit or other security functions;
- Development of new authentication methods.

7 Use Cases and Results

As each authentication mechanism has its own interaction model, for each authentication method different use cases were implemented. This section describes the use cases for each of the investigated authentication methods and recommended authentication methods in the context of a Web Map Service (WMS). Other OGC web services may be added as engineering time and resources are available. Use cases were tested during Internet-based demonstrations of functioning client and service components as Technology Integration Experiments by volunteer participants and results documented. Results of the Auth IE were presented at the June 2010 OGC Technical Committee meetings in Silver Spring, MD, USA. Please note – service URLs are provided in several of the examples. The Auth IE participants cannot guarantee the services will be available at the time of the user’s reading of this document.

7.1 HTTP Authentication Use Case

HTTP provides a simple challenge-response authentication mechanism which may be used by a server to challenge a client request and by a client to provide authentication information.[1] It uses an extensible, case-insensitive token to identify the authentication scheme, followed by a comma-separated list of attribute-value pairs which carry the parameters necessary for achieving authentication via that scheme.

```
auth-scheme    = token
auth-param     = token "=" quoted-string
```

The 401 (unauthorized) response message is used by an origin server to challenge the authorization of a user agent. This response must include a `WWW-Authenticate` header field containing at least one `challenge` applicable to the requested resource.

```
challenge      = auth-scheme 1*SP realm *("," auth-param)
 realm          = "realm" "=" realm-value
 realm-value    = quoted-string
```

The realm attribute (case-insensitive) is required for all authentication schemes which issue a challenge. The realm value (case-sensitive), in combination with the canonical root URL of the server being accessed, defines the protection space. These realms allow the protected resources on a server to be partitioned into a set of protection spaces, each with its own authentication scheme and/or authorization database. The realm value is a string, generally assigned by the origin server, which may have additional semantics specific to the authentication scheme.

A user agent that wishes to authenticate itself with a server--usually, but not necessarily, after receiving a 401 response--may do so by including an `Authorization` header field with the request. The `Authorization` field value consists of `credentials` containing the authentication information of the user agent for the realm of the resource being requested.

```
credentials    = basic-credentials
              | ( auth-scheme #auth-param )
```

The domain over which credentials can be automatically applied by a user agent is determined by the protection space. If a prior request has been authorized, the same credentials may be reused for all other requests within that protection space for a period of time determined by the authentication scheme, parameters, and/or user preference. Unless otherwise defined by the authentication scheme, a single protection space cannot extend outside the scope of its server. If the server does not wish to accept the credentials sent with a request, it should return a 403 (forbidden) response.

[1] http://www.w3.org/Protocols/HTTP/1.0/draft-ietf-http-spec.html#AA

The HTTP protocol does not restrict applications to this simple challenge-response mechanism for access authentication. Additional mechanisms may be used, such as encryption at the transport level or via message encapsulation, and with additional header fields specifying authentication information.

7.1.1 Assumptions and Interactions

Assumptions:

1. The client knows the URL of the service
2. The client is already known by the service
    a. User name and password has been distributed by some out-of-band mechanism

Interactions:

1. Client makes one standard OGC request to service
2. If no authentication information is supplied, the service answers with HTTP 401
3. Client makes new request including authentication information
4. Service verifies the authentication information, and serves the OGC request

7.1.2 Method for HTTP Authentication on Web Map Service (WMS)

The "basic" authentication scheme for an OGC Web Map Service (WMS) is based on the *Hypertext Transfer Protocol -- HTTP/1.0*[2] model which indicates that the user agent must authenticate itself with a user-ID and a password for each realm.

The realm value should be considered an opaque string which can only be compared for equality with other realms on that server. The server will authorize the request only if it can validate the user-ID and password for the protection space of the `Request-URI`. There are no optional authentication parameters.

Upon receipt of an unauthorized request for a URI within the protection space, the server should respond with a challenge like the following:

```
WWW-Authenticate: Basic realm="WallyWorld"
```

where "WallyWorld" is the string assigned by the server to identify the protection space of the `Request-URI`.

To receive authorization, the client sends the user-ID and password, separated by a single colon (":") character, within a base64 [5] encoded string in the `credentials`.

```
basic-credentials = "Basic" SP basic-cookie
basic-cookie      = <base64 [5] encoding of userid-password,
```

[2] http://www.w3.org/Protocols/HTTP/1.0/draft-ietf-http-spec.html#BasicAA

```
                          except not limited to 76 char/line>
      userid-password   = [ token ] ":" *TEXT
```

If the user agent wishes to send the user-ID "Aladdin" and password "open sesame", it would use the following header field:

```
      Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
```

The basic authentication scheme is a non-secure method of filtering unauthorized access to resources on an HTTP server. It is based on the assumption that the connection between the client and the server can be regarded as a trusted carrier. As this is not generally true on an open network, the basic authentication scheme should be used accordingly. In spite of this, clients may implement the scheme in order to communicate with servers that use it.

7.1.3 Technology Integration Experiments

In the Auth IE existing OGC clients and services were used to accommodate the HTTP Authentication use-case described earlier in this section. Multiple client/service pairs were tested for the HTTP authentication method.

As the scope of this IE is to leverage mechanisms at the HTTP layer, these mechanisms may be applied in a similar fashion to all OGC service specifications. For simplicity reasons, participants demonstrated the authentication capabilities using WMS client/service components. For this project a distributed test environment for HTTP authentication was developed to:

- Reconcile requirements and expertise across organizations.
- Provide a collaborative, distributed, service-oriented test environment.
- Demonstrate a shared, service-oriented runtime test environment where prototype capabilities can be verified and validated against common requirements.
- Execute scenarios and document best practices for HTTP Authentication.

The benefits of a common testing capability were substantial. In particular, the distributed test environment ensured that HTTP Authentication best practices for WMS were implementable under operational conditions.

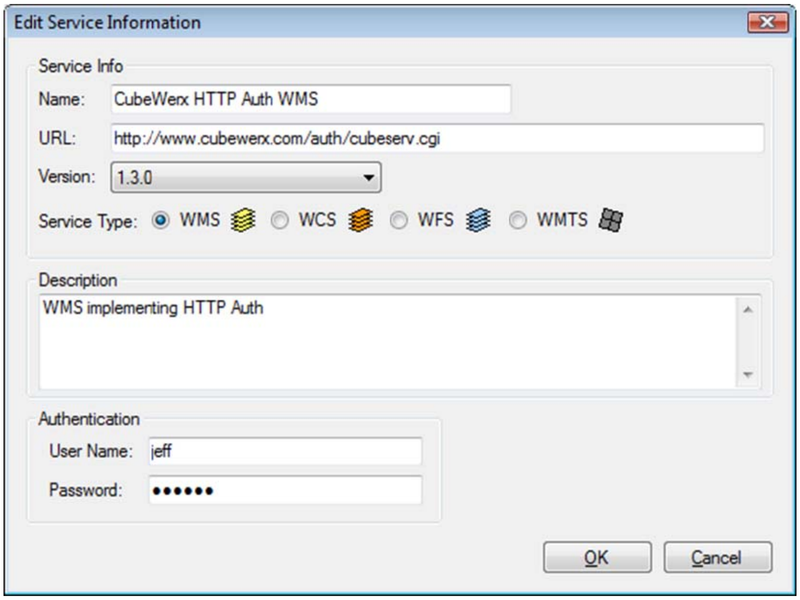
For this project a WMS in multiple versions was provided by CubeWerx, hosting VMAP Level 0 and digital elevation data for the world. The WMS was accessed by WMS clients from The Carbon Project and CubeWerx using the method described in Section 7.1.2. As of February 2010 the WMS is at:

http://www.cubewerx.com/auth/cubeserv.cgi

The WMS is protected by HTTP basic authentication. Users can log in as username: jeff, password: carbon. Users may exercise this WMS by downloading the free Gaia application from The Carbon Project:

http://www.thecarbonproject.com/gaia.php

The basic process is to click "Add new service to the list" in the "Add layer to map" dialog of the Gaia application - and complete the Authentication section. If users do not complete this, they will not be able to access the WMS. The sequence is shown below:



Edit Service Information

Service Info

Name: CubeWerx HTTP Auth WMS

URL: http://www.cubewerx.com/auth/cubeserv.cgi

Version: 1.3.0

Service Type: ● WMS ○ WCS ○ WFS ○ WMTS

Description

WMS implementing HTTP Auth

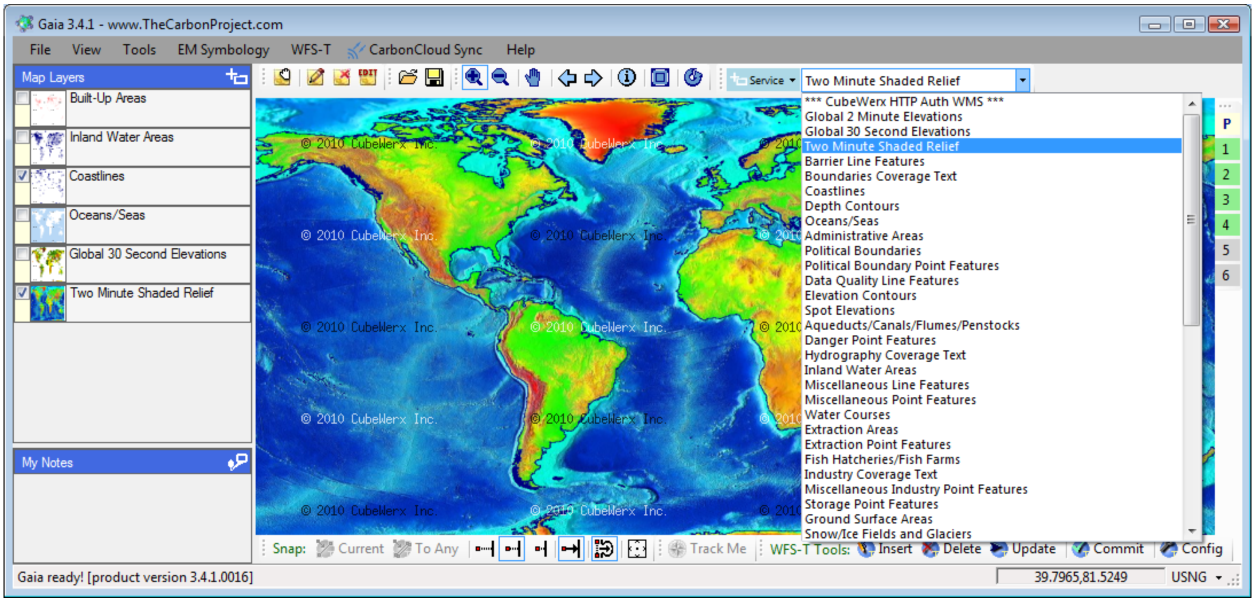
Authentication

User Name: jeff

Password: ••••••

Figure 3 - Credentials for CubeWerx WMS prompted by Gaia

Once credentials are provided to Gaia the secure CubeWerx WMS is accessible:



