NERC DATA GRID AUTHORISATION ARCHITECTURE

Neil Bennett[1], Ray Cramer[3], Glen Drinkwater[1], Marta Gutierrez[2], Phil Kershaw[2], Kerstin Kleese van Dam[1], Siva Kondapalli[3], Susan Latham[2], Bryan Lawrence[2], Roy Lowry[3], Ananta Manandhar[1], Kevin O'Neill[1], Ag Stephens[2], Shoaib Sufi[1], Andrew Woolf[1]

[1]CCLRC e-Science Centre
[2]British Atmospheric Data Centre
[3]British Oceanographic Data Centre

Abstract: The NERC Data Grid (NDG) uses RFC3820 proxy certificates for user Authentication. This paper focuses on the additional Token-based Authorisation mechanism used. Authorisation is required by NDG when users wish to access data or detailed metadata at many institutions. NDG users are allocated an Authorisation Token by their home institution that encapsulates details such as the issuing body, the user's data access roles and the token's expiry date. This token is an XML file, digitally signed using the XML-Security standard. The token can be used to access other data providers that have a relevant trust agreement with the user's own institution.

1. Introduction

The NERC Data Grid (NDG) is a collaboration of scientists from various domains including atmospheric scientists, oceanographers and ecologists across various institutions. NDG will enable these scientists to search for, discover and retrieve data that is relevant to their work. NDG development is still underway; the work detailed here represents one part of the jigsaw.

1.1. Overview

Anyone can search the NDG discovery metadata, and gain access to the high-level metadata (termed D-metadata within the NDG). When the user has found relevant high-level metadata and wishes to access the corresponding data or detailed metadata (termed B-metadata), they will require authorisation. Assuming they are permitted access, they can link to the actual data from either the B- or D- metadata.

There are two main steps for security. The initial step is authentication. This is followed by authorisation when the user wishes to go from summary to detailed metadata. Assignment of credentials to support these steps is not necessarily carried out at the same institution. This paper is concerned with authorisation.

The main structure of the authorisation system is as follows. Each data provider has an authorisation server (AS). The AS examines an incoming user's proxy certificate to identify them (based on distinguished name - DN). The AS uses this DN along with its user-role database to determine the user's data access roles. All relevant information is encapsulated in a digitally signed authorisation token.

Each time a user approaches a data provider to access data or B-metadata, the corresponding AS examines the user's tokens to see whether they have the required roles. Each AS possesses a mapping file or database mapping table. This contains details of role mappings with trusted organisations. New 'mapped' tokens can be generated from 'original' tokens if the data provider trusts an organisation from which the user holds an authorisation token. However, 'mapped' tokens cannot be used to generate further tokens via a trust relationship chain.

1.2. Architecture

NDG security has thus far been implemented in Java and uses Web Services to allow remote machines to communicate. This section describes that initial implementation, which uses Apache Axis.

When a user attempts to log into an NDG Portal, their username and password will be compared against those held by a MyProxy server. If there is a match, a corresponding proxy certificate is returned. This is then stored in a temporary 'user wallet' associated with the current user session.

```
<?xml version="1.0" encoding="UTF-8" ?>
<attributeCertificate>
        <acInfo>
                <version>1.0</version>
                <holder>/C=UK/O=eScience/OU=CLRC/L=DL/CN=neil bennett</holder>
                <issuer>EMAILADDRESS=ca-operator@grid-support.ac.uk, CN=CA, OU=Authority,
                        O=eScience, C=UK</issuer>
                <issuerName>BADC</issuerName>
                <issuerSerialNumber>1</issuerSerialNumber>
                <validity>
                        <notBefore>2005 0 19 12 23 37</notBefore>
                        <notAfter>2005 0 19 22 1 27</notAfter>
                </validity>
                <attributes>
                        <roleSet>
                                <role>
                                        <name>postdoc</name>
                                </role>
                                <role>
                                        <name>PhD student</name>
                                </role>
                        </roleSet>
                </attributes>
                <provenance>mapped</provenance>
        </acInfo>
       + <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
</attributeCertificate>
```

Figure 1 – Sample authorisation token with Signature element minimised (expanded in Figure 2).

