

GWD-I (proposed)

3/5/2009

GWD-I (proposed)  
Category – Informational  
OGSA Authorization working group

Vincenzo Ciaschini, INFN CNAF  
Valerio Venturi, INFN CNAF  
Andrea Ceccanti, INFN CNAF

**Sep 11, 2006**

## **The VOMS Attribute Certificate Format**

### Status of This Memo

This document provides a description of the VOMS AC. Distribution is unlimited.

### Copyright Notice

Copyright © Global Grid Forum (2004-2005). All Rights Reserved.

### **Abstract**

This document provides a complete description of the VOMS AC format, both syntax and semantics. It also describe the related extensions that must be used in a proxy certificate to make it fully VOMS-compliant.

Contents

Abstract.....	1
1. Introduction.....	3
2. Conventions used in this Specification .....	3
3. AC Format .....	3
3.1 Holder .....	4
3.1.1 Syntax.....	4
3.1.2 Semantics .....	4
3.2 AttCertIssuer .....	5
3.2.1 Syntax.....	5
3.2.2 Semantics .....	5
3.3 AttCertValidityPeriod .....	5
3.3.1 Syntax & Semantics .....	5
3.4 Attribute .....	5
3.4.1 Attribute Fully Qualified Attribute Name (FQAN).....	5
3.4.1.1 Syntax.....	5
3.4.1.2 Semantics .....	6
3.4.1.3 Examples .....	6
3.5 Extensions .....	7
3.5.1 ACTarget.....	7
3.5.1.1 Syntax.....	7
3.5.1.2 Semantics .....	7
3.5.2 NoRevAvail .....	7
3.5.2.1 Syntax.....	7
3.5.2.2 Semantics .....	7
3.5.3 IssuerCerts.....	8
3.5.3.1 Syntax.....	8
3.5.3.2 Semantics .....	8
3.5.4 Tags.....	8
3.5.4.1 Syntax.....	8
3.5.4.2 Semantics .....	8
3.6 IssuerUniqueID .....	9
4. VOMS compliant proxy certificates .....	9
4.1 AC Sequence .....	9
4.1.1 Syntax.....	9
4.1.2 Semantics .....	9
4.2 KeyUsage extension.....	9
4.3 Obsolete Extensions.....	9
5. Non-normative example .....	10
6. Security Considerations .....	23
Author Information.....	23
Intellectual Property Statement.....	23
Full Copyright Notice .....	23
Normative References.....	24
Informational References .....	24
Appendix A. ChangeLog .....	24

## 1. Introduction

This document is a companion to the "Attributes used in OGSA Authorization" GWD 57 [OGSI-Authz-Attr] and also requires knowledge of RFC 3281, RFC 3280, RFC 3820, and assumes that the reader is familiar with those documents, though to simplify understanding, part of the information from those documents will be duplicated here.

Attribute Certificates (ACs) provide a standardized method to associate a set of attributes to an identity. However, they may be created in thousand of different ways and so it becomes necessary to also have a complete description of the format of an AC before you can use it.

The aim of this document is to provide a complete specification of the Attribute Certificates (AC) generated by VOMS, to simplify and insure interoperability about services that need to parse and interpret them.

Section 2 will give a very brief account of conventions and abbreviations used in this specification, Section 3 will document the format of the AC, while section 4 will document how ACs are included in a proxy certificate. Section 5 will present a (non-normative) example of a proxy containing VOMS informations. Finally, section 6 will briefly talk about security considerations.

## 2. Conventions used in this Specification

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC2119].

The following abbreviations will also be used: AC (Attribute Certificate), AA (Attribute Authority), PKC (Public Key Certificate), FQHN (Fully Qualified Host Name)

## 3. AC Format

This is the general format of an AC as defined by RFC 3281. Customizations used by VOMS will be discussed in individual subsections. Everything not specifically mentioned here is intended to be in accordance with RFC 3281.

```
AttributeCertificate ::= SEQUENCE {
    acinfo           AttributeCertificateInfo,
    signatureAlgorithm AlgorithmIdentifier,
    signatureValue   BIT STRING
}

AttributeCertificateInfo ::= SEQUENCE {
    version           AttCertVersion,
    holder            Holder,
    issuer            AttCertIssuer,
    signature         AlgorithmIdentifier,
    serialNumber      CertificateSerialNumber,
    attrCertValidityPeriod AttCertValidityPeriod,
    attributes        SEQUENCE OF Attribute,
    issuerUniqueID    UniqueIdentifier OPTIONAL,
    extensions        Extensions OPTIONAL
}

AttCertVersion ::= INTEGER { v2(1) }

Holder ::= SEQUENCE {
    baseCertificateID [0] IssuerSerial OPTIONAL,
    entityName        [1] GeneralNames OPTIONAL,
    objectDigestInfo  [2] ObjectDigestInfo OPTIONAL
}
```

```

AttCertIssuer ::= CHOICE {
  v2Form      [0] V2Form
}

V2Form ::= SEQUENCE {
  issuerName      GeneralNames OPTIONAL,
  baseCertificateID [0] IssuerSerial OPTIONAL,
  objectDigestInfo [1] ObjectDigestInfo OPTIONAL
}

IssuerSerial ::= SEQUENCE {
  issuer      GeneralNames,
  serial      CertificateSerialNumber,
  issuerUID   UniqueIdentifier OPTIONAL
}

AttCertValidityPeriod ::= SEQUENCE {
  notBeforeTime GeneralizedTime,
  notAfterTime  GeneralizedTime
}

Attribute ::= SEQUENCE {
  type      AttributeType,
  values    SET OF AttributeValue
  -- at least one value is required
}

AttributeType ::= OBJECT IDENTIFIER

AttributeValue ::= ANY DEFINED BY AttributeType