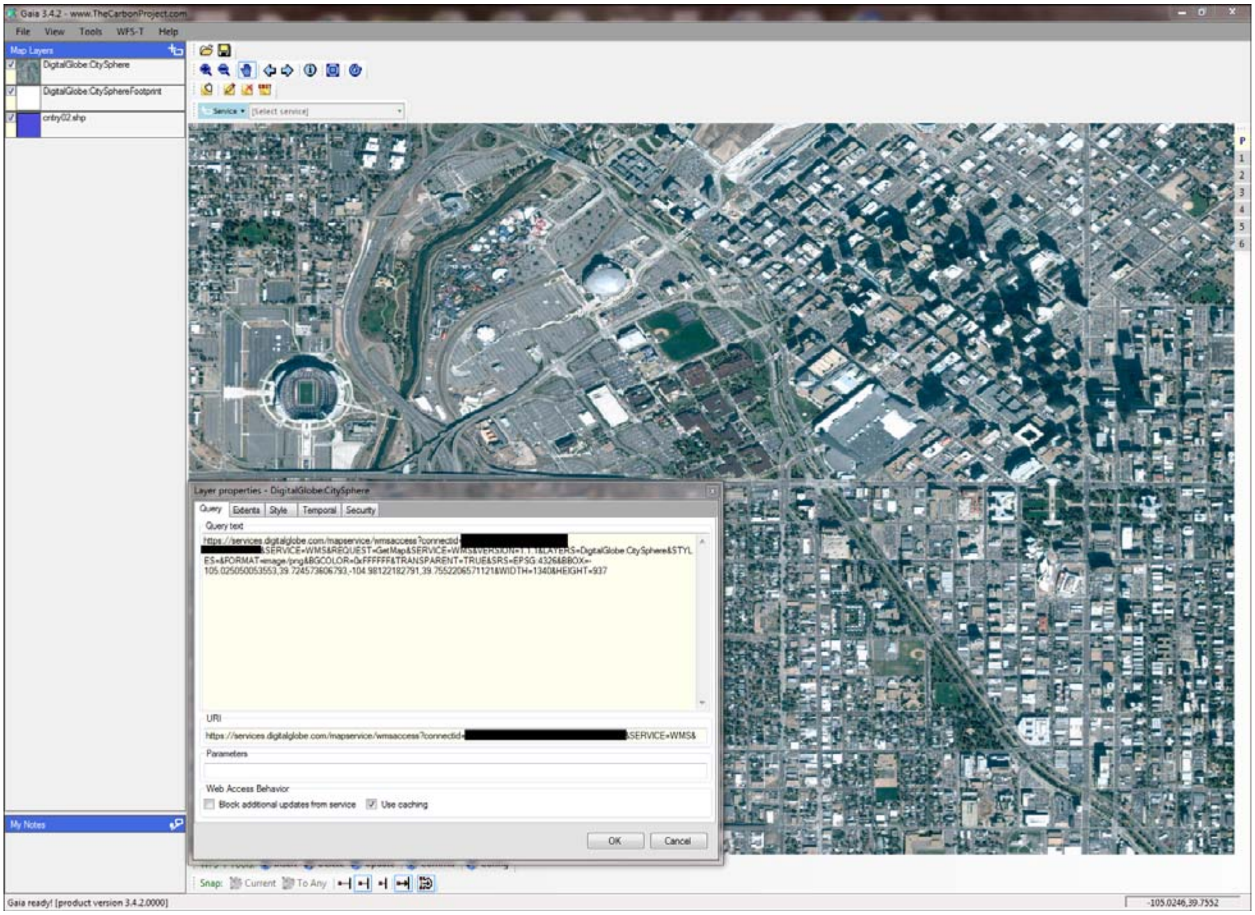
Gaia ready! [product version 3.4.1.0016]

Figure 4 - CubeWerx WMS implementing HTTP Auth in Gaia

Users may also exercise the CubeWerx service protected by HTTP basic authentication in a browser:

http://www.cubewerx.com/auth/cubeserv.cgi?service=wms&request=getcapabilities

In addition, the Gaia client was successfully tested using HTTPS for an OGC WMS against services provided by DigitalGlobe. An example from DigitalGlobe WMS is shown below from June 2010. This Technology Integration Experiment (TIE) was especially important since DigitalGlobe's web-based access services provide imagery via WMS (and other standards) for a number of operational uses including emergency planning, risk assessment, monitoring, emergency response, damage assessment, recovery and others.



Layer properties - DigitalGlobe:CitySphere

Query | Extents | Style | Temporal | Security

Query text

```
https://services.digitalglobe.com/mapservice/wmsaccess?connectid=[redacted]&SERVICE=WMS&REQUEST=GetMap&SERVICE=WMS&VERSION=1.1.1&LAYERS=DigitalGlobe:CitySphere&STYLES=&FORMAT=image/png&BGCOLOR=0xFFFFFF&TRANSPARENT=TRUE&SRS=EPSG:4326&BBOX=-105.025050053553,39.724573606793,-104.98122182791,39.7552206571121&WIDTH=1340&HEIGHT=937
```

URI

```
https://services.digitalglobe.com/mapservice/wmsaccess?connectid=[redacted]&SERVICE=WMS&
```

Parameters

Web Access Behavior

□ Block additional updates from service ☑ Use caching

OK | Cancel

Figure 5 - DigitalGlobe WMS implementing HTTPS in Gaia. For privacy purposes the access credentials passed to the WMS are "blacked out".

To support this functionality The Carbon Project used the CarbonTools PRO[3] capability to support password-based authentication schemes such as basic, digest, NTLM, and Kerberos authentication in the Gaia client. These authentication schemes are supported via the HTTP protocol and are negotiated with the Web Server. To ensure correct access to the relevant services CarbonTools PRO includes the username and password credentials in all queries to the Web Service.

Another HTTP Authentication example was provided by lat/lon as an instance of a deegree Web Map Service 1.3.0. The capabilities are available here:

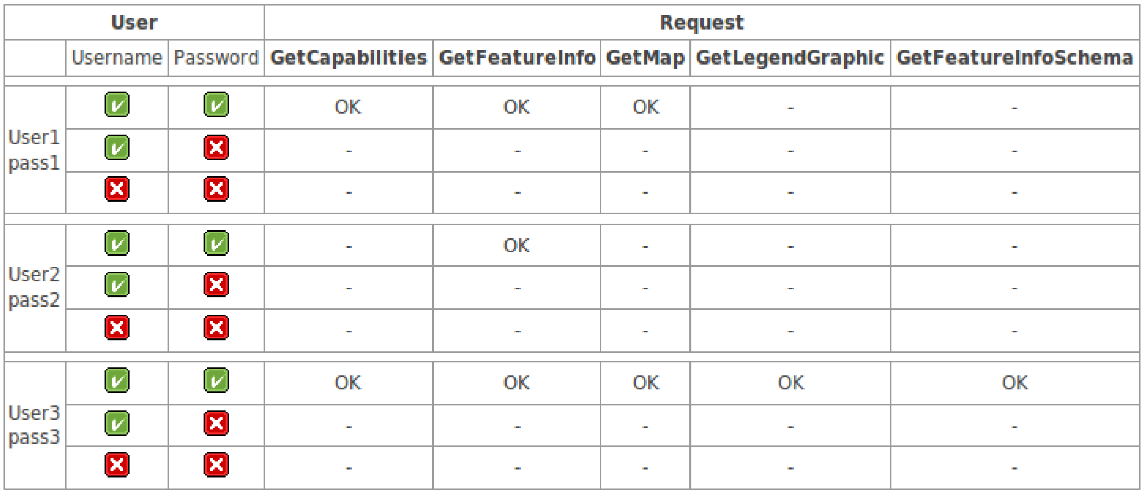
`http://authie.lat-lon.de/wmss/services?service=WMS&request=GetCapabilities`

Since this WMS requires authentication, a request against this URL will result in a "HTTP 401 Unauthorized" response. Table 1 illustrates the different combinations of

[3] www.carbontools.com

users and passwords to access the functionality of the WMS. The response code *HTTP 200 OK* is marked as OK and *HTTP 403 FORBIDDEN* is marked as minus (-). If authentication doesn't succeed, a *HTTP 403 FORBIDDEN* response will be generated. In summary, the different users have the following options:

- User1 (password=pass1) has the ability to request GetCapabilities, GetFeatureInfo and GetMap.
- User2 (password=pass2) is allowed to request just one operation (GetFeatureInfo) on the WMS. All other requests will be refused.
- User3 (password=pass3) is able to request all operations implemented in this WMS.



User			Request				
	Username	Password	**GetCapabilities**	**GetFeatureInfo**	**GetMap**	**GetLegendGraphic**	**GetFeatureInfoSchema**
User1 pass1	✓	✓	OK	OK	OK	-	-
	✓	✗	-	-	-	-	-
	✗	✗	-	-	-	-	-
User2 pass2	✓	✓	-	OK	-	-	-
	✓	✗	-	-	-	-	-
	✗	✗	-	-	-	-	-
User3 pass3	✓	✓	OK	OK	OK	OK	OK
	✓	✗	-	-	-	-	-
	✗	✗	-	-	-	-	-

Figure 6 - Username / Password combinations to access different methods of deegree WMS

The simplest way to access the capabilities of this is using GNU wget (www.gnu.org/software/wget) from a command line or shell.

Example 1 – Valid credentials:

```
wget --http-user=User1 --http-passwd=pass1 'http://authie.lat-
lon.de/wmss/services?service=WMS&request=GetCapabilities'
```

results in a response containing the WMS capabilities. A valid combination of username and password has been submitted to the WMS.

Example 2 – Invalid credentials:

```
wget --http-user=InvalidUser --http-passwd=invalidPasswd
'http://authie.lat-
lon.de/wmss/services?service=WMS&request=GetCapabilities'
```

results in a HTTP 401 Unauthorized response, thus denying access to the WMS. Username as well as password were invalid, so no access will be granted to the requested operation.

Example 3 – No credentials:

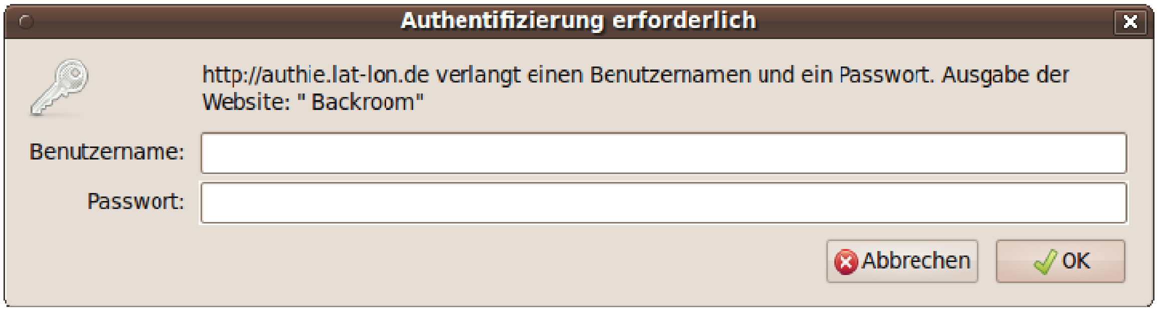
```
wget 'http://authie.lat-lon.de/wmss/services?service=WMS&request=GetCapabilities'
```

results in a HTTP 401 Unauthorized response, thus denying access to the WMS. Neither username nor password have been submitted

Example 4 – Not authorized for specific operation:

wget --http-user=User2 --http-passwd=pass2 'http://authie.lat-lon.de/wmss/services?service=WMS&request=GetCapabilities' results in a HTTP 403 FORBIDDEN response, thus denying access to the GetCapabilities-operation of the WMS. In contrast, the same credentials for a different operation (GetFeatureInfo) would lead to a HTTP 200 OK response. This example demonstrates the capabilities of the service to configure a fine-grained access-restriction (e.g. based on each operation).

A simple Open Layers (www.openlayers.org) client was also made available to display the map layers served by the WMS. By accessing http://authie.lat-lon.de/wmss/ the browser will pop up a window prompting for credentials. Depending on the locale of the operating system, the prompt will appear in a different language (shown below in German).



