

25.09.06

Deliverable DJ5.2.3,1: Best Practice Guide - AAI Cookbook - First Edition

Guidelines for Connecting to the eduGAIN AA
Infrastructure



Deliverable DJ5.2.3,1

Contractual Date:	31/05/06
Actual Date:	19/09/06
Contract Number:	511082
Instrument type:	Integrated Infrastructure Initiative (I3)
Activity:	JRA5
Work Item:	5
Nature of Deliverable:	R (Report)
Dissemination Level	PU
Lead Partner	Red.es
Document Code	GN2-06-236v5

Authors: Diego R. Lopez (RedIRIS), José-Manuel Macías (RedIRIS), Maurizio Molina (DANTE), Jürgen Rauschenbach (DFN), Andreas Åkre Solberg (UNINETT), Manuela Stanica (DFN), GN2 JRA5 team

Abstract

This deliverable provides the eduGAIN structure and the basic concepts, and describes how to connect an already existent AAI to eduGAIN.

Table of Contents

0	Executive Summary	iv
1	Introduction	1
	1.1 Notation	1
2	The eduGAIN Basic Concepts	2
3	The eduGAIN Components	4
	3.1 Component Description	4
	3.1.1 Bridging Elements (BE)	4
	3.1.2 Federation Peering Points (FPP)	5
	3.1.3 Metadata Service (MDS)	5
	3.2 Component Identifiers	5
	3.2.1 Examples	6
	3.3 The eduGAIN Naming Registry	7
	3.4 The eduGAIN Schema	7
4	The eduGAIN Protocols and Profiles	8
	4.1 Publishing and retrieving metadata	9
	4.2 eduGAIN Profiles	9
	4.2.1 Web Single Sign On (SSO) Profile	9
	4.2.2 Automated Client Profile	11
5	The eduGAIN Trust Fabric	12
	5.1 PKI Structure	12
	5.2 Certificate Profiles	13
	5.3 Trust Validation Procedures	15
	5.4 TLS Validation	16
	5.5 XML Signature Validation	16
6	The eduGAIN API	17
7	A Checklist for Connecting to eduGAIN	19

8	Conclusions	21
9	References	22
10	Acronyms	24

Table of Figures

Figure 2.1: Using the eduGAIN components and protocols to establish trust links across federation limits	2
Figure 4.1: Schematic overview of an abstract eduGAIN operation	8
Figure 5.1: Structure of the eduGAIN PKI	13

0 Executive Summary

This deliverable summarises the main concepts, protocols and profiles of the GEANT Authorisation Infrastructure (eduGAIN) and explains the trust model used in a nutshell. Equipped with this knowledge, the reader will find the first guideline for the steps that are necessary to connect an already operational Authentication and Authorization Infrastructure (AAI), such as, for example, the SWITCHai in Switzerland, to eduGAIN. This document is not intended as an AAI or eduGAIN primer, but as a guide for AAI administrators willing to participate in eduGAIN. Taking into account the currently established AAIs, which exist in a small number of European NRENs and the development stage of eduGAIN itself, this document can be seen to be provided quite early. However, federated services are moving more and more into the centre of interest in many countries, which makes it important to be informed from the beginning about how such national federation developments can be integrated into the international cooperative environment.

The eduGAIN basic concepts are introduced in the beginning of this document (a more detailed presentation can be found in the “GÉANT2 AAI Architecture” document DJ5.2.2, [GN2DJ522]), followed by a description of the role and functioning of the architecture components and the naming conventions used to designate them. This set of naming conventions contributes, together with the Metadata Service (MDS) and the Public Key Infrastructure (PKI), to ensuring safe and trustworthy communication between the resource owner and the users’ home institution belonging to different local federations. A description of the metadata interactions (publishing and retrieval) necessary in this process is provided further on in the document.

The currently implemented eduGAIN protocols and profiles are also presented so as to provide a better understanding of its functioning in practice. Each profile is defined as the precise exchange of messages and the processing rules for the messages in a particular use case. An essential part of the eduGAIN functionality is its trust model, which needs to be thoroughly understood and applied within each participating federation. Therefore, the different elements constituting the eduGAIN trust fabric are described, including the validation strategies for connections and signatures that must be followed. Finally, the roadmap for connecting to eduGAIN provides an overview of the necessary steps to be taken for joining the confederation.