```

Also, the `voms` OID is defined and reserved for VOMS uses, and its value is 1.3.6.1.4.1.8005.100.100

### 3.1 Holder

#### 3.1.1 Syntax

The holder of a VOMS AC **MUST** always be an X.509 PKC. As a consequence of this, in VOMS ACs the only admissible choice for the field is the `baseCertificateID`, while `entityName` and `objectDigestInfo` **MUST** be absent. This means that the `IssuerSerial` structure **MUST** be used for this field.

#### 3.1.2 Semantics

The `issuer` and `serial` fields **MUST** be copies of those in the holder's PKC, while the `issuerUID` field is usually empty. It **MUST** be present if and only if it is also present in the holder's PKC, and in this case they **MUST** have the same value. Note that RFC 3280 says that conforming implementations of PKCs **SHOULD NOT** use this field, but that implementations **SHOULD** be capable to handle it.

Note that the holder here is the user's own PKC, and **NOT** the proxies he may use.

## 3.2 AttCertIssuer

### 3.2.1 Syntax

RFC 3281 requires that the `v2Form` MUST be used to specify this field. Furthermore, it requires that only the `issuerName` field be used and that it MUST contain just one `GeneralName`, which in turn must contain only a single distinguished name in its `dir` field.

### 3.2.2 Semantics

The included distinguished name MUST be the distinguished name of the issuer's own PKC. This in turn requires that the issuer's PKC MUST have a non-empty distinguished name, as required by RFC 3281.

## 3.3 AttCertValidityPeriod

### 3.3.1 Syntax & Semantics

This is a standard validity field. Its values should be expressed in UTC format, with seconds always included.

## 3.4 Attribute

This is the core of the AC, where the important data actually is. The following subsections will describe the attributes that are used by VOMS. Further attributes than those defined here MAY be present and, if so, conforming application MAY choose to ignore them.

The attributes `Role` and `Group` defined in RFC 3281 SHOULD NOT be used. Instead a new attribute, `FQAN` (see below) is defined.

### 3.4.1 Attribute Fully Qualified Attribute Name (FQAN)

#### 3.4.1.1 Syntax

This attribute uses the following syntax:

```
name       : voms-attribute
OID        : { voms 4 }
syntax     : IetfAttrSyntax
values     : Multiple not allowed
```

`IetfAttrSyntax` is defined in RFC 3281 and reprinted here for convenience:

```
IetfAttrSyntax ::= SEQUENCE {
  policyAuthority [0] GeneralNames OPTIONAL,
  values SEQUENCE OF CHOICE {
    octets OCTET STRING,
    oid OBJECT IDENTIFIER,
    string UTF8String
  }
}
```

The `policyAuthority` field of the `IetfAttrSyntax` MUST be present and MUST contain an encoding of both the VO to which the AC issuer belongs and the server which generated this particular attribute, in the following format: `<vo name>://<fqhn>:<port>`, where the characters '`<`' and '`>`' are only used to highlight the field names and should not be used in the actual encoding.

This attribute **MUST** be present in a conforming AC. If multiple values are needed (and usually they are), they can be encoded in the `values SEQUENCE`. The `octets` encoding of `values` **MUST** be used.

#### 3.4.1.2 Semantics

This attribute encodes the position of the Holder inside the VO. A Holder may be a member of several groups, and he may hold a special role inside some of his groups.

Groups are organized in a tree structure, meaning that a group may have subgroups, which in turn may have subgroups, etc... The group name is then represented in the following way:

```
/<root group>/<subgroup>/.../<subgroup>
```

Where `<root group>` **MUST** be the name of the virtual organization.

Roles are not organized in a hierarchical structure. Ownership of a role is always associated to membership in a group.

All groups of which the Holder is a member are represented in the attribute, but no information on role ownership is represented unless the Holder specifically asked for it while contacting the Attribute Authority (AA). This is indeed the main difference between groups and roles: group membership is compulsory and cannot be denied, while role ownership is an optional thing that the holder may or may not want to be specified.

This information is encoded in a Fully Qualified Attribute Name (FQAN), in the following format:

```
<group name>/Role=<role name>/Capability=<capability name>
```

This syntax means that the user holds the role `<role name>` in the group `<group name>`. If no specific role is held, the `<role name>` is `NULL`. The `/Capability=<capability name>` part is deprecated and will disappear in the future: conforming applications **SHOULD** be able to handle FQANs where it is absent and **SHOULD NOT** rely on its presence.

Future compatibility issue: It is possible that in the future a `/Role=NULL` component may be omitted in its entirety. The same goes for a `/Capability=NULL` part. Conforming applications **SHOULD** be prepared to handle these cases.