Authentifizierung erforderlich

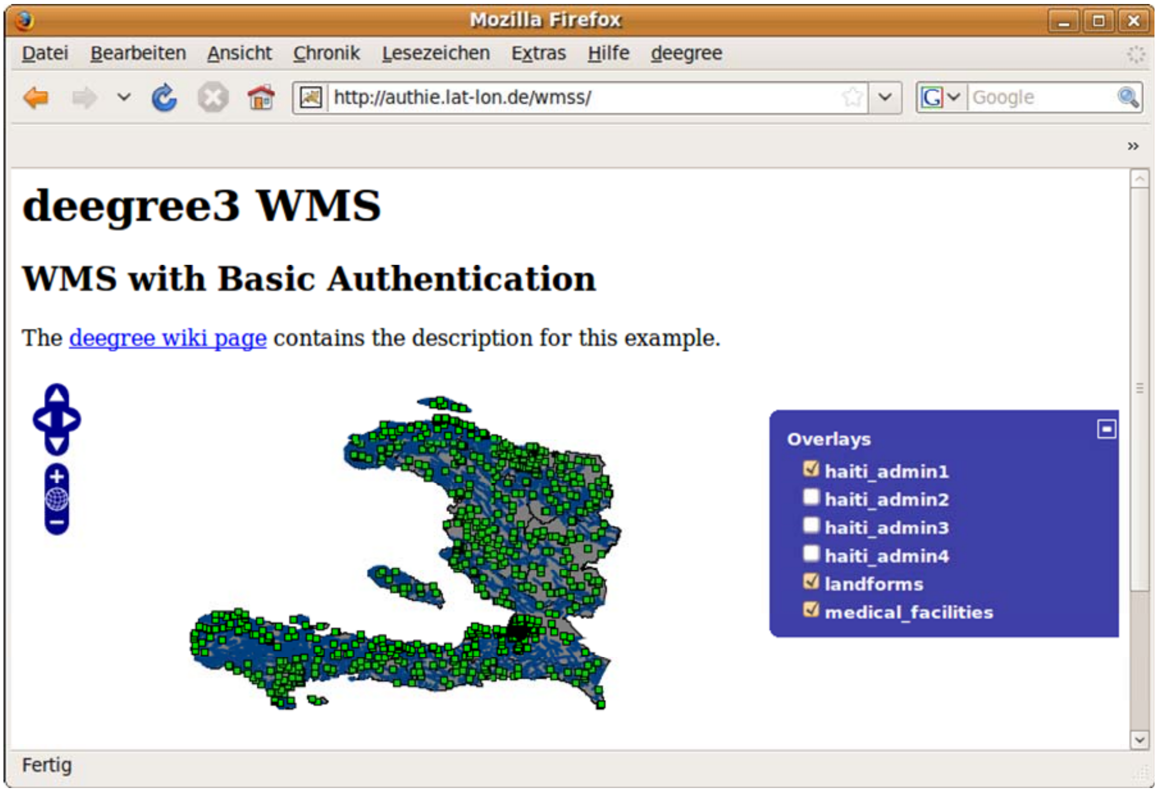
http://authie.lat-lon.de verlangt einen Benutzernamen und ein Passwort. Ausgabe der Website: " Backroom"

Benutzername:

Passwort:

Figure 7: Credentials prompted by Mozilla Firefox (German locale)

Entering valid credentials according to Figure 6, results in a map similar to the one shown below.


Mozilla Firefox
Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe deegree
http://authie.lat-lon.de/wmss/
Google
Fertig

deegree3 WMS

WMS with Basic Authentication

The deegree wiki page contains the description for this example.

Overlays
☑ haiti_admin1
□ haiti_admin2
□ haiti_admin3
□ haiti_admin4
☑ landforms
☑ medical_facilities

Figure 8 - OpenLayers client on top of secured deegree WMS (after successful authentication)

An integration experiment is optimal if different organizations provide client and service. For this use case we have The Carbon Project and lat/lon providing the clients and CubeWerx, lat/lon and DigitalGlobe providing the services. In addition, web browsers may be used to test CubeWerx, lat/lon WMS. Since vendor participants operated on a volunteer basis during this process, the experiment represented a solid effort in this important technology area.

Service Provider / Service	Gaia Client (HTTP Auth)	Gaia Client (HTTPS)	OpenLayers Client	Web Browser
CubeWerx WMS	√			√
lat/lon deegree WMS	√		√	√
DigitalGlobe WMS		√		

TIE Matrix

7.1.4 Summary and Future Work

The basic authentication scheme tested in this use case highlighted a non-secure method of filtering unauthorized access to resources on an HTTP server. The practice is based on the assumption that the connection between the client and the server can be regarded as a trusted carrier. As this is not generally true on an open network, the basic authentication scheme should be used accordingly. In addition, this use case extended the original planned scope and assessed HTTPS using commercially available WMS to provide secure communications. This method should be investigated further by the open geospatial community as resources are available since it supports the assumption that the connection between the client and the server can be regarded as a trusted carrier.

7.2 WS-Security Use Case

For WS-Security different scenarios can be imagined as there are several WS-Security tokens defined by OASIS (user name, X.509, SAML, Kerberos, ISO REL). Each of these tokens corresponds to one authentication method with its own interaction model. Furthermore, there are different ways in which authentication can be performed with each token type: if authentication is performed by means of digital signatures, one can sign the whole SOAP message, the SOAP body, or only some part of the SOAP body.

For this reasons, within this Interoperability Experiment only the simplest use-case was taken into consideration: authentication by means of user-name password where the service implemented SOAP.

7.2.1 Assumptions and Interactions

Assumptions:

1. The client knows the URL of the service
2. The client is capable of formulating correct SOAP requests with WS-Security headers
3. The client is already known by the service
    a. User name and password has been distributed by some out-of-band mechanism

Interactions:

1. The client formulates a correct SOAP request. The request has WS-Security headers that contain authentication information
2. Service receives the request with authentication information, verifies the authentication information and serves the OGC request as expected

7.2.2 Method for Authentication on Catalog Service

In contrast to HTTP Basic Authentication, the WS-Security use case requires the username / password to be transmitted through SOAP messages. The capabilities of the service are available through the following Catalog:

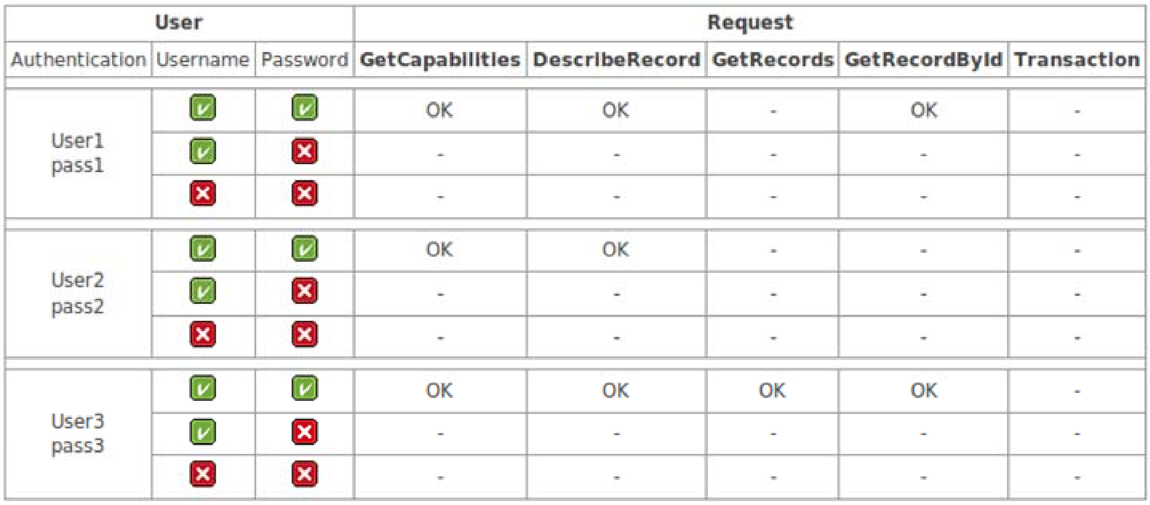
http://authie.lat-lon.de/csws/services?service=CSW&request=GetCapabilities

Within a SOAP request the credentials will be integrated into a SOAP header (see Listing 1 below).

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

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd">

   <soapenv:Header>

        <wsse:Security

            soapenv:mustUnderstand="1">

            <wsse:UsernameToken>

                <wsse:Username>User3</wsse:Username>

                <wsse:Password Type="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">pass3</wsse:Password>

            </wsse:UsernameToken>

        </wsse:Security>

    </soapenv:Header>

[…]
```

Listing 1: SOAP header containing username / password

A Catalogue Service-Web containing metadata was implemented during the Auth IE. Authentication has to be handled within the SOAP communication layer, i.e. by utilizing the SOAP header elements for transfering username and password. The Figure below illustrates three username / password combinations and lists the available operations. Only users providing valid credentials may access the operaions provided by the CS-W.



User			Request				
Authentication	Username	Password	GetCapabilities	DescribeRecord	GetRecords	GetRecordById	Transaction
User1 pass1	☑	☑	OK	OK	-	OK	-
	☑	☒	-	-	-	-	-
	☒	☒	-	-	-	-	-
User2 pass2	☑	☑	OK	OK	-	-	-
	☑	☒	-	-	-	-	-
	☒	☒	-	-	-	-	-
User3 pass3	☑	☑	OK	OK	OK	OK	-
	☑	☒	-	-	-	-	-
	☒	☒	-	-	-	-	-

Figure 9 – Username/Password combinations for accessing CS-W operations

To access this service, you can either use a generic client (http://authie.lat-lon.de/csws/) to access the service or any other client capable of posting XML messages to a URL. The service endpoint is http://authie.lat-lon.de/csws/services.

To excercise the server, you can use one of the examples:

1) http://authie.lat-lon.de/csws/soap_xml/1id_fullSOAP.xml
2) http://authie.lat-lon.de/csws/soap_xml/getAllRecords_DCSOAP.xml
3) http://authie.lat-lon.de/csws/soap_xml/Not_AndSOAP.xml
4) http://authie.lat-lon.de/csws/soap_xml/PropEqualSOAP.xml
5) http://authie.lat-lon.de/csws/soap_xml/describeRecord_3SOAP.xml
6) http://authie.lat-lon.de/csws/soap_xml/getCapabilitiesSOAP.xml
7) http://authie.lat-lon.de/csws/soap_xml/PropEqual_MD_MetadataSOAP.xml

In case a user provides invalid credentials, the service issues a SOAP-Fault (see Listing 2 below).

```
<?xml version="1.0" ?>

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.w3.org/2003/05/soap-envelope
http://www.w3.org/2003/05/soap-envelope">

  <soapenv:Body>

    <soapenv:Fault>
```

```
      <soapenv:faultcode>


<soapenv:faultcode>wsse:FailedAuthentication</soapenv:faultcode>

      </soapenv:faultcode>

      <soapenv:faultstring>

        <soapenv:Text xml:lang="en">The security token could not be
authenticated or authorized</soapenv:Text>