This document, also named “The eduGAIN Cookbook” to indicate its main purpose, will be updated in a later stage of the project to reflect experiences collected in the pilot phase of eduGAIN and to integrate possible extensions. Note that the eduGAIN confederation policy will be covered in a separate document in a later project phase.

Project:	GN2
Deliverable Number:	DJ5.2.3,1
Date of Issue:	25/09/06
EC Contract No.:	511082
Document Code:	GN2-06-236v5

1 Introduction

eduGAIN is the confederation technology¹ developed by the GÉANT2 project in order to achieve the interconnection of federated Authentication and Authorisation Infrastructures (AAI). This document provides a series of guidelines for connecting a given national AAI to eduGAIN. It is intended to collect and provide, in a concise manner, the information available about the eduGAIN technological procedures and requirements, which are currently scattered throughout the rest of the eduGAIN technical documentation and code.

It is important to note that this version provides only technological guidelines. Organisational issues will be addressed in a separate document, covering the eduGAIN confederation policy.

1.1 Notation

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as described in [RFC 2119]:

... they MUST only be used where it is actually required for interoperation or to limit behaviour which has potential for causing harm (e.g., limiting retransmissions) ...

These keywords are thus capitalized when used to unambiguously specify requirements over protocol and application features and behaviour that affect the interoperability and security of implementations. When these words are not capitalized, they are meant in their natural language sense.

Example code and listings of XML schemas appear like this.

¹ Although not fully internationally standardized, the term “confederation” is more and more used to refer to infrastructures allowing different federated AAIs interoperability. The eduGAIN group has been actively promoting its use as we think it fully reflects the way in which federation interconnection is achieved using similar principles

2 The eduGAIN Basic Concepts

eduGAIN is an authentication and authorisation infrastructure (AAI) based on the **confederation** concept. As a confederation, eduGAIN provides the means to interconnect a set of national or community-wide federated AAls. These participating federations cooperate to provide services to their member organisations and users beyond their limits. The confederation requires that both identity management and authentication/authorisation services are properly handled by the participating federations, as it only provides the means to enable their interoperation.

Since members of a participant federation do not know in advance about members in other federations, a procedure to establish trust among them is required. Trust links are established by means of a common trusted source for metadata, the Metadata Service (MDS), and used by specific eduGAIN components, the Bridging Elements (BEs), that perform the appropriate adaptation between the eduGAIN and the local federation trust environments. Metadata about a certain federation are maintained by its Federation Peering Point, that is the only element that the MDS recognizes as authoritative source for metadata pertaining that federation.