The order in which the FQANs are present in the attribute is significant, since it is the order in which the Holder wished the FQANs to be evaluated. Conforming applications **SHOULD** be capable of accepting an unlimited number of FQANs, however if an application is not capable of this but is limited to accept only `n`, then they **MUST** be the first `n` present in this extension. In particular, if an application can accept only one FQAN, then it **MUST** be the first one.

#### 3.4.1.3 Examples

Examples of valid FQANs:

```
/cms/Role=NULL/Capability=NULL
```

```
/cms/Role=VO-Admin/Capability=NULL
```

```
/cms/Role=sgm/Capability=NULL
```

```
/cms/production/Role=NULL/Capability=NULL
```

```
/cms/production/Role=writer/Capability=NULL
```

```
/cms/analysis/Role=NULL/Capability=NULL
```

The same FQANs in the compact format:

```
/cms
```

```
/cms/Role=VO-Admin
```

```
/cms/Role=sgm
```

```

/cms/production
/cms/production/Role=writer
/cms/analysis

```

### 3.5 Extensions

Here will be defined the extensions that are defined for use into the VOMS AC. Other extensions may still be present, but they **MUST NOT** be critical.

#### 3.5.1 ACTarget

##### 3.5.1.1 Syntax

```

name          id-ce-targetInformation
OID           { id-ce 55 }
syntax        SEQUENCE OF Targets
criticality    MUST be TRUE

Targets ::= SEQUENCE OF Target

Target ::= CHOICE {
  targetName   [0] GeneralName,
  targetGroup  [1] GeneralName,
  targetCert   [2] TargetCert
}

TargetCert ::= SEQUENCE {
  targetCertificate  IssuerSerial,
  targetName         GeneralName OPTIONAL,
  certDigestInfo    ObjectDigestInfo OPTIONAL
}

```

When this extension is used `targetName` **MUST** be the chosen encoding. It must contain the URI of some resources, encoded in the `IA5STRING` format.

If this extensions is present, conforming applications **MUST** honor it.

##### 3.5.1.2 Semantics

The intent of this extension is to be able to specify the exact set of targets where the AC can be accepted. Every other target **SHOULD** refuse should be refused regardless of everything else. To this intent, the content of the extension is supposed to be a set of fully qualified domain names, indicating where verification of the AC can succeed.

#### 3.5.2 NoRevAvail

##### 3.5.2.1 Syntax

```

name          id-ce-noRevAvail
OID           { id-ce 56 }
syntax        NULL (i.e. '0500'H is the DER encoding)
criticality    MUST be FALSE

```

##### 3.5.2.2 Semantics

The intent of this extension is to specify that CRL for the AA may not exist, and even should they exist they will **NEVER** refer to this AC

### 3.5.3 IssuerCerts

#### 3.5.3.1 Syntax

```
name          pk-cert-list
OID           { voms 10 }
syntax        X509_CERTS
criticality   MUST be FALSE
```

```
X509_CERTS ::= SEQUENCE OF X509Certificate
```

#### 3.5.3.2 Semantics

This extension is meant to include the AA's public key certificate and the whole certificate chain leading to it, up to and excluding the CA certificate that is expected to be on the evaluator's machine (typically, the root CA).

If this extension is present, the evaluator MAY choose to use this certificate to verify the AC, but it is suggested that the evaluator has some other method to know that this certificate identifies a trusted AA.

### 3.5.4 Tags

#### 3.5.4.1 Syntax

```
name          : tags
OID           : { voms 11 }
syntax        : TagContainer
values        : Multiple not allowed
```

```
TagContainer ::= SEQUENCE OF TagList
TagList ::= SEQUENCE {
    policyAuthority GeneralNames,
    tags SEQUENCE OF Tag
}
Tag ::= SEQUENCE {
    name OCTECT STRING
    value OCTECT STRING
    qualifier OCTECT STRING
}
```

The `policyAuthority` field follows the syntax as the homonymous field in the FQAN attribute and, if both attributes are present, they MUST be set to the same value.

#### 3.5.4.2 Semantics

The intent of this extension is to provide a way to specify attributes that do not map well in the group/role paradigm. It allows for this by specifying a set of (name, value, qualifier) triples that could be used to describe almost everything.

Not all conforming applications need to be understand the semantics associated to each `Tag` name. In such a case the tags may be safely ignored.

If multiple `values` need to be associated with a single `name`, it is possible to use several (name, value, qualifier) `Tags` with the same name, and possibly different qualifiers. A name-specific

syntax that encodes multiple values in a single pair is also allowed. Conforming applications that are aware of a specific `name` MUST consider the two syntaxes as completely equivalent. The same `Tag` MUST NOT appear more than once.

The `policyAuthority` is specific to each `TagList` object, and MUST indicate the name of the authority that is the source of the enclosed tags.

### 3.6 IssuerUniqueID

This field should be present if and only if it is also present in the issuer's certificate, in which case the two MUST be identical

## 4. VOMS compliant proxy certificates

ACs, once created, need to be available to application that must evaluate them. To both maintain the single login feature of the grid, and to let the user choose what ACs to present to an application, the best thing is to include them in the proxy certificate. The intent of this section is to show how this is done.