      </soapenv:faultstring>

    </soapenv:Fault>

  </soapenv:Body>

</soapenv:Envelope>
```

Listing 2: SOAP Fault declaring a failed authentication

The Auth IE identified the following Shortcomings / Open Issues in this Use Case:

- No WSDL file available for CS-W
- The capabilities do not provide any information on how to access the service through SOAP
- WS-Security support is restricted to transfering username / password within SOAP header as well as receiving information on invalid credentials within SOAP fault messages

7.3 SAML based Authentication Use Case

This section is provided by the EDINA National Data Centre based at the University of Edinburgh. In association with the European Spatial Data Infrastructure Network (ESDIN) project and the European Persistent Geospatial Testbed for Research and Education (PTB), EDINA has established a federation of trusted partners for the purpose of sharing protected OGC Web Services. The protection of the services is done according to a simple access control use case: “Authenticated users have full access to the protected services; other users do not have access”.

The realization of the authentication is based on Security Assertion Markup Language (SAML) version 2, a standard by OASIS. In particular, the Web Browser Single-Sign-On and Enhanced Client Profile are supported. It is important to note that in a federation user management and hosting of services is a separate issue. Therefore, the participants created a federation with separation of concerns, having two different kinds of participants:

- **Service Provider** (in SAML called a relying party) is making available protected services such as OGC Web Services to users of other parties of the federation.
- **Identity Provider** (in SAML called an asserting party) is hosting the user accounts for the federation.

This implies that the user does not authenticate with the Service Provider, as is the case with other direct authentication methods such as HTTP Authentication. One of the advantages of SAML is the support for Single-Sign-On (SSO), even though user accounts are distributed among different Identity Providers within the federation. And because of SSO support, a user can login in once and "consume" all protected services within the federation for which they are authorized. For demonstration purposes, the federation currently has two Identity Providers: "*EDINA Test IdP*" and "*GeoRM Test IdP*".

The following references are used in this section:

[1] **SAML**: Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard, 15 March 2005: http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

[2] **SAML-Bindings**: Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard, 15 March 2005: http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf

[3] **SAML-Profiles**: Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard, 15 March 2005: http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf

At the moment, these Service Providers host the following OGC Web Map Services (WMS):

Protected Service	Service Provider	Service GetCapabilities URL
Demis World Map	GeoRM	https://www.georm.org/service/WorldMap?Request=capabilities&SERVICE=WMS&VERSION=1.1.0
Ordnance Survey Raster 250k	EDINA	https://esdin.edina.ac.uk:7111/cgi-mapserv/mapserv?map=mapfiles/raster250k.map&version=1.1.1&service=WMS&request=GetCapabilities

For the proof of interoperability an OpenLayers based demo client was made available to view the protected WMSs of the federation:

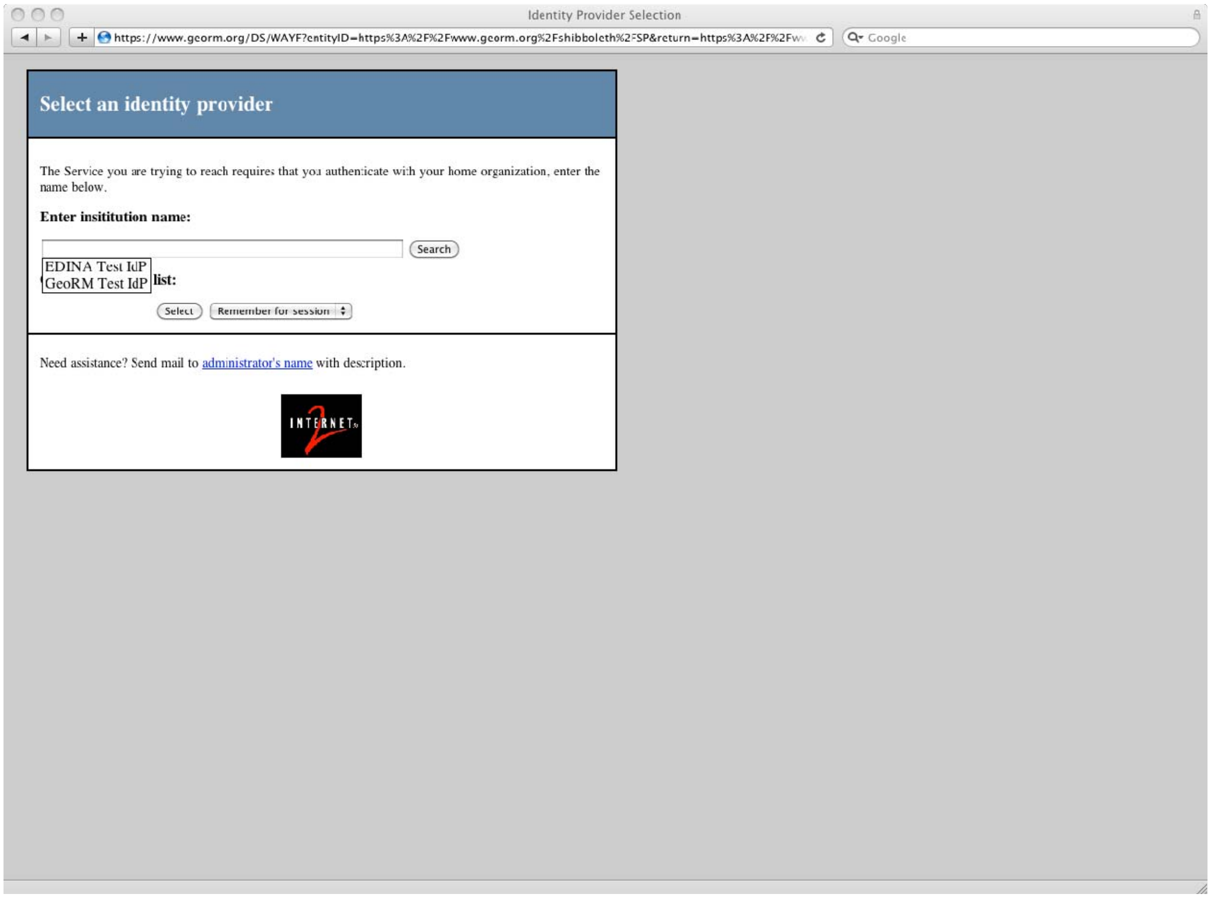
https://www.georm.org/secure/wms.html

In order to run the demo client, the user must login to the federation. This is possible by selecting the Identity Provider "*EDINA Test IdP*" with user name *ogcuser* and password *ogcuser*.

7.3.1 Technology Integration Experiments

The following screen shots illustrate step by step how to begin exercising this use case:

1) Type the OpenLayers Client URL https://www.georm.org/secure/wms.html in your Web Browser. You are going to see the HTML Page (as illustrated below) to select your home institution which is "EDINA Test IdP".



Identity Provider Selection

https://www.georm.org/DS/WAYF?entityID=https%3A%2F%2Fwww.georm.org%2Fshibboleth%2FSP&return=https%3A%2F%2Fw

Select an identity provider

The Service you are trying to reach requires that you authenticate with your home organization, enter the name below.

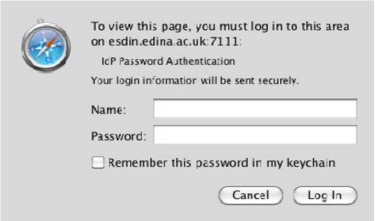
Enter insititution name:

EDINA Test IdP
GeoRM Test IdP **list:**

Need assistance? Send mail to administrator's name with description.

Figure 10 - WAYF page to select the home institution

2) After clicking the “select” button, the Web Browser prompts you to input your username and password, as illustrated on the figure below. Note - Please use ***ogcuser*** as username and password.



To view this page, you must log in to this area on esdin.edina.ac.uk:7111:

IcP Password Authentication

Your login information will be sent securely.

Name:

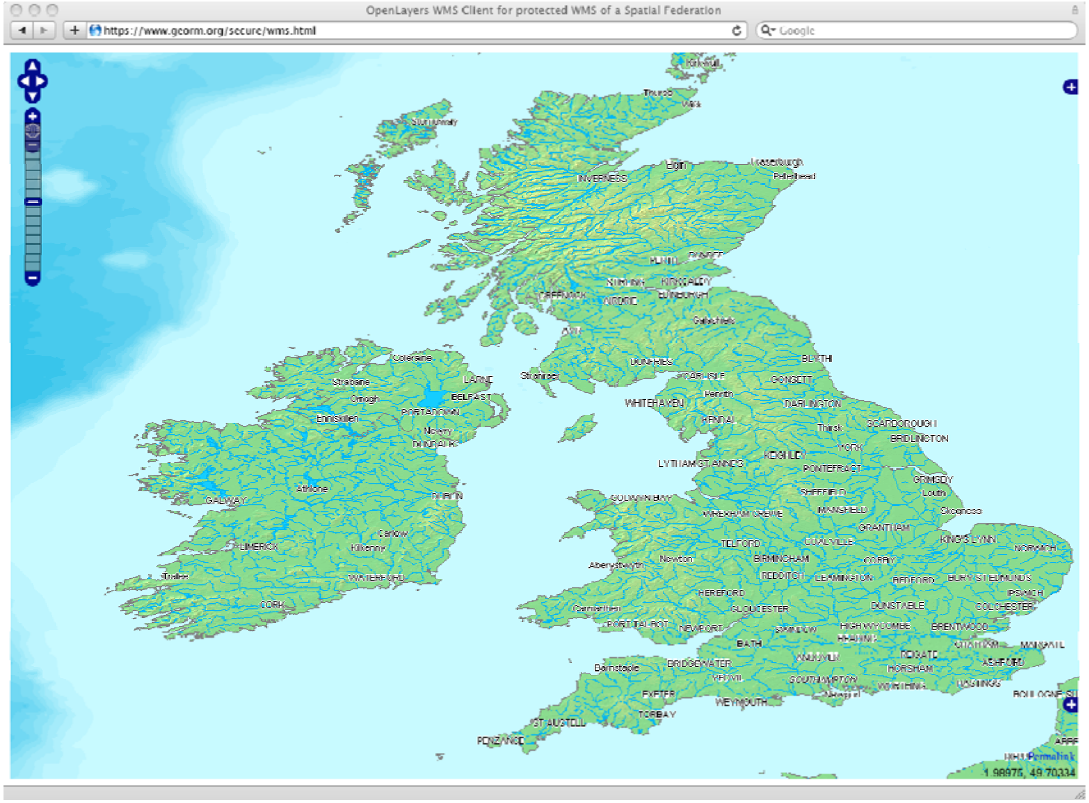
Password:

□ Remember this password in my keychain

Cancel Log In

Figure 11 - HTTPS Authentication Login

3) After clicking the “log in” button, the OpenLayers client is loaded and you are going to see the Demis World Map as illustrated below.



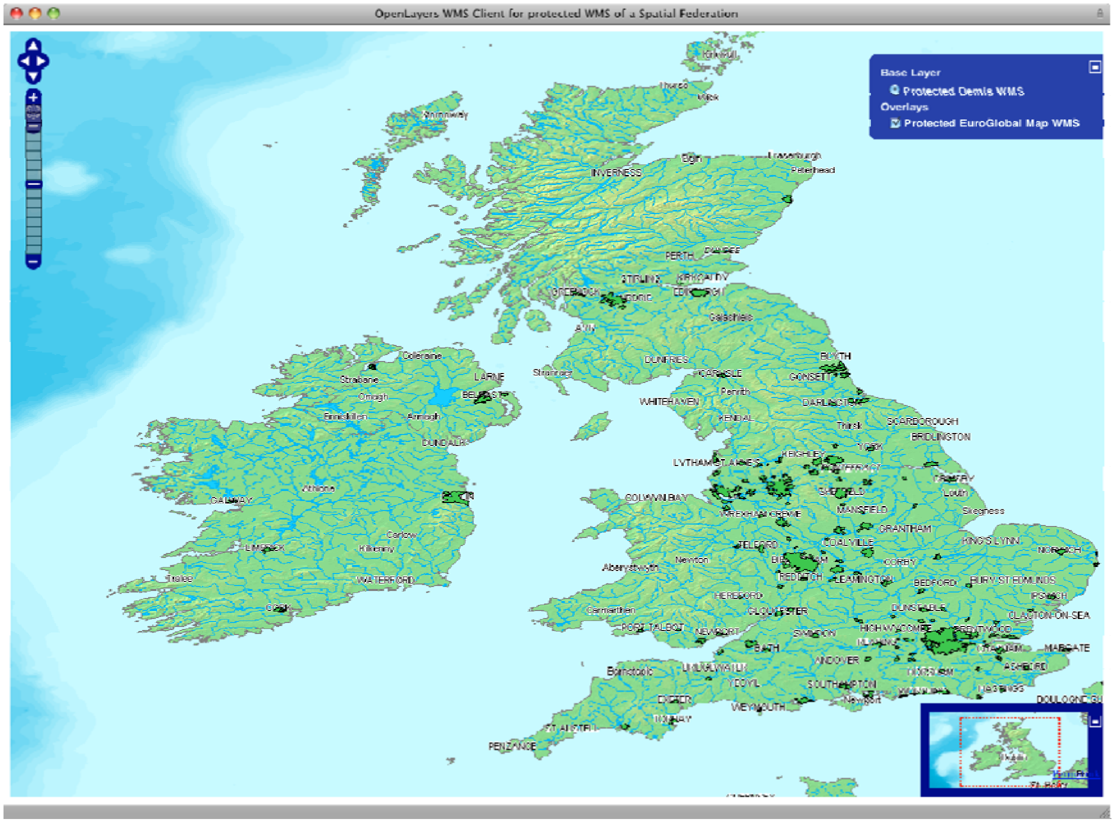
OpenLayers WMS Client for protected WMS of a Spatial Federation

https://www.gcorm.org/secure/wms.html

Figure 12 - OpenLayers client showing the protected Service “Demis World Map”

You can open the overview map by clicking on the "+" symbol at the right bottom of the page. You can select the other base layer "Raster 250k" by clicking on the "+" symbol at the right top of the page.

The following figure illustrates the overlay ESDIN licensed data on top of the Demis World Map. Unfortunately, access to that WMS cannot be made available for a live demo.



Base Layer

● Protected Demis WMS

Overlays

☑ Protected EuroGlobal Map WMS

Figure 13 - OpenLayers client showing EuroGlobalMap data on top of Demis World Map

7.3.2 SAML Background Information

The Security Assertion Markup Language (SAML) is an OASIS standard (see 7.3), concerned with the standardization of assertions and their exchange between an asserting and a relying party. It is therefore typically used in distributed environments where the user account is not located at the same entity that is asking for authentication. In that sense, SAML defines different profiles that define the interactions between a relying party (the entity receiving assertions about a user) and the asserting party (the entity that is creating statements about the user). Different assertions about the user are defined in SAML. The most important ones are:

- **Authentication Assertion** which provides information about the method of authentication used to verify the claimed identity of the user;