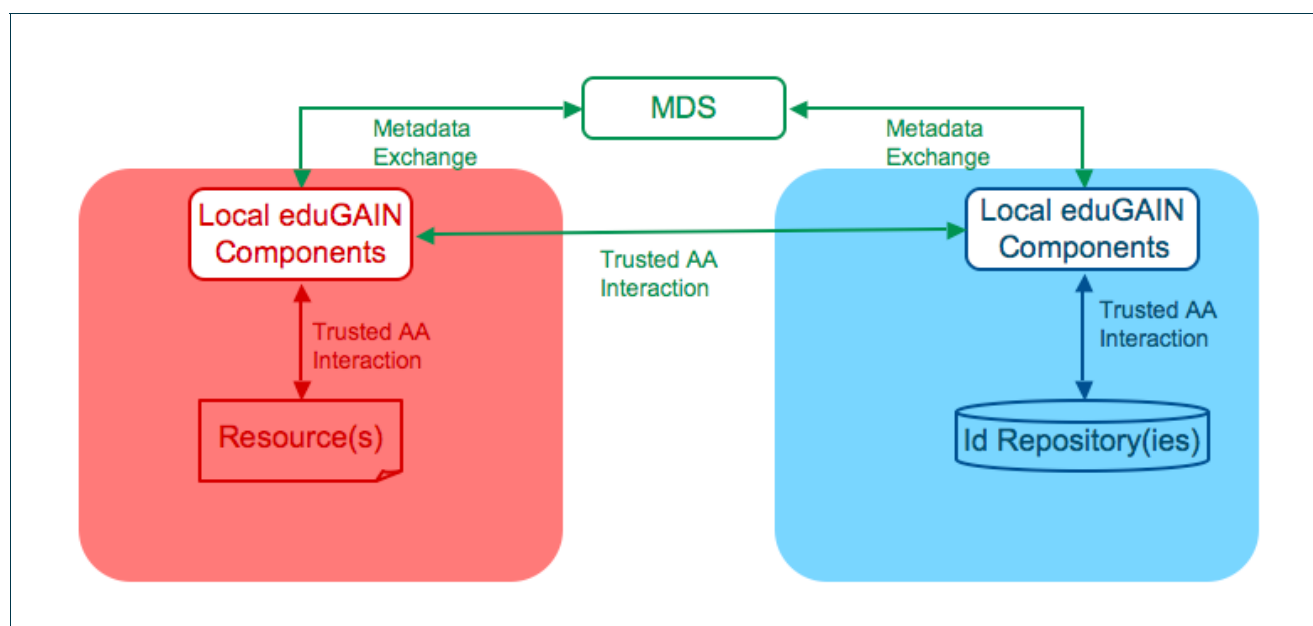


Figure 2.1: Using the eduGAIN components and protocols to establish trust links across federation limits

eduGAIN establishes trust through its **Public Key Infrastructure** (PKI), and a set of **naming conventions** for its components. The relevant information about the eduGAIN components is stored at the MDS and dynamically retrieved and updated via a metadata exchange interface based on the **REST** (Representational State Transfer [REST]) architecture model. Exchange of security information between components is enhanced by the use of the XML-based OASIS standard **SAML** (Security Assertion Markup Language).

Figure 2.1 shows how the eduGAIN components and protocols (in green) are used to establish trust links among resources and identity repositories inside different participating federations, without affecting their local procedures and protocols, shown in different colours for each of the two federations in the diagram.

3 The eduGAIN Components

The eduGAIN architecture consists of three main types of components: **Bridging Elements** (BE), **Federation Peering Points** (FPP) and the **Metadata Service** (MDS). Their interactions contribute to establishing the trust among the participating federations.

3.1 Component Description

3.1.1 Bridging Elements (BE)

BEs are always integrated within a participating federation and serve as a means of establishing appropriate trust links among federation components and user applications, and of adapting syntax, semantics and procedures used by the participating federations. They are the objects (and subjects) of trust when crossing federation limits. The eduGAIN trust will be maintained among these elements (as they are eduGAIN-aware, the MDS will know about them). The internal trust of each BE with respect to its local federation must be established according to the appropriate local procedures. This transforms the problem of maintaining an NxM trust matrix problem into a one-to-one mapping from eduGAIN trust into the corresponding internal federation trust.

There can be one single BE in a federation, acting as a trust aggregator for the other AAI elements in the local federation, or multiple BEs. In the first case we speak of a **Local Federation Adaptor** (LFA), generally corresponding to the case where established infrastructures are in place but the AAI elements are not yet eduGAIN aware. Here a LFA is used to adapt the established infrastructure's own protocols/profiles/procedures to the eduGAIN interfaces and serves as a means of aggregating trust for the whole federation.

In time, as local federation software becomes eduGAIN-aware, it is expected that individual components of the established infrastructure will be allowed to interact directly within eduGAIN by means of BEs called **Local Adaptors** (LA). These are very similar to LFAs except that they don't aggregate trust for the entire federation, and may in fact even interface to a single host.

3.1.2 Federation Peering Points (FPP)

FPPs serves as a means of publishing metadata about a federation through the MDS (see below). For each federation connected to eduGAIN there is exactly one FPP, which SHOULD be dynamically informed of the state and changes of all BEs within its federation. The FPP thus plays the role of a central administration system by means of which each federation can announce its practices via the MDS and keep other participants informed of changes.

3.1.3 Metadata Service (MDS)

The MDS serves as a means of storing and providing metadata about eduGAIN interfaces such as identity providers (IdP), Attribute Authorities (AA), Service Providers (SP) and others. Its main use consists of locating the appropriate identity providers able to identify a certain entity in a given federation.

For this purpose, the FPP or authorized BEs within a federation publish, at the MDS, the relevant metadata related to the available local interfaces. Upon inquiry from other BEs, this metadata can then be dynamically retrieved during the trust establishment process by means of HTTP query/response exchanges taking place via the REST interface between BEs and the MDS.