Here will be defined the extensions that are defined in VOMS-compliant proxies. Other extensions may be used at will.

### 4.1 AC Sequence

#### 4.1.1 Syntax

```
name: acseq
OID: { voms 5 }
Syntax: acSequence
```

```
acSequence = SEQUENCE OF AttributeCertificate
```

#### 4.1.2 Semantics

This is the way to include ACs generated by VOMS inside a certificate. They should be included in the order in which they were requested. Conforming applications that are not capable of accepting multiple ACs SHOULD at least accept the first, the "Default" VO.

### 4.2 KeyUsage extension

Though not necessary, empirical tests have shown that proxies lacking this extension may not successfully be used between different versions of globus. For this reason, this extension SHOULD always be present.

### 4.3 Obsolete Extensions

Due to compatibility with old, pre-AC version of VOMS, it is possible to find in VOMS proxies extensions with OID 1.3.6.1.4.1.8005.100.100.6 and 1.3.6.1.4.1.8005.100.100.1. These are obsolete now and can be safely ignored. For this reason, their syntax and semantics is not documented.

## 5. Non-normative example

This section will present an example of a proxy certificate containing an AC issued by VOMS. Please note that for simplicity, this is a Globus Toolkit 2-compatible proxy certificate.

```

0 2708: SEQUENCE {
4 2428: SEQUENCE {
8   3:   [0] {
10  1:   INTEGER 2
      :   }
13  2:   INTEGER 4033
17 13:   SEQUENCE {
19  9:   OBJECT IDENTIFIER md5withRSAEncryption (1 2 840 113549 1 1 4)
30  0:   NULL
      :   }
32 103:  SEQUENCE {
34  11:  SET {
36  9:   SEQUENCE {
38  3:   OBJECT IDENTIFIER countryName (2 5 4 6)
43  2:   PrintableString 'IT'
      :   }
      :   }
47 13:  SET {
49  11:  SEQUENCE {
51  3:   OBJECT IDENTIFIER organizationName (2 5 4 10)
56  4:   PrintableString 'INFN'
      :   }
      :   }
62 29:  SET {
64 27:  SEQUENCE {
66  3:   OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
71 20:  PrintableString 'Personal Certificate'
      :   }
      :   }
93 13:  SET {
95 11:  SEQUENCE {
97  3:   OBJECT IDENTIFIER localityName (2 5 4 7)
102 4:   PrintableString 'CNAF'
      :   }
      :   }
108 27: SET {
110 25: SEQUENCE {
112  3:   OBJECT IDENTIFIER commonName (2 5 4 3)
117 18:  PrintableString 'Vincenzo Ciaschini'
      :   }
      :   }
      :   }

```

```

137 30: SEQUENCE {
139 13:     UTCTime 22/09/2006 13:45:15 GMT
154 13:     UTCTime 23/09/2006 01:50:15 GMT
      :     }
169 123: SEQUENCE {
171 11:     SET {
173  9:         SEQUENCE {
175  3:             OBJECT IDENTIFIER countryName (2 5 4 6)
180  2:             PrintableString 'IT'
      :         }
      :     }
184 13:     SET {
186 11:         SEQUENCE {
188  3:             OBJECT IDENTIFIER organizationName (2 5 4 10)
193  4:             PrintableString 'INFN'
      :         }
      :     }
199 29:     SET {
201 27:         SEQUENCE {
203  3:             OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
208 20:             PrintableString 'Personal Certificate'
      :         }
      :     }
230 13:     SET {
232 11:         SEQUENCE {
234  3:             OBJECT IDENTIFIER localityName (2 5 4 7)
239  4:             PrintableString 'CNAF'
      :         }
      :     }
245 27:     SET {
247 25:         SEQUENCE {
249  3:             OBJECT IDENTIFIER commonName (2 5 4 3)
254 18:             PrintableString 'Vincenzo Ciaschini'
      :         }
      :     }
274 18:     SET {
276 16:         SEQUENCE {
278  3:             OBJECT IDENTIFIER commonName (2 5 4 3)
283  9:             PrintableString '781094397'
      :         }
      :     }
294 92:     SEQUENCE {
296 13:         SEQUENCE {
298  9:             OBJECT IDENTIFIER rsaEncryption (1 2 840 113549 1 1 1)
309  0:             NULL

```

```

:      }
311 75: BIT STRING, encapsulates {
314 72: SEQUENCE {
316 65: INTEGER
:      00 A0 96 28 AA 34 B3 28 EB 22 69 15 30 D8 CB 7C
:      5F 3D 2A F2 4D B4 15 F8 00 54 E1 97 32 24 04 77
:      C2 F6 5B F2 0F 03 3C 99 EE 96 6C E9 CA 8E CD 0B
:      6B EF A9 78 4E A3 7D 07 E7 91 42 69 0B 9C 10 21
:      FD
383 3: INTEGER 65537
:      }
:      }
:      }
388 2044: [3] {
392 2040: SEQUENCE {
396 1971: SEQUENCE {
400 10: OBJECT IDENTIFIER '1 3 6 1 4 1 8005 100 100 5'
412 1955: OCTET STRING, encapsulates {
416 1951: SEQUENCE {
420 1947: SEQUENCE {
424 1943: SEQUENCE {
428 1792: SEQUENCE {
432 1: INTEGER 1
435 79: SEQUENCE {
437 77: [0] {
439 71: SEQUENCE {
441 69: [4] {
443 67: SEQUENCE {
445 11: SET {
447 9: SEQUENCE {
449 3: OBJECT IDENTIFIER
:      countryName (2 5 4 6)
454 2: PrintableString 'IT'
:      }
:      }
458 13: SET {
460 11: SEQUENCE {
462 3: OBJECT IDENTIFIER
:      organizationName (2 5 4 10)
467 4: PrintableString 'INFN'
:      }
:      }
473 37: SET {
475 35: SEQUENCE {
477 3: OBJECT IDENTIFIER
:      commonName (2 5 4 3)