- **Attribute Assertion** which provides (detailed) information about the user concerning her characteristics;
- **Authorization Assertion** provides information about the permissions a user has on a protected resource.

Particularly important for our contribution to the Authentication IE is the Web Browser SSO (Single-Sign-On) and the Enhanced Client Profile (ECP) as defined in the SAML 2 Profile standard (see 7.3). In order to use these profiles, one of the bindings as defined in SAML 2 Bindings standard (see 7.3) must be used. For the ECP only one binding is available: PAOS which is an acronym for Reverse SOAP Binding as defined by the Liberty Project. The Web Browser SSO Profile can be used with the HTTP POST or HTTP Artifact Binding.

7.3.2.1 Web Broswer SSO Profile

The following interaction diagram (see 7.3) defines the normative interactions between a user agent, a Service Provider and an Identity Provider for the purpose of enabling access to a protected resource. As a pre-condition it is assumed that the user agent is requesting a protected resource from a SP but no session has previously being established.

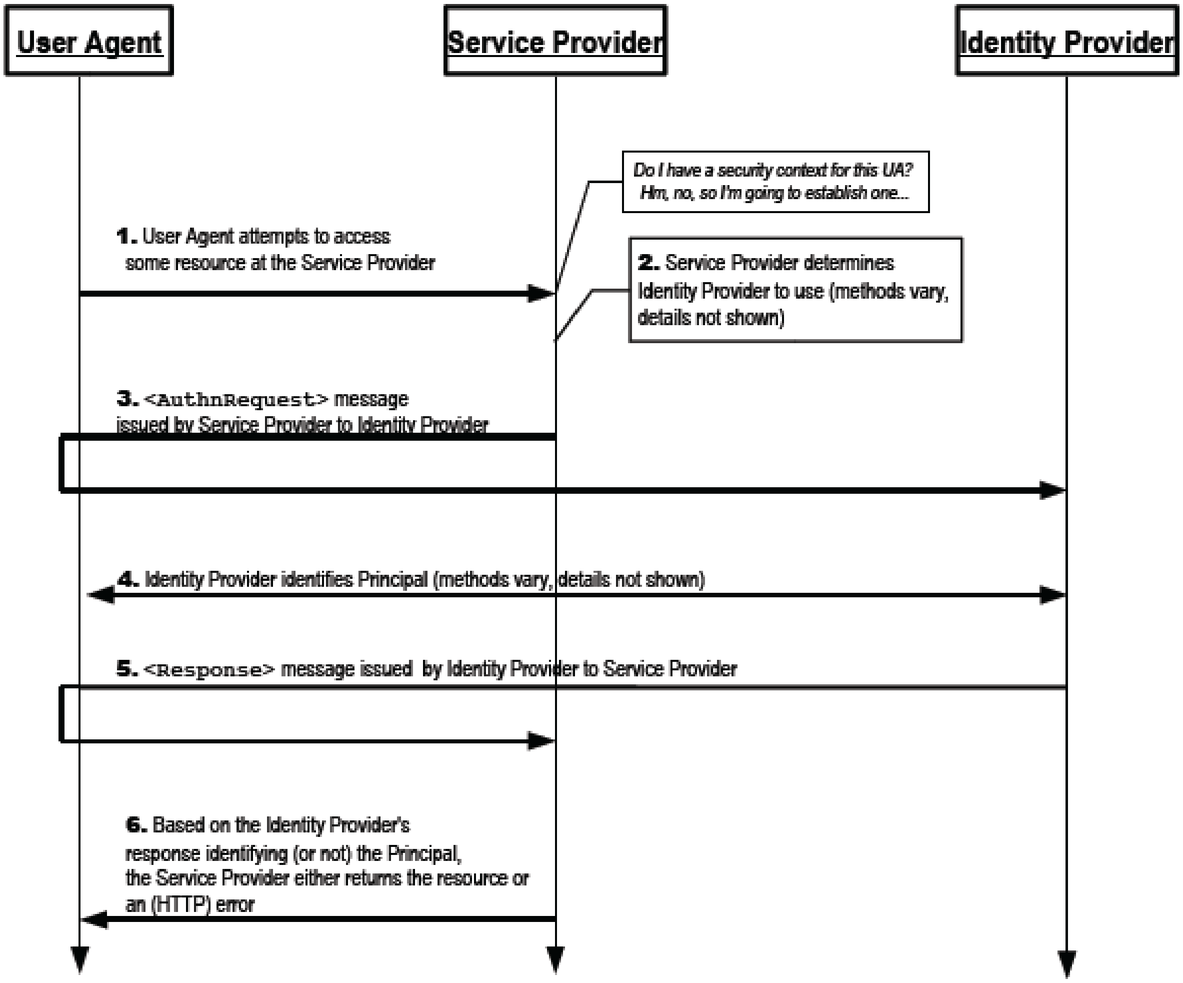

User Agent
Service Provider
Identity Provider
Do I have a security context for this UA? Hm, no, so I'm going to establish one...
1. User Agent attempts to access some resource at the Service Provider
2. Service Provider determines Identity Provider to use (methods vary, details not shown)
3. <AuthnRequest> message issued by Service Provider to Identity Provider
4. Identity Provider identifies Principal (methods vary, details not shown)
5. <Response> message issued by Identity Provider to Service Provider
6. Based on the Identity Provider's response identifying (or not) the Principal, the Service Provider either returns the resource or an (HTTP) error

Figure 14 - SAML 2 Web Browser SSO Profile Sequence Diagram

As illustrated, the User Agent (the user with the client) approaches the Service Provider (SP) requesting access to a protected resource; an OGC Web Service for example. After getting redirected to the Identity Provider (IdP), logging in with the username and password and redirected back to the protected resource, the User Agent potentially gets access. It is important to note that it is the responsibility of the SP (see interaction #2 in the figure above) to redirect the user agent to the appropriate IdP. This can be challenging in a federation with more than one IdP. The general problem to solve is called the Identity Provider Discovery problem. SAML 2 defines a profile with a similar name that states that a common domain cookie shall be used to store a list of IdPs already visited by the user. But, this does not solve the boot-strapping; how to figure out the first IdP to contact for an unknown user. The solution proposed by the Internet2 in their open source implementation of SAML called Shibboleth, is a Discovery Service (DS) called a WAYF (Where Are You From) service. This service intercepts the redirect from the SP to the IdP (see interaction #2 from above figure) and provides a list of available IdPs to the user. Therefore, one solution to the IdP Discovery problem is provided by a Discovery Service, which allows the user selecting the correct IdP.

7.3.2.2 Enhanced Client Profile

The following interaction diagram (see 7.3) defines the normative interactions between an enhanced client, a Service Provider and an Identity Provider for the purpose of enabling access to a protected resource. As a pre-condition it is assumed that the enhanced client is requesting a protected resource from a SP but no session has previously being established.

As illustrated below, the Enhanced Client (the user with the client) approaches the SP requesting access to a protected resource. The SP answers the request by returning a SAML AuthnRequest (SOAP) message. It is now the responsibility of the client to function as a relay between the appropriate IdP and the SP as defined in this profile. It is important to note that firstly, the client only relays SOAP messages and that no processing of SOAP messages, including applying and verifying XML digital signatures as well as encrypting SOAP messages, is required. An important difference to the Web Browser SSO Profile is that with the ECP it is the responsibility of the client to determine the appropriate IdP (see #3 in the figure below). Therefore, the client must provide a solution to the IdP discovery problem.

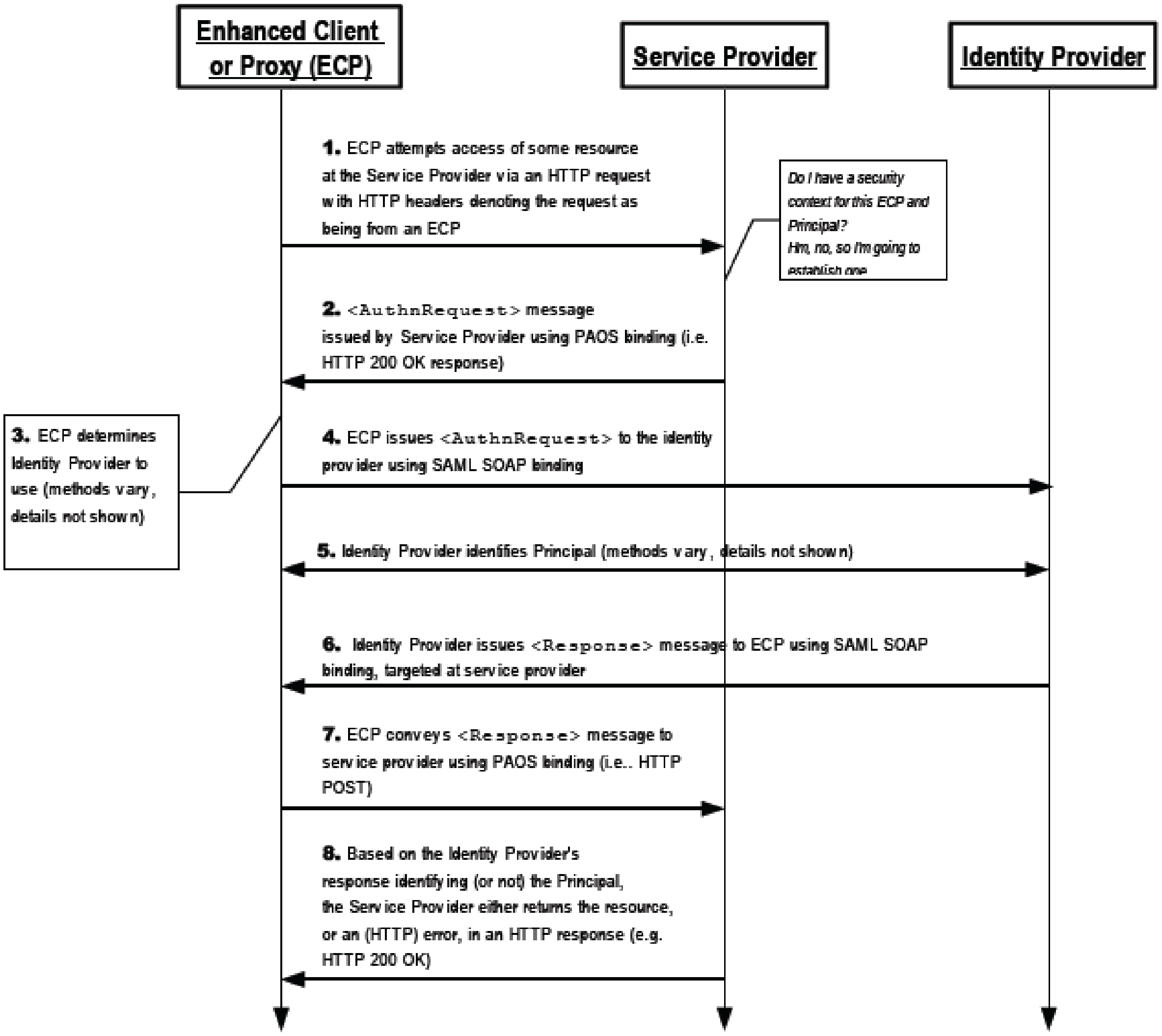

Enhanced Client or Proxy (ECP)
Service Provider
Identity Provider
1. ECP attempts access of some resource at the Service Provider via an HTTP request with HTTP headers denoting the request as being from an ECP
Do I have a security context for this ECP and Principal?
Hm, no, so I'm going to establish one
2. <AuthnRequest> message issued by Service Provider using PAOS binding (i.e. HTTP 200 OK response)
3. ECP determines Identity Provider to use (methods vary, details not shown)
4. ECP issues <AuthnRequest> to the identity provider using SAML SOAP binding
5. Identity Provider identifies Principal (methods vary, details not shown)
6. Identity Provider issues <Response> message to ECP using SAML SOAP binding, targeted at service provider
7. ECP conveys <Response> message to service provider using PAOS binding (i.e.. HTTP POST)
8. Based on the Identity Provider's response identifying (or not) the Principal, the Service Provider either returns the resource, or an (HTTP) error, in an HTTP response (e.g. HTTP 200 OK)

Figure 15 - SAML 2 Enhanced Client Profile Sequence Diagram

7.3.3 Client Development

As OGC Web Services specifications are silent about security, we are not aware of any currently available client implementations that support the required functionality for using SAML 2 based authentication.

For this Authentication IE, we used an OpenLayers based client for testing the SAML 2 Web Browser SSO and an extension to OpenJump 1.3 for testing the SAML 2 Enhanced Client Profile.

For the SAML 2 Web Browser SSO Profile, the OGC client must support all functions of a regular Web Browser. This requires a full implementation of HTTP (e.g. 1.1) including support for HTTPS, HTTP redirects, HTTP Cookies and HTTP Authentication. It is also mandatory that (X)HTML processing and support for JavaScript be implemented. All of these required functions are available for an OWS Client that runs inside the Web Browser. One prominent example is OpenLayers which provides a JavaScript based API to easily connect to different OGC Web Services such as WMS and WFS and create a composite map.

For the SAML 2 Enhanced Client Profile, the OGC client must support HTTPS, HTTP Cookies and HTTP Authentication as well as functionality to relay SOAP messages between the SP and the chosen IdP. But, as an ECP compliant client, it is not necessary that the client implement Web Browser like functions for the processing of (X)HTML pages. Therefore, ECP provides simple requirements a template for Desktop type clients. In order to test the interoperability for the ECP, we have extended the OpenJump open source desktop client.