3.2 Component Identifiers

Since the MDS serves as a means of acting as an authoritative and trusted source of metadata among otherwise mutually unaware federations, a way for uniquely identifying a certain element within the whole eduGAIN fabric is required. Neither identity nor service providers at each participating federation have direct access to the certificates used during peer validation. They need to establish a dynamic trust link through the BE and the trust anchors¹ exchanged via the MDS. The trust validation process is obviously enhanced (both in its processing and in its further auditing) by using identifiers with a formal, well-established format.

Therefore, all components in eduGAIN message elements and assertions MUST be identified according to the following rules:

- Identifiers SHALL be coded by means of URNs in the `urn:geant:edugain:component` namespace.
- Identifiers SHALL establish the kind of component they apply to by means of the following predefined prefixes:
 - `urn:geant:edugain:component:mds` for a Metadata Server.
 - `urn:geant:edugain:component:fpp` for a Federation Peering Point.

¹ The elements where the evaluation of the level of trust on a connecting component is started at. In PKI-based trust schemas, trust anchors are typically the self-signed certificate(s) of the trusted root CA(s).

- o `urn:geant:edugain:component:be` for a Bridging Element.
- o `urn:geant:edugain:component:sp` for a Service Provider.
- o `urn:geant:edugain:component:idp` for an Identity Provider.
- Identifiers SHALL follow the hierarchy of the trust establishing process, including the identifiers of the eduGAIN FPP and the BE it has to run through.
- Identifiers for service providers MUST include a local service identifier, that MAY typically consist of the initial or root URL for the service.

Other naming schemas MAY be considered as acceptable for identifying eduGAIN components, as long as they fulfil the requirements of uniqueness and appropriate registration. These candidate schemas MUST be accepted by the eduGAIN confederation participants prior to their inclusion in the eduGAIN deployment procedures.

3.2.1 Examples

A typical MDS identifier should be like:

```
urn:geant:edugain:component:mds:galaxian
```

A typical FPP identifier should be like:

```
urn:geant:edugain:component:fpp:starfleet
```

A typical BE identifier should be like:

```
urn:geant:edugain:component:be:starfleet:enterprise
```

A typical SP identifier should be like:

```
urn:geant:edugain:component:sp:starfleet:enterprise:captainlog:http://enterprise.starfleet.sf/logs/captain/
```

A typical IdP identifier should be like:

```
urn:geant:edugain:component:idp:starfleet:enterprise:roll
```

3.3 The eduGAIN Naming Registry

The proper management of the identifiers described above, as well as many other elements in the eduGAIN protocols and infrastructure (protocol components, attribute references, well-defined attribute values, etc.) requires the existence of an **eduGAIN Naming Registry**, serving as a means of publishing and maintaining namespace allocations.

This registry SHALL operate the `urn:geant:edugain` namespace, by direct delegation from the `urn:geant` registry. It will contain the branches and final values acceptable inside the namespace, including the corresponding delegations where applicable. In principle, it will consist of a Web interface providing human-readable HTML pages to be used as reference, although using other formats, specifically more oriented towards direct machine access.

The only identifiers that MUST be acceptable to an eduGAIN infrastructure element are those included or directly derived from this registry, which is the only reference for eduGAIN software development and deployment. Should other namespaces become acceptable in the future (as described in the previous section), there SHALL be an explicit reference to them at the eduGAIN Naming Registry.

3.4 The eduGAIN Schema

Attributes exchanged by the eduGAIN components SHALL be in accordance to the SCHAC schema [SCHAC]. SCHAC stands for SCHEMA for ACademia and provides a set of attributes, agreed among the European NRENs, for the exchange of person and institution related information and not yet covered by previously defined schemas, which SCHAC is based upon. Any application using eduGAIN will be able to select the appropriate subset of SCHAC attributes. It is important to note that when talking about “SCHAC attributes” we refer to the whole set of them as defined by SCHAC, i. e., not only those SCHAC-specific attributes (identified by the `schac` prefix in their identifier), but also those defined by the schemas that the SCHAC document assumes are available and properly coded.

Metadata about eduGAIN components SHOULD include the appropriate references to the attributes they may request and/or assert. When doing this, they MUST use the attribute identifiers defined by the corresponding standards and registries.