```

```

482 28:          PrintableString 'INFN Certification Authority'
      :          }
      :          }
      :          }
      :          }
      :          }
512 2:          INTEGER 4033
      :          }
      :          }
516 96:         [0] {
518 94:         SEQUENCE {
520 92:         [4] {
522 90:         SEQUENCE {
524 11:         SET {
526 9:         SEQUENCE {
528 3:         OBJECT IDENTIFIER
      :         countryName (2 5 4 6)
533 2:         PrintableString 'IT'
      :         }
      :         }
537 13:        SET {
539 11:        SEQUENCE {
541 3:        OBJECT IDENTIFIER
      :        organizationName (2 5 4 10)
546 4:        PrintableString 'INFN'
      :        }
      :        }
552 13:        SET {
554 11:        SEQUENCE {
556 3:        OBJECT IDENTIFIER
      :        organizationalUnitName (2 5 4 11)
561 4:        PrintableString 'Host'
      :        }
      :        }
567 13:        SET {
569 11:        SEQUENCE {
571 3:        OBJECT IDENTIFIER
      :        localityName (2 5 4 7)
576 4:        PrintableString 'CNAF'
      :        }
      :        }
582 30:        SET {
584 28:        SEQUENCE {
586 3:        OBJECT IDENTIFIER
      :        commonName (2 5 4 3)
591 21:        PrintableString 'datatag6.cnaf.infn.it'

```

```

:           }
:           }
:           }
:           }
:           }
:           }
614 13:     SEQUENCE {
616  9:     OBJECT IDENTIFIER
:           md5withRSAEncryption (1 2 840 113549 1 1 4)
627  0:     NULL
:           }
629  4:     INTEGER 415417170
635 34:     SEQUENCE {
637 15:     GeneralizedTime 22/09/2006 13:50:14 GMT
654 15:     GeneralizedTime 23/09/2006 01:50:14 GMT
:           }
671 186:    SEQUENCE {
674 183:    SEQUENCE {
677 10:    OBJECT IDENTIFIER '1 3 6 1 4 1 8005 100 100 4'
689 168:    SET {
692 165:    SEQUENCE {
695 39:    [0] {
697 37:    [6] 'valerio://datatag6.cnaf.infn.it:50002'
:           }
736 122:    SEQUENCE {
738 34:    OCTET STRING '/valerio/Role=NULL/Capability=NULL'
774 41:    OCTET STRING
:           '/valerio/asdasd/Role=NULL/Capability=NULL'
817 41:    OCTET STRING
:           '/valerio/qwerty/Role=NULL/Capability=NULL'
:           }
:           }
:           }
:           }
:           }
860 1360:   SEQUENCE {
864 123:   SEQUENCE {
866 10:   OBJECT IDENTIFIER '1 3 6 1 4 1 8005 100 100 11'
878 109:   OCTET STRING, encapsulates {
880 107:   SEQUENCE {
882 105:   SEQUENCE {
884 103:   SEQUENCE {
886 39:   SEQUENCE {
888 37:   [6] 'valerio://datatag6.cnaf.infn.it:50002'
:           }
927 60:   SEQUENCE {

```

```

929 28: SEQUENCE {
931 12:     OCTET STRING 'attributeOne'
945  3:     OCTET STRING '111'
950  7:     OCTET STRING 'valerio'
      :     }
959 28: SEQUENCE {
961 12:     OCTET STRING 'attributeTwo'
975  3:     OCTET STRING '222'
980  7:     OCTET STRING 'valerio'
      :     }
      :     }
      :     }
      :     }
      :     }
      :     }
      :     }
889  9: SEQUENCE {
991  3:     OBJECT IDENTIFIER '2 5 29 56'
996  2:     OCTET STRING, encapsulates {
998  0:     NULL
      :     }
      :     }
1000 31: SEQUENCE {
1002  3:     OBJECT IDENTIFIER
      :     authorityKeyIdentifier (2 5 29 35)
1007 24:     OCTET STRING, encapsulates {
1009 22:         SEQUENCE {
1011 20:             [0]
      :             00 95 79 11 CC B8 9B D2 1D 2A 19 84 C0 36 8B F6
      :             E8 9E B3 FE
      :             }
      :         }
      :     }
1033 1187: SEQUENCE {
1037 10:     OBJECT IDENTIFIER '1 3 6 1 4 1 8005 100 100 10'
1049 1171:     OCTET STRING, encapsulates {
1053 1167:         SEQUENCE {
1057 1163:             SEQUENCE {
1061 1159:                 SEQUENCE {
1065  879:                     SEQUENCE {
1069  3:                         [0] {
1071  1:                             INTEGER 2
      :                             }
1074  2:                             INTEGER 4675
1078 13:                             SEQUENCE {
1080  9:                                 OBJECT IDENTIFIER