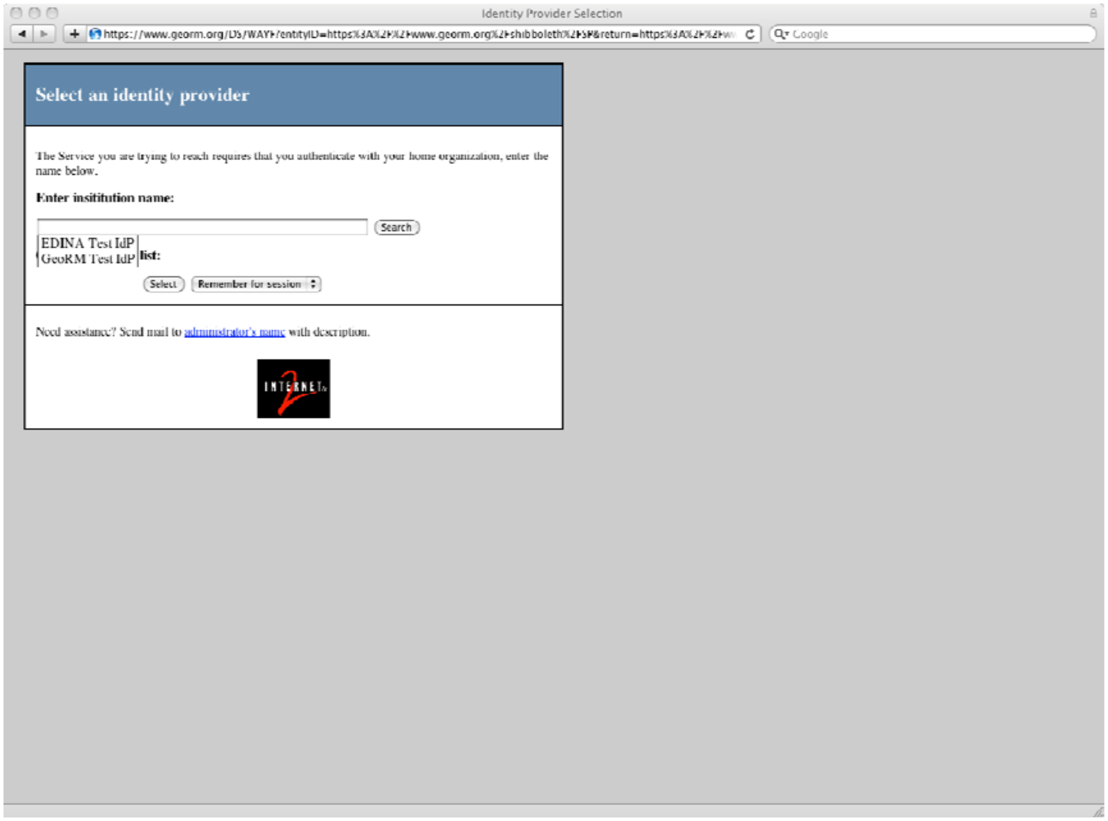
7.3.3.1 OpenLayers based client

We provide an OpenLayers based client based on the stream line version of the OpenLayers API with no modifications to provide access to the protected OGC Web Services of the test federation.

The following screen shots illustrate step by step how to use the OpenLayers client in your favorite[4] web browser to access protected Web Map Services provided by the federation.

1) Use your favorite web browser (e.g. Firefox, Safari, Google Chrome or IE) and open the OpenLayers Client URL https://www.georm.org/secure/wms.html.
2) The browser gets redirected to the Discovery Service and you are going to see a page (as illustrated below) to select your institution. Please use “EDINA Test IdP” and click the “select” button.

[4] Please note that this solution is based on HTTPS and Cookies. We have experienced that the default cookie handling policy for the Safari Browser restricts the acceptance of cookies to visited sites only. Please change the setting to accept cookies from all sites as this is required to ensure Single-Sign-On.



Identity Provider Selection

https://www.georm.org/DS/WAYF?entityID=https%3A%2F%2Fwww.georm.org%2Fshibboleth%2FSP&return=https%3A%2F%2Fw

Select an identity provider

The Service you are trying to reach requires that you authenticate with your home organization, enter the name below.

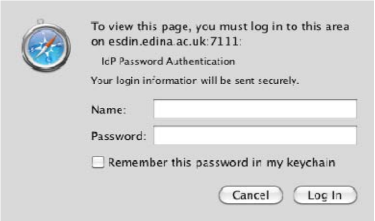
Enter insititution name:

EDINA Test IdP
GeoRM Test IdP **list:**

Need assistance? Send mail to administrator's name with description.

Figure 16 – Discovery Service page to select the home institution

3) After clicking the "select" button, the Web Browser is redirected to the selected IdP. As this endpoint is taken care of the user login, a HTTP Authentication challenge is started. The browser therefore prompts you to input your username and password, as illustrated on the figure below. Please use *ogcuser* as username and password.



To view this page, you must log in to this area on esdin.edina.ac.uk:7111:

IdP Password Authentication

Your login information will be sent securely.

Name:

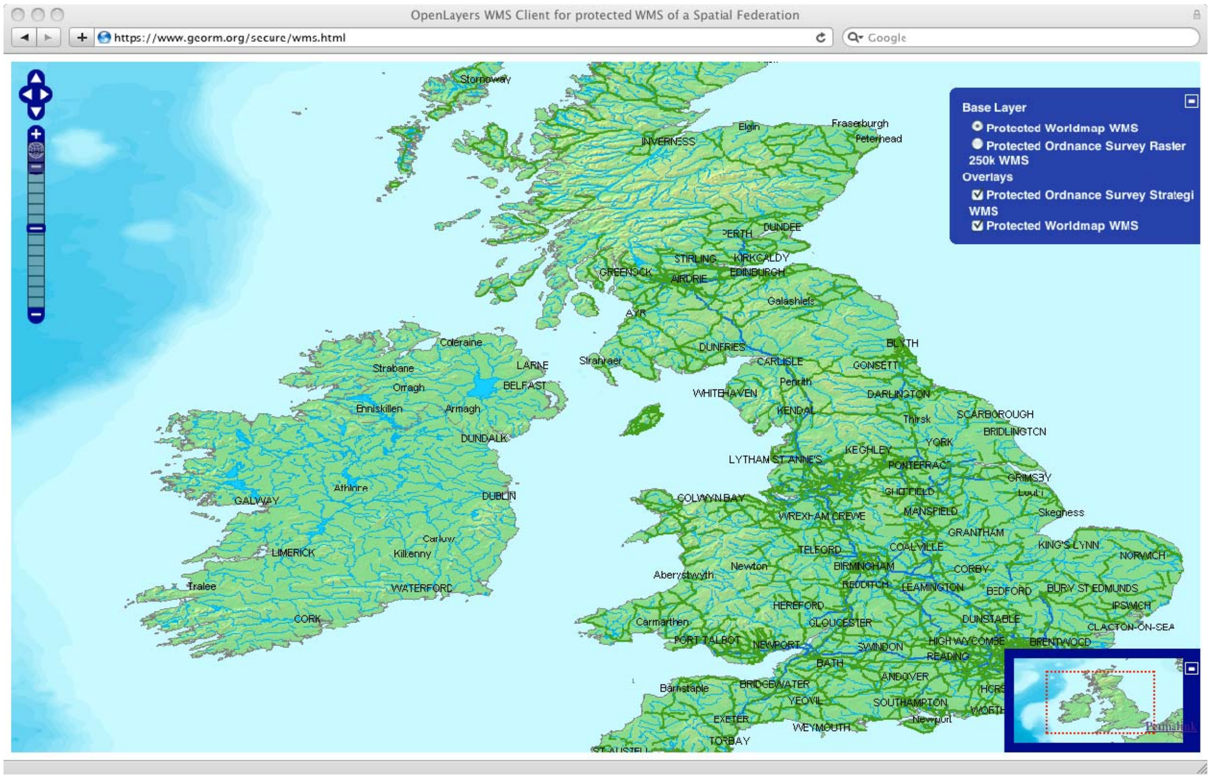
Password:

□ Remember this password in my keychain

Figure 17 - HTTP Authentication Login provided by EDINA

4) After clicking the "log in" button, the Web Browser is redirected back to the original requested URL and the OpenLayers client is loading.

5) After a couple of seconds, you are going to see the a composite map[5] as illustrated below.



Base Layer
○ Protected Worldmap WMS
○ Protected Ordnance Survey Raster 250k WMS
Overlays
☑ Protected Ordnance Survey Strategi WMS
☑ Protected Worldmap WMS

Figure 18 - OpenLayers client showing protected service from the test federation

You can open the overview map by clicking on the "+" symbol at the right bottom of the page. In order to explore additional overlay maps, you can open the layer switch window by clicking on the "+" symbol at the right top of the page. At the time of writing, only an alternative base map layer (OS Raster 250k) is available. But this should change before the end of the Authentication IE.

Please Note: In order to maintain Single-Sign-On it is important to ensure that the web browser accepts cookies from all pages; not only those visited. We noticed that the default configuration of Safari 4.0 is set to "accept cookies from visited pages" which prevents the correct display of the composite map.

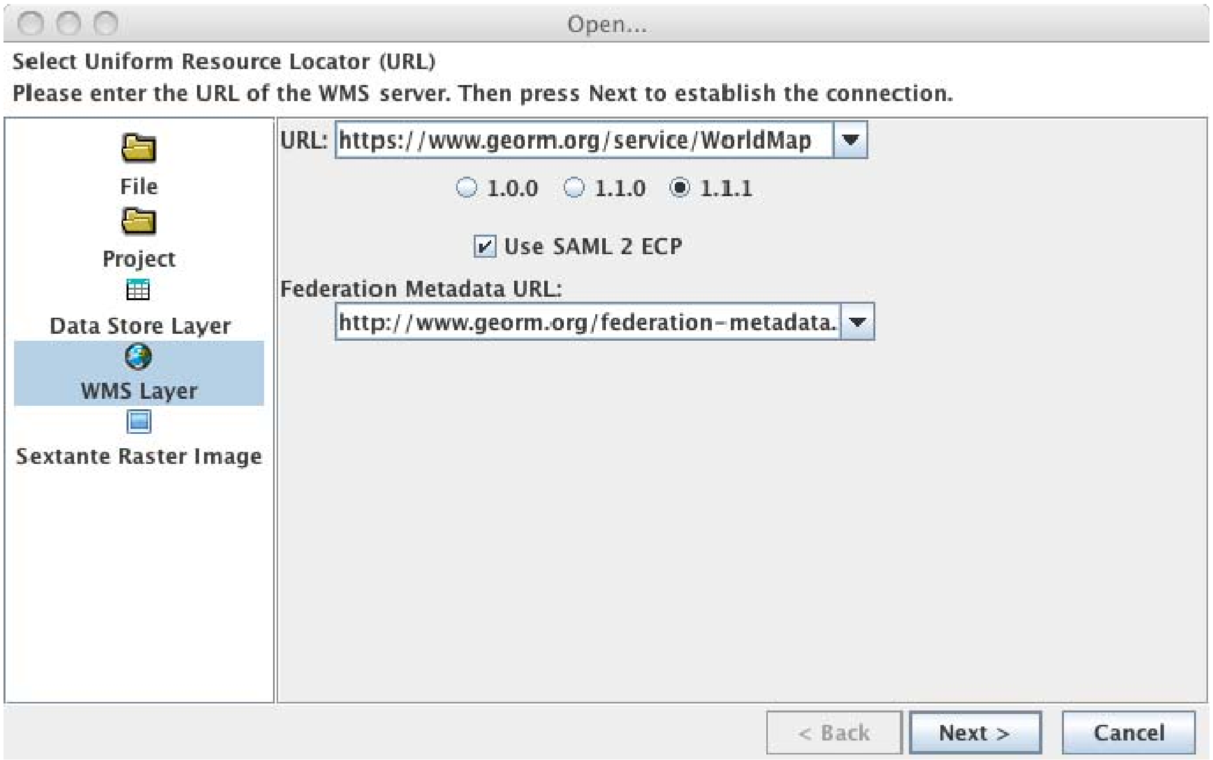
[5] Please note that this screenshot represents all protected services at the time of writing.

7.3.3.2 OpenJump 1.3 based client

OpenJump 1.3 (available from source forge[6]) is an open source desktop client, implemented in Java, which supports the display of maps from OGC Web Map and Web Feature Service. Based on this version, we have created an extension that can be used for SAML 2 ECP protected services.

The following screen shots illustrate the use of the extended OpenJump client to map protected Web Map Services provided by the federation.

1) Start OpenJump Desktop client as explained in the release. Click on “Working”, “Open …” and “WMS Layer”. Please insert the URL provided above for connecting to the protected Demis World Map service. It is also required to select „Use SAML 2 ECP“ and provide the federation metadata URL http://www.georm.org/federation-metadata.xml
.



Select Uniform Resource Locator (URL)

Please enter the URL of the WMS server. Then press Next to establish the connection.

URL: https://www.georm.org/service/WorldMap

○ 1.0.0 ○ 1.1.0 ● 1.1.1

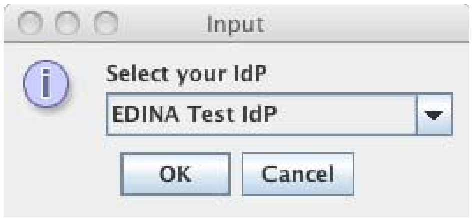
☑ Use SAML 2 ECP

Federation Metadata URL: http://www.georm.org/federation-metadata.

Figure 19 – OpenJump add WMS selection box

2) Select “EDINA Test IdP” from the list as illustrated in the figure below.

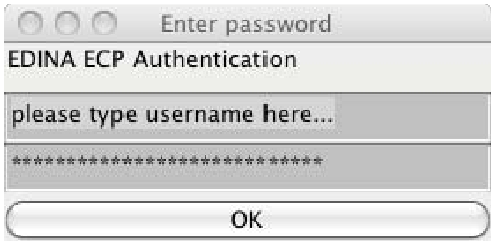
[6] `https://jump-pilot.svn.sourceforge.net/svnroot/jump-pilot jump-pilot`



Select your IdP: EDINA Test IdP

Figure 20 – OpenJump client input box for the IdP selection

3) When prompted for username and password, please insert *ogcuser* as the user name and the password.



EDINA ECP Authentication

please type username here...

****

Figure 21 – OpenJump client username password box

4) After selecting layers, you are going to see the map from the protected Demis World Map.
5) In order to connect to other WMSs of the federation, please repeat 1) but with the appropriate URL. Please make sure you have selected “Use SAML 2 ECP” as the service is protected. But you do not have to login again, as the client supports SSO.

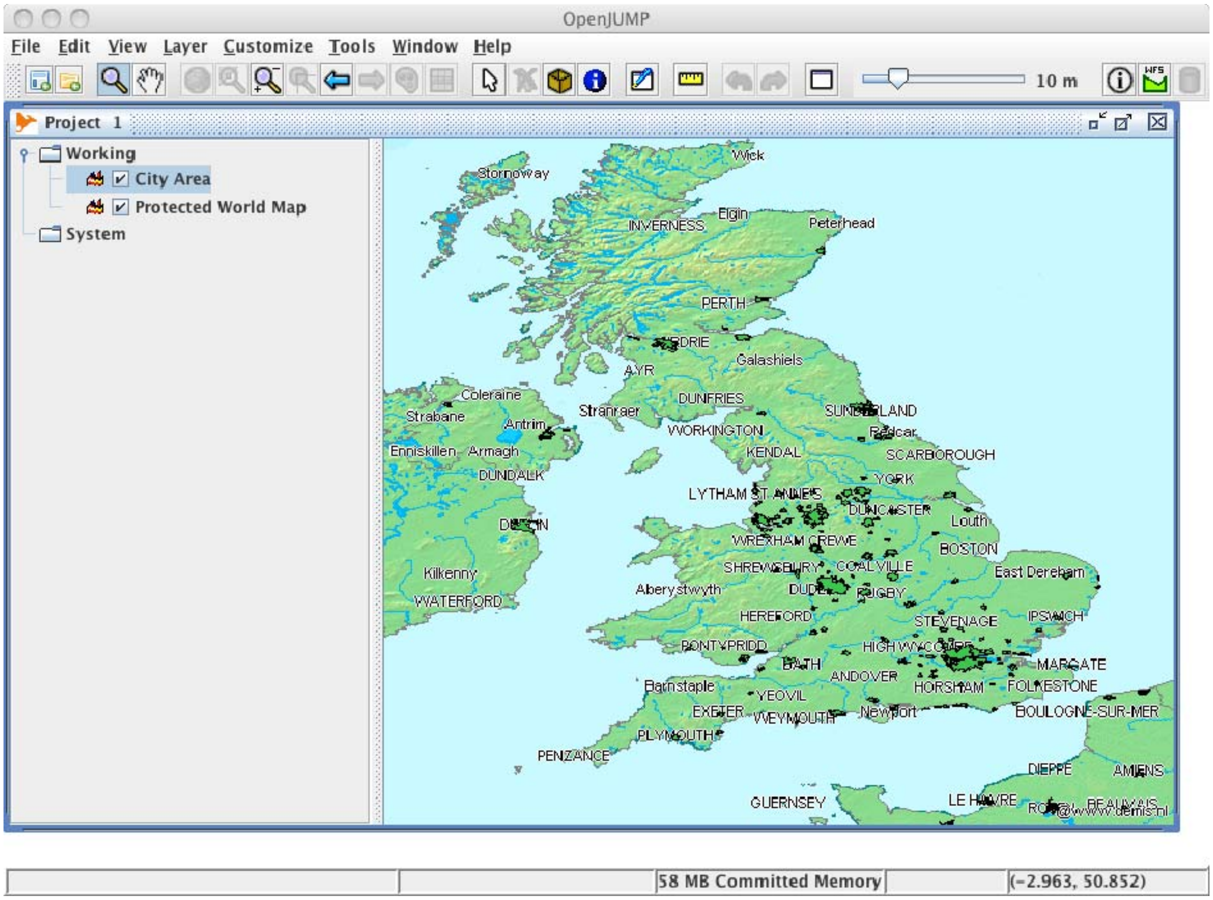


Figure 22 - OpenJump client showing a composite map of protected services from the federation

7.3.4 Technology Integration Experiments

For the purpose of this Authentication IE, we have developed two different types of clients.

The OpenLayers client, as illustrated earlier, is based on the currently available main stream OpenLayers JavaScript API release with no changes. When using the SP endpoints that engage the Web Browser SSO Profile, the client naturally qualifies for dealing with SAML based authentication.

In order to demonstrate ECP, we have extended the Open Source Desktop client implementation of OpenJump 1.3 available from sourceforge.net. We had to extend the available client to handle HTTPS, Cookies, Authentication and Redirect properly but also had to add functionality required for relaying SOAP messages as introduced in the ECP sequence diagram.

In order to demonstrate interoperability, we have setup different SPs provided by different organizations making available protected different types of OGC Web Services.