Using the user DN extracted from the proxy certificate, a search is made in the persistent 'user repository' for the corresponding authorisation tokens. These tokens are also inserted into the 'user wallet'.

The contents of this wallet will then be read each time the user tries to access data or detailed metadata held by data providers. Access to such data / detailed metadata will be dealt with by an AS. A data provider may have its own AS or it may use the AS of another data provider that is willing to share.

When a user tries to access restricted data or metadata, the AS checks whether the user wallet contains a relevant token. If no such token exists, the AS takes the DN from the proxy certificate to see whether this is present in the user-role database, and if so generates a new token. If this token allows access to the data/metadata desired, then access is granted. Otherwise, access is refused and the user is informed of any relevant trust agreements with other data providers (section 2).

2. Authorisation Tokens and Trust Relationships

Each authorisation token (Figure 1) is an XML document and contains details such as who issued it, the user's DN, its dates of issue and expiry, the roles the user possesses, what algorithms/parameters were used to digitally sign the token and the digital signature itself.

The enveloped option of the XML Signature [1] standard has been used to represent signature information. This means the digital signature is expressed as an XML element within the token being signed (Figure 2). As the token contains the digital signature and the algorithms or parameters used to generate it, any recipient can verify it without referring to other files apart from any required to validate root signatures.

The Apache Security API [2] has been used to implement functionality that concerns creation/verification of the digital signatures.

Once a user has been assigned an authorisation token they will be able to use this to access data or detailed metadata held

```
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
    <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
    <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
    <Reference URI="">
      <Transforms>
        <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
        <Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments" />
      </Transforms>
      <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
      <DigestValue>EPxBQfM4r7A7+p76mNLANNN7T6E=</DigestValue>
    </Reference>
  </SignedInfo>
  <SignatureValue>WQ8vusc.....</SignatureValue>
  <KeyInfo>
    <X509Data>
      <X509Certificate>MIIFHzCCB.....</X509Certificate>
    </X509Data>
    <KeyValue>
      <RSAKeyValue>
        <Modulus>1D6Irf.....</Modulus>
        <Exponent>AQAB</Exponent>
      </RSAKeyValue>
    </KeyValue>
  </KeyInfo>
</Signature>
```

Figure 2 – Expanded Signature element for the sample authorisation token in Figure 1.

by hosts which have relevant trust agreements with the issuing data provider. The token contains the user's roles and details of the issuer. Each AS holds a mapping file or a mapping table in their user-role database showing how roles at external organisations map to local ones (Figure 3).

Thus, a new local token may be generated from a remote token and the role mappings. Each token is identified as 'original' or if generated from a role mapping, 'mapped'. 'Original' tokens can be used to generate new tokens via mapping relations, while 'mapped' tokens cannot.

In the circumstances where a user cannot present either 'original' or 'mapped' tokens, they may still be known to a remote data provider who could issue a token suitable for mapping (Figure 4). In such cases, the data holder automatically informs the user process of any data providers that have a trust agreement with it for the data desired. The user process can then obtain a suitable token for mapping, or the user can independently contact the relevant data provider to become appropriately qualified.

3. Authorisation Servers

The properties of an AS are set via a configuration file. This specifies the name of the data provider hosting the file, database connection & schema details, the digital signature algorithm to be used and authorisation token lifetime. A token is given the lifetime specified in the configuration file unless this exceeds the lifetime of the proxy certificate from which it was generated.