```

```

:
:
1091 0:      NULL
:
1093 67:    SEQUENCE {
1095 11:    SET {
1097 9:      SEQUENCE {
1099 3:      OBJECT IDENTIFIER
:          countryName (2 5 4 6)
1104 2:      PrintableString 'IT'
:
:
1108 13:    SET {
1110 11:    SEQUENCE {
1112 3:      OBJECT IDENTIFIER
:          organizationName (2 5 4 10)
1117 4:      PrintableString 'INFN'
:
:
1123 37:    SET {
1125 35:    SEQUENCE {
1127 3:      OBJECT IDENTIFIER
:          commonName (2 5 4 3)
1132 28:    PrintableString 'INFN Certification
Authority'
:
:
:
1162 30:    SEQUENCE {
1164 13:    UTCTime 15/05/2006 12:10:02 GMT
1179 13:    UTCTime 15/05/2007 12:10:02 GMT
:
1194 90:    SEQUENCE {
1196 11:    SET {
1198 9:      SEQUENCE {
1200 3:      OBJECT IDENTIFIER
:          countryName (2 5 4 6)
1205 2:      PrintableString 'IT'
:
:
1209 13:    SET {
1211 11:    SEQUENCE {
1213 3:      OBJECT IDENTIFIER
:          organizationName (2 5 4 10)
1218 4:      PrintableString 'INFN'
:
:
1224 13:    SET {

```

```

1226 11: SEQUENCE {
1228 3:   OBJECT IDENTIFIER
      :   organizationalUnitName (2 5 4 11)
1233 4:   PrintableString 'Host'
      :   }
      :   }
1239 13: SET {
1241 11:   SEQUENCE {
1243 3:     OBJECT IDENTIFIER
      :     localityName (2 5 4 7)
1248 4:     PrintableString 'CNAF'
      :     }
      :     }
1254 30: SET {
1256 28:   SEQUENCE {
1258 3:     OBJECT IDENTIFIER
      :     commonName (2 5 4 3)
1263 21:     PrintableString 'datatag6.cnaf.infn.it'
      :     }
      :     }
1286 159: SEQUENCE {
1289 13:   SEQUENCE {
1291 9:     OBJECT IDENTIFIER
      :     rsaEncryption (1 2 840 113549 1 1 1)
1302 0:     NULL
      :     }
1304 141:   BIT STRING, encapsulates {
1308 137:     SEQUENCE {
1311 129:       INTEGER
      :       00 B4 0B 0B B9 DF F7 FC E0 E8 62 AE 32 A2 9D D5
      :       B1 2C F9 D4 0F 1C AC EA 27 6E 24 8B 14 68 39 FF
      :       FB 61 AE 4A 4B AC 69 62 43 8F B2 91 99 FD 23 67
      :       BE 6F 39 4D 94 9F 8C 1A D9 A6 44 B2 93 DB 1D 31
      :       DB D0 E6 B3 D8 7E 6E 97 40 E9 A4 F8 9F 83 CB D3
      :       D3 81 8F 40 5C B8 DE 39 BB 19 06 60 EC 30 DC 2B
      :       99 1F 30 68 F0 6B 05 5D 28 ED D6 52 8F 99 EE 2C
      :       83 1A BE 3A 2D 7C 89 FF F2 8C CC 6E C8 A2 D5 A5
      :       [ Another 1 bytes skipped ]
1443 3:     INTEGER 65537
      :     }
      :     }
1448 496: [3] {
1452 492:   SEQUENCE {
1456 12:     SEQUENCE {

```

```

1458 3: OBJECT IDENTIFIER
      :
1463 1: basicConstraints (2 5 29 19)
1466 2: BOOLEAN TRUE
1468 0: OCTET STRING, encapsulates {
      : SEQUENCE {}
      : }
1470 14: SEQUENCE {
1472 3: OBJECT IDENTIFIER
      : keyUsage (2 5 29 15)
1477 1: BOOLEAN TRUE
1480 4: OCTET STRING, encapsulates {
1482 2: BIT STRING 4 unused bits
      : '1111'B
      : }
1486 52: SEQUENCE {
1488 3: OBJECT IDENTIFIER
      : extKeyUsage (2 5 29 37)
1493 45: OCTET STRING, encapsulates {
1495 43: SEQUENCE {
1497 8: OBJECT IDENTIFIER
      : serverAuth (1 3 6 1 5 5 7 3 1)
1507 8: OBJECT IDENTIFIER
      : clientAuth (1 3 6 1 5 5 7 3 2)
1517 10: OBJECT IDENTIFIER
      : serverGatedCrypto (1 3 6 1 4 1 311 10
3 3)
1529 9: OBJECT IDENTIFIER
      : serverGatedCrypto (2 16 840 1 113730
4 1)
      : }
      : }
      : }
1540 54: SEQUENCE {
1542 3: OBJECT IDENTIFIER
      : cRLDistributionPoints (2 5 29 31)
1547 47: OCTET STRING, encapsulates {
1549 45: SEQUENCE {
1551 43: SEQUENCE {
1553 41: [0] {
1555 39: [0] {
1557 37: [6]
'http://security.fi.infn.it/CA/crl.crl'
      : }
      : }
      : }
      : }