The following table provides the tentative list of Organizations and their SP endpoints, representing a protected WMS or WFS.

Service Type & Name	**Service Provider**	**Service GetCapabilities URL**
WMS Demis World Map	GeoRM	https://www.georm.org/service/WorldMap?Request=capabilities&SERVICE=WMS&VERSION=1.1.0
WMS Ordnance Survey Raster 250k	EDINA	https://esdin.edina.ac.uk:7111/cgi-mapserv/mapserv?map=mapfiles/raster250k.map&version=1.1.1&service=WMS&request=GetCapabilities

Table 1: List of organizations and their protected services as of March 2010

Please Note: At the time of writing this draft other participating organization may be in the process of setting up their SPs and IdPs.

As an integration experiment is only meaningful if different organizations provide client and service, we have EDINA developing the extension to the OpenJump client and GeoRM developing the OpenLayers client. In addition, different organizations have deployed a SAML conformant Service Provider. The main focus of integration is concerned with the fact that different organizations make their protected OGC Web Services available and that a combined use is possible with the two clients.

Service Provider / Service	OpenLayers Client (Web Browser SSO Profile)	OpenJump based Client (Enhanced Client Profile)
GeoRM / Demis World Map	√	√
EDINA / OS Raster 250k	√	√
EDINA / OS Strategi	√	√

TIE Matrix

7.3.5 Discussion of the approach concerning compliance to OGC Service Specifications

It is important to point out that we understand our demonstrated approach to be conformant with the current OGC Web Services specifications. As we are protecting a WMS of version 1.1.1 and a version 1.0 WFS, we consider it important to discuss why we think the approach is specification compliant.

The Web Browser SSO Profile is leveraging methods of the underlying transport protocol. For example, it uses HTTP status codes, HTTP cookies and HTTP Authentication. As the OGC Service specifications do not prevent the use of any features from the underlying protocol, we understand our approach to be compliant.

The Enhanced Client Profile is basically intercepting the GetCapabilities request by relaying SOAP messages between the IdP and SP. However, the client receives the capabilities document in case the user logged in successfully.

7.3.6 Summary and Future Work

The work undertaken for the Authentication IE was concerned with testing SAML 2 based Single-Sign-On authentication for a federation of protected OGC Web Services. We have provided two different types of clients to demonstrate interoperability to combine different protected services. We have proven that it is possible to protect OGC Web Services based on the existing specifications without modification to the interface. We have demonstrated that for the Web Map Service 1.1.1 and the Web Feature Service 1.0.

Please Note: Only WMSs have been setup at the March 2010 date of this draft.

Topics not addressed include inter-federation authentication and Single-Logout. The former is relevant whenever multiple federations shall provide access to protected services across federations. The latter is independent from cross-federation authentication but at least as important. Whenever a user gets authenticated, a session is created that gets invalid only if the application is closed. But in order to ensure that no other user can take advantage of an existing session, logout across all involved SPs should be ensured. We recommend further work towards Single-Logout before the SAML based authentication in a federation is used for a production system, as it cannot always be assumed that a user closes the application to invalidate the existing session by closing the application.

We have successfully demonstrated that the SAML Web Browser SSO Profile can be used to protect OGC Web Services with no change to the existing interface. However, the solution involves use of HTTP status codes, in particular 302 (temporarily moved) to redirect the request from the SP to the IdP. It is our belief that this is compliant with current OGC specifications and demonstrates good practice in how to leverage communication protocol features. We would like to encourage the OGC community to start discussing to correct the strict / limiting use of the underlying protocol features in a Best Practices paper.

7.3.7 Additional SAML Use Cases

Additional SAML Use Cases were provided by Land Information Ontario and are included as Appendix A to this report. These use cases provide valuable descriptions of SAML requests for authentication of a user through a web browser interface and a rich client interface.

Appendix A

SAML Use Case Document for Auth IE Engineering Report

1.0 Introduction

Security Access Markup Language (SAML) is an Organization for the Advancement of Structured Information Standards (OASIS) standard to provide a security protocol to access restricted data. SAML usage in a web services context can function as shown in Figure 1. Authentication parameters such as “username” and “password” are validated against a security database. A SAML component may require a third parameter: “data exchange”. Together the three parameters are used by SAML to confirm that the user has at least one data access privilege as made available through a “data exchange”. Once the user is authenticated, a SAML encrypted artefact is generated and returned as a response string. Artefacts remain cached in memory by the SAML provider application for a predefined period of time. The artefact will remain live until then. See Figure 1.

Figure 1: Sequence for SAML Authentication.

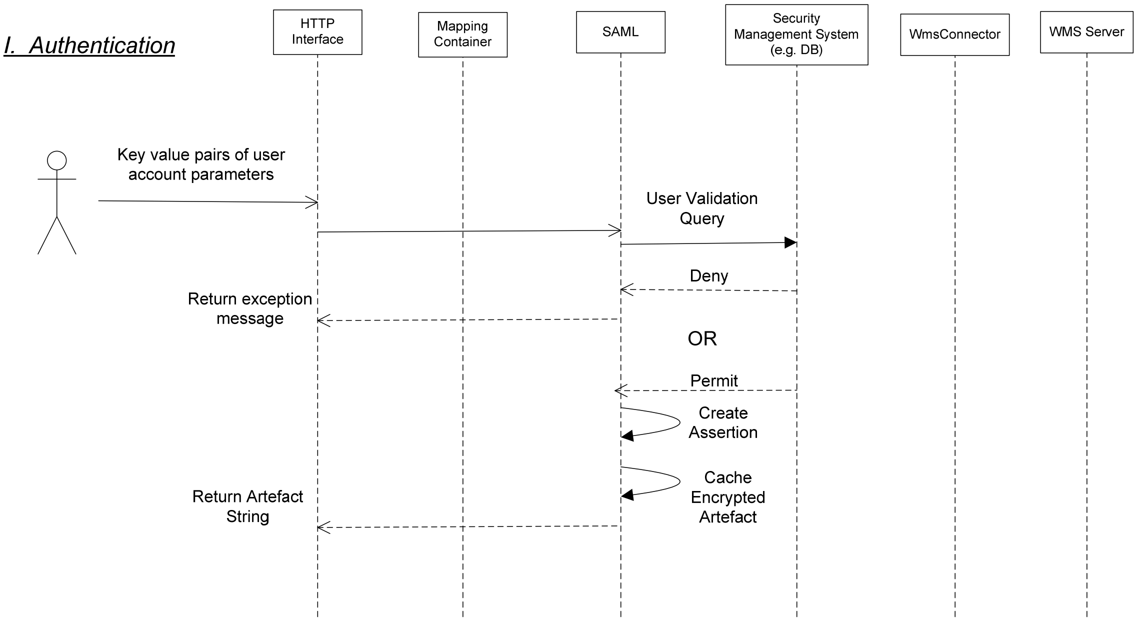
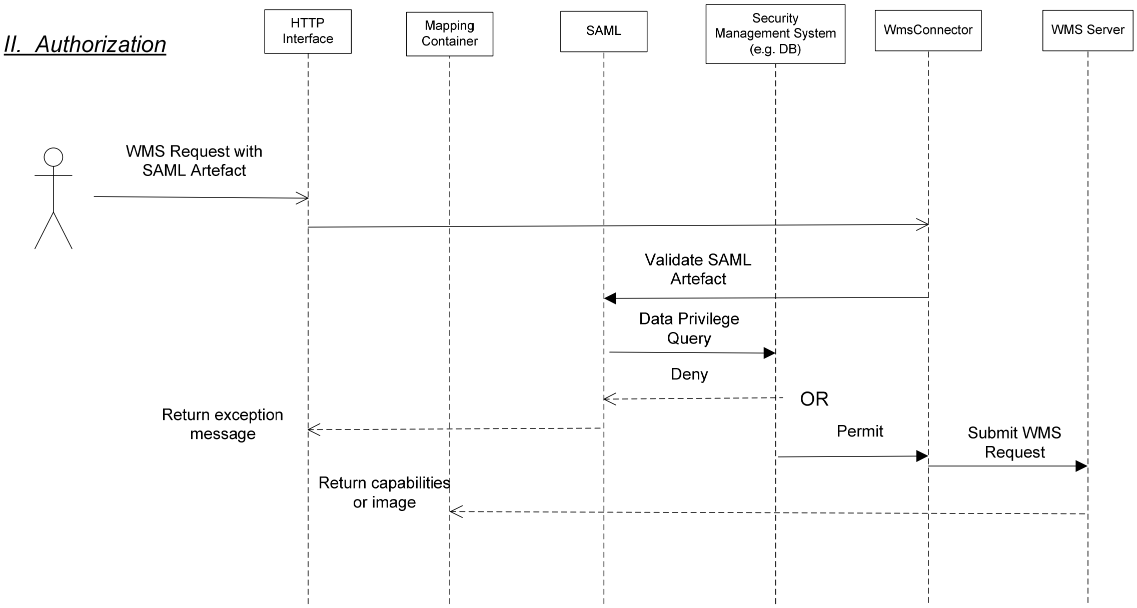


Figure 2: Sequence for SAML Authorization.



Any WMS request (i.e. GetCapabilities, GetMap) for a restricted WMS layer must include the SAML artefact attached to the URL string, for example:

http://www.lio.gov.on.ca/LioOgcWms21/lioogcwmsserver/wms_saml?request=GetMap&&info_format=text/plain&&styles=&&&&&layers=ARA_SURVEY_POINT&&srs=epsg:4326&&BBOX=-80,43,-79,44&&format=jpeg&&width=630&&height=630&&SAMLart=AAHyvDZcL3aXSDlWDPK

The SAML component proceeds with authorization using the artefact to confirm the user and the WMS layer request to determine what kind of access is permitted. For example, a GetCapabilities request will only return the Capabilities document that shows the layers for which the user has privileges, not any others that might be a part of the WMS. See Figure 2.

2.0 Assumptions and Integration

A security management system needs to exist to manage user accounts for SAML authentication and authorization purposes. SAML login is required to generate an encrypted artefact. This could be invoked through a user interface or as a SAML url request, for example:

3.0 SAML with Web Browser

3.1 Use Case: Web Browser and SAML Authentication

<table>