The configuration file also contains details of the format of DNs used by trusted hosts. This is important because data providers use DNs to identify users. The AS will check to see whether the user's DN is present in the user-role database. If it is, they are a local user and if not, they are classed as external. If the user is external and is a member of a trusted organisation, a check is made to see whether there is an appropriate role mapping. Thus, the AS needs to understand what external DNs represent so that it can translate them to the local format and search for them in the database.

```
<?xml version="1.0"?>
<AAmap>
        <trusted name='BADC'>
                <role remote='PhD_student' local='deltaflume'/>
                <role remote='PhD_student' local='COAST_OBS'/>
                <role remote='postdoc' local='deltaflume'/>
                <role remote='postdoc' local='BODC'/>
                <role remote='postdoc' local='mfmb'/>
        </trusted>
        <trusted name='CEH'>
                <role remote='lakes' local='deltaflume'/>
                <role remote='ECN' local='deltaflume'/>
                <role remote='all' local='BODC'/>
                <role remote='countryside survey' local='mfmb'/>
        </trusted>
</AAmap>
```

Figure 3 – Example of a mapping file for BODC (British Oceanographic Data Centre) showing trust relationships with the BADC (British Atmospheric Data Centre) and CEH (Centre for Ecology & Hydrology).

The decision over whether the mapping is held in a file or a database table is also made in the configuration file.

4. Future

The NDG authorisation architecture is currently being applied at the BADC, and will shortly be adopted by the CEH in the EcoGrid project.

Developments in authorisation standards are being closely monitored to see how the system could be improved.

5. References

[1] XML-Signature Syntax & Processing, www.w3.org/TR/xmldsig-core/

[2] Apache XML Project, http://xml.apache.org/security/

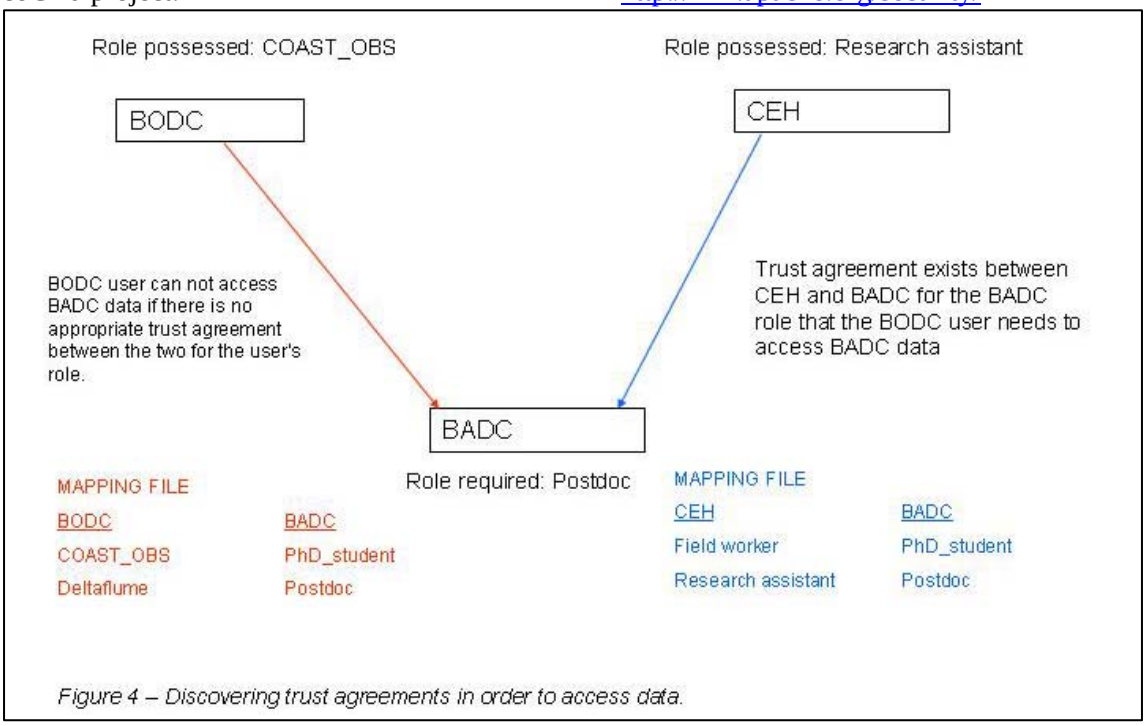

Figure 4 – Discovering trust agreements in order to access data.