```

```

:
:
1596 23: SEQUENCE {
1598 3:   OBJECT IDENTIFIER
:       certificatePolicies (2 5 29 32)
1603 16:   OCTET STRING, encapsulates {
1605 14:     SEQUENCE {
1607 12:       SEQUENCE {
1609 10:         OBJECT IDENTIFIER '1 3 6 1 4 1 10403
10 1 4'
:
:
:
:
1621 29: SEQUENCE {
1623 3:   OBJECT IDENTIFIER
:       subjectKeyIdentifier (2 5 29 14)
1628 22:   OCTET STRING, encapsulates {
1630 20:     OCTET STRING
:
:       00 95 79 11 CC B8 9B D2 1D 2A 19 84 C0 36 8B F6
:
:       E8 9E B3 FE
:
:
:
1652 107: SEQUENCE {
1654 3:   OBJECT IDENTIFIER
:       authorityKeyIdentifier (2 5 29 35)
1659 100:   OCTET STRING, encapsulates {
1661 98:     SEQUENCE {
1663 20:       [0]
:
:       CA 11 EF 5D 1D 07 04 98 A9 A5 B5 58 1A 66 4E 0A
:
:       16 2B E0 49
:
:
1685 71:       [1] {
1687 69:         [4] {
1689 67:           SEQUENCE {
1691 11:             SET {
1693 9:               SEQUENCE {
1695 3:                 OBJECT IDENTIFIER
:                   countryName (2 5 4 6)
1700 2:                 PrintableString 'IT'
:
:               }
:             }
1704 13:           SET {
1706 11:             SEQUENCE {
1708 3:               OBJECT IDENTIFIER
:                   organizationName (2 5 4 10)
1713 4:               PrintableString 'INFN'
:
:             }

```

```

      :
      :
1719 37:      SET {
1721 35:      SEQUENCE {
1723 3:      OBJECT IDENTIFIER
      :      commonName (2 5 4 3)
1728 28:      PrintableString 'INFN
Certification Authority'
      :      }
      :      }
      :      }
      :      }
      :      }
1758 1:      [2] 00
      :      }
      :      }
      :      }
1761 62: SEQUENCE {
1763 3:      OBJECT IDENTIFIER
      :      subjectAltName (2 5 29 17)
1768 55: OCTET STRING, encapsulates {
1770 53:      SEQUENCE {
1772 21:      [2] 'datatag6.cnaf.infn.it'
1795 28:      [1] 'valerio.venturi@cnaf.infn.it'
      :      }
      :      }
      :      }
1825 61: SEQUENCE {
1827 3:      OBJECT IDENTIFIER
      :      issuerAltName (2 5 29 18)
1832 54: OCTET STRING, encapsulates {
1834 52:      SEQUENCE {
1836 18:      [1] 'infn-ca@fi.infn.it'
1856 30:      [6] 'http://security.fi.infn.it/CA/'
      :      }
      :      }
      :      }
1888 58: SEQUENCE {
1890 8:      OBJECT IDENTIFIER
      :      authorityInfoAccess (1 3 6 1 5 5 7 1 1)
1900 46: OCTET STRING, encapsulates {
1902 44:      SEQUENCE {
1904 42:      SEQUENCE {
1906 8:      OBJECT IDENTIFIER
      :      caIssuers (1 3 6 1 5 5 7 48 2)
1916 30:      [6] 'http://security.fi.infn.it/CA/'
      :      }
      :      }

```

```

:
:
:
:
:
:
1948 13: SEQUENCE {
1950 9:   OBJECT IDENTIFIER
:       sha1withRSAEncryption (1 2 840 113549 1 1 5)
1961 0:   NULL
:
:
1963 257: BIT STRING
:
:   A8 80 95 4E B7 2F B8 7A 49 67 A7 9C D1 2E E3 23
:
:   AA 26 20 92 B6 F7 35 8E 09 5C 15 40 38 96 95 D1
:
:   BA A8 CF DD 68 49 85 9F E8 99 C5 5E 7B E0 7D 53
:
:   56 14 53 5C 45 C5 ED 2E 0D 05 37 E0 BD EE C6 2D
:
:   B0 98 A6 7A 64 D3 59 9C D9 5D 68 12 FC 2D 72 59
:
:   3F D8 C2 6B D4 FB 48 94 09 11 B6 CF CB B3 4A B2
:
:   C6 4B B0 5C 75 3C 40 5C 28 DC 6B 00 8F A4 1A 79
:
:   CB 04 4D 04 BF 94 16 0F 3F DD 11 9A CF AF 08 E6
:
:   [ Another 128 bytes skipped ]
:
:   }
:
:   }
:
:   }
:
:   }
:
:   }
:
:   }
2224 13: SEQUENCE {
2226 9:   OBJECT IDENTIFIER
:       md5withRSAEncryption (1 2 840 113549 1 1 4)
2237 0:   NULL
:
:
2239 129: BIT STRING
:
:   06 DC 21 A3 01 2D D3 9D B8 23 63 38 BA FD 93 EA
:
:   0A E0 27 D5 F9 70 E2 10 23 67 17 E1 32 0C A7 24
:
:   B3 53 FD 0A B5 C5 26 26 21 17 1C 8B 30 FD 5E 81
:
:   20 53 4B 92 A5 C7 8E 54 79 DC A1 33 00 CF 24 E4
:
:   82 29 B8 4F A9 C2 C1 BA 09 D2 0C 41 FE 0B E1 29
:
:   19 68 51 3B E4 E4 8D A7 20 52 44 84 EE 5E BC 21
:
:   76 24 29 DA DF F6 63 02 B6 A0 BE 1B 20 B5 ED 00
:
:   93 7D 48 C7 01 7B 2A 54 61 7F 9D 56 70 81 D7 63
:
:   }
:
:   }
:
:   }
:
:   }