<tr><th colspan="6">UC_SAMLAuthenticationWithWebBrowser</th></tr>
<tr><td><b>Version</b></td><td>1.0</td><td>Date</td><td>2010-03-03</td><td>Author</td><td>R. Baehre</td></tr>
<tr><td><b>Summary</b></td><td colspan="5">SAML request for authentication of a user occurs through a web browser interface.</td></tr>
<tr><td><b>Goal</b></td><td colspan="5">To authenticate a user and retrieve an artefact.</td></tr>
<tr><td><b>Actors</b></td><td colspan="5">Web browser users.</td></tr>
<tr><td><b>Business Rules</b></td><td colspan="5">SAML must return an artefact, a default string value, or an exception.</td></tr>
<tr><td><b>Pre-conditions</b></td><td colspan="5">A user security system must exist that includes access privileges to specific WMS layers by specific data users.</td></tr>
<tr><td rowspan="6"><b>Basic Scenario: SAML login interface</b></td><td colspan="3"><b><i>User Responsibilities</i></b></td><td colspan="2"><b><i>System Responsibilities</i></b></td></tr>
<tr><td>1.</td><td colspan="2">Open Web Browser.</td><td colspan="2"></td></tr>
<tr><td>2.</td><td colspan="2">Request SAML login page.</td><td colspan="2"></td></tr>
<tr><td>3.</td><td colspan="2">Enter Username, Password and other parameter values such as Data Exchange.</td><td colspan="2"></td></tr>
<tr><td>4.</td><td colspan="2">Send request.</td><td colspan="2">GET action request to the SAML component. User security database or system is queried for authentication of user. If true, then SAML generates an artefact, caches it and returns it in a response object.</td></tr>
<tr><td>5.</td><td colspan="2">View SAML response. A positive result should show the artefact in web browser as a part of an xml string.</td><td colspan="2"></td></tr>
<tr><td rowspan="3"><b>Alternate Scenario: SAML login request url</b></td><td>1.</td><td colspan="2">Open Web Browser.</td><td colspan="2"></td></tr>
<tr><td>2.</td><td colspan="2">Enter url string to browser address that includes login parameters such as username, password, data exchange.</td><td colspan="2">e.g., http:</td></tr>
<tr><td>3.</td><td colspan="2">Repeat Steps 4 and 5 as above.</td><td colspan="2">GET action request to the SAML component. User security database or system is queried for authentication of user. If true, then SAML generates an artefact, caches it and returns it in a response object.</td></tr>
</table>

3.2 Use Case: Web Browser and SAML Authorization.

<table>
<tr><th colspan="6">UC_SAMLAuthorizationWithWebBrowser</th></tr>
<tr><td><b>Version</b></td><td>1.0</td><td>Date</td><td>2010-03-03</td><td>Author</td><td>R. Baehre</td></tr>
<tr><td><b>Summary</b></td><td colspan="5">SAML request for authorization of a user occurs through a web browser interface.</td></tr>
<tr><td><b>Goal</b></td><td colspan="5">To authorize a user using SAML and return a restricted wms layer.</td></tr>
<tr><td><b>Actors</b></td><td colspan="5">Web browser users.</td></tr>
<tr><td><b>Business Rules</b></td><td colspan="5">SAML must return an artefact, a default string value, or an exception.</td></tr>
<tr><td><b>Pre-conditions</b></td><td colspan="5">A user security system must exist that includes access privileges to specific wms layers by specific data users.</td></tr>
<tr><td rowspan="5"><b>Basic Scenario: GetCapabilities</b></td><td colspan="2"><b><i>User Responsibilities</i></b></td><td colspan="3"><b><i>System Responsibilities</i></b></td></tr>
<tr><td>1.</td><td colspan="2">Open Web Browser.</td><td colspan="2"></td></tr>
<tr><td>2.</td><td colspan="2">Enter wms request url for a GetCapabilities that includes the SAMLart parameter and artefact value in the browser address</td><td colspan="2"></td></tr>
<tr><td>3.</td><td colspan="2">Send request.</td><td colspan="2">GET action request to the SAML Filter Connector. User security database or system is queried for authorization of user. If true, then the appropriate response is generated.</td></tr>
<tr><td>4.</td><td colspan="2">View SAML response. A positive result should show the appropriate capabilities doc for the user and their layer access privileges assigned.</td><td colspan="2"></td></tr>
<tr><td rowspan="3"><b>Alternate Scenario: GetMap</b></td><td>1.</td><td colspan="2">Open Web Browser.</td><td colspan="2"></td></tr>
<tr><td>2.</td><td colspan="2">Enter wms request url for a GetMap that includes the SAMLart parameter and artefact value in the browser address</td><td colspan="2"></td></tr>
<tr><td>3.</td><td colspan="2">Repeat Steps 3 and 4 as above. If user authorized then the appropriate image should appear in the browser.</td><td colspan="2">GET action request to the SAML Filter Connector . User security database or system is queried for authorization of user. If true, then the appropriate response is generated.</td></tr>
<tr><td><b>Related Use Cases</b></td><td colspan="5">n/a</td></tr>
<tr><td><b>Business Owner</b></td><td colspan="5">LIO</td></tr>
<tr><td><b>Last Updated</b></td><td colspan="5">2009-11-25</td></tr>
</table>

4.0 SAML with Enriched Client

4.1 Use Case: Enriched Client and SAML Authentication.

The UC_SAMLAuthenticationWithWebBrowser use case can be used to retrieve a SAML artefact.

4.2 Use Case: Enriched Client and SAML Authorization.

UC_SAMLAuthorizationWithWebBrowser					
Version	1.0	Date	2010-03-03	Author	R. Baehre
Summary	SAML request for authorization of a user occurs through a rich client interface.				
Goal	To authorize a user using SAML and return a restricted wms layer.				
Actors	Web browser users.				
Business Rules	SAML must return an artefact, a null value, or an exception.				
Pre-conditions	A user security system must exist that includes access privileges to specific wms layers by specific individuals.				
Basic Scenario: Viewed authorized layer in client viewer	***User Responsibilities***			***System Responsibilities***	
	1.	Open Web Mapper or Viewer such as Gaia3.			
	2.	Retrieve SAML artefact (**UC_SAMLAuthenticationWithWebBrowser**)			
	3.	Add the secure wms to the list of wms available for adding layers to the viewer.			
	4.	Add the SAMLart parameter and artefact to the url string of the wms just added.			
	5.	Load the wms (send request) from the client application to retrieve the Capabilities doc.		GET action request to the SAML Filter Connector . User security database or system is queried for authorization of user. If true, then a Capabilities doc will be returned and loaded into the client.	
	6.	Select a layer. Add the SAMLart as a parameter to be added to the query string of the GetMap request that the client uses.			
	7.	Load the layer (send request) into a preview window or the full view window of the client.		GET action request to the SAML Filter Connector . User security database or system is queried for authorization of user. If true, then the appropriate image is returned.	
	8.	View the layer in the viewer main mapping window.			
Related Use Cases	n/a				
Business Owner	LIO				
Last Updated	2010-03-03				