```

```

2371 14: SEQUENCE {
2373 3:   OBJECT IDENTIFIER keyUsage (2 5 29 15)
2378 1:   BOOLEAN TRUE
2381 4:   OCTET STRING, encapsulates {
2383 2:     BIT STRING 4 unused bits
      :     '1101'B
      :   }
      : }
2387 16: SEQUENCE {
2389 10:  OBJECT IDENTIFIER '1 3 6 1 4 1 8005 100 100 6'
2401 2:  OCTET STRING 30 33
      :  }
2405 29: SEQUENCE {
2407 8:  OBJECT IDENTIFIER '1 3 6 1 5 5 7 1 14'
2417 1:  BOOLEAN TRUE
2420 14: OCTET STRING, encapsulates {
2422 12:  SEQUENCE {
2424 10:  SEQUENCE {
2426 8:  OBJECT IDENTIFIER '1 3 6 1 5 5 7 21 1'
      :  }
      : }
      : }
      : }
      : }
      : }
2436 13: SEQUENCE {
2438 9:  OBJECT IDENTIFIER md5withRSAEncryption (1 2 840 113549 1 1 4)
2449 0:  NULL
      :  }
2451 257: BIT STRING
      : 97 66 74 B6 AB 14 EB EC 49 37 DD E5 F5 A1 97 E2
      : EB AB FD 9C 2A 86 96 29 C3 1F E4 50 30 EC 73 C8
      : 4C 9A 51 80 83 BD 69 10 06 43 DF 09 E7 1F 79 46
      : 11 52 72 D9 07 D2 57 B2 C3 6A 5B F4 B1 31 A8 94
      : D4 6D 3B 7A 70 27 46 4F F0 3F F1 28 D9 C9 A4 4E
      : 39 A7 93 AD 87 62 C0 C9 A6 58 99 55 13 A7 B0 68
      : D9 CE 69 8A 2F C8 C2 A4 2A 44 C1 4E BF 96 D0 39
      : C3 5A 35 42 8A 44 3A 35 C9 E4 3E E2 3A 3A CB 3E
      : [ Another 128 bytes skipped ]
      : }

```

## 6. Security Considerations

This specification defines the elements and use of attributes for authorization services. Implementers of attributes need to be aware that errors in implementation could lead to denial of service or improper granting of service to unauthorized users. Users of attribute assertions should be aware of the situations in which they must require and verify signed assertions.

### Author Information

Valerio Venturi  
INFN – CNAF  
Viale Bertini Pichat, 6/2  
I – 40127 BOLOGNA  
[valerio.venturi@cnaf.infn.it](mailto:valerio.venturi@cnaf.infn.it)

Vincenzo Ciaschini  
INFN - CNAF  
Viale Bertini Pichat, 6/2  
I - 40127 BOLOGNA  
[vincenzo.ciaschini@cnaf.infn.it](mailto:vincenzo.ciaschini@cnaf.infn.it)

### Intellectual Property Statement

The GGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the GGF Secretariat.

The GGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the GGF Executive Director.

### Full Copyright Notice

Copyright (C) Global Grid Forum (2005). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the GGF or other organizations, except as needed for the purpose of developing Grid Recommendations in which case the procedures for copyrights defined in the GGF Document process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the GGF or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE GLOBAL GRID FORUM DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN

WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."

#### **Normative References**

[RFC3280] R. Housley, W. Polk, W. Ford, D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile" RFC3280, Apr 2002

[RFC3281] Farrell, S., Housley, R. "An Internet Attribute Certificate Profile for Authorization", RFC 3281, May 2002.

#### **Informational References**

[VOMS1] "VOMS Architecture v1.1," [http://grid-auth.infn.it/docs/VOMS-v1\\_1.pdf](http://grid-auth.infn.it/docs/VOMS-v1_1.pdf), February 2003.

[OGSI-Authz-Attrs] Thompson, M., Welch, V., Lorch, M., Lepro, R., Chadwick, D., Ciaschini V. "Attributes used in OGSA Authotization", GWD-57.

#### **Appendix A. ChangeLog**

This section to be deleted by the GGF editor prior to publication.

- First version