Project:	GN2
Deliverable Number:	DJ5.2.3,1
Date of Issue:	25/09/06
EC Contract No.:	511082
Document Code:	GN2-06-236v5

4 The eduGAIN Protocols and Profiles

In the purpose of performing authentication and authorisation interactions, the Security Assertion Mark-up Language (SAML), in combination with SOAP transport over a secured channel, is in wide use and already provides large parts of the required functionality. SAML is a set of standards well suited to the eduGAIN tasks. Actual messages to be exchanged by the eduGAIN elements consist essentially of variants of the SAML messages. The SAML data type definitions used in this document correspond to version 1.1 [SAML11], since this version is supported in the federation-aware software packages deployed in higher education and research today. A smaller part of the eduGAIN functionality requires the use of SAML2.0 [SAML20]. As the deployed federation software begins to integrate SAML 2.0, this will also be reflected in its use within eduGAIN.

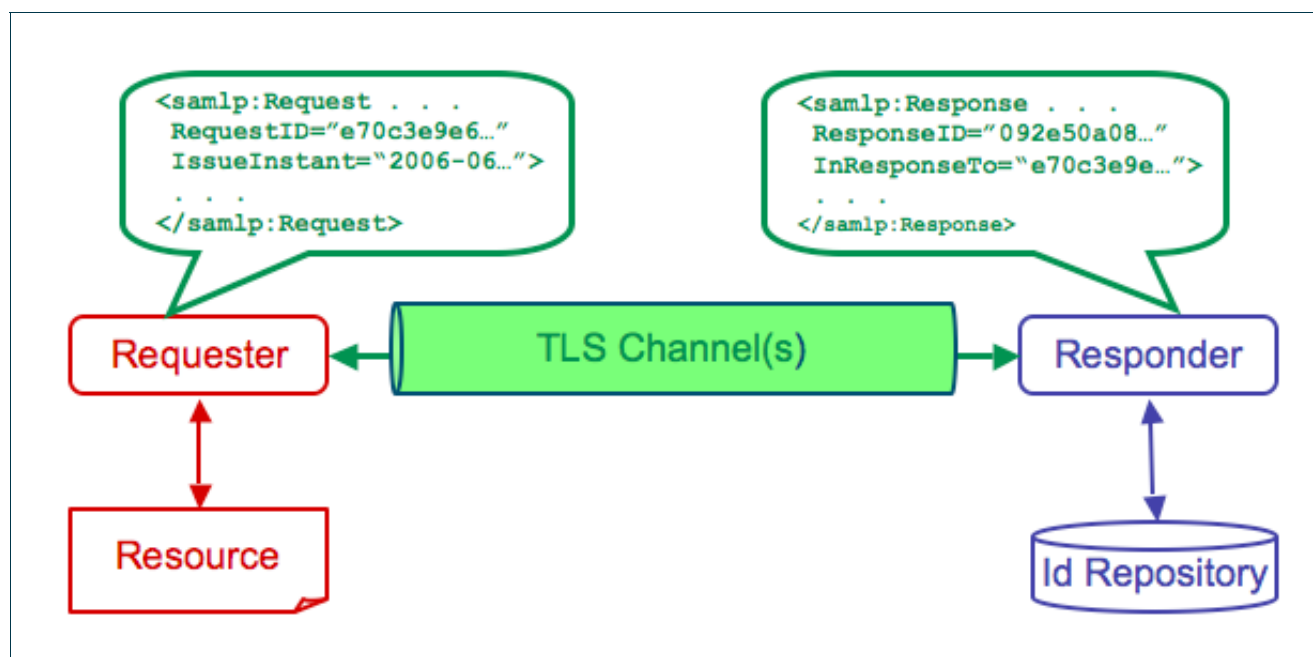


Figure 4.1: Schematic overview of an abstract eduGAIN operation

With the exception of the operations defined for the MDS, eduGAIN abstract messages are mapped into SAML `Response` and `Request` elements, according to the following general rules:

1. All **Requests** are conveyed as SAML `Request` elements.
2. All **Responses** are conveyed as SAML `Response` elements.
3. Error responses are distinguished from other responses by means of their main `StatusCode` element.

4.1 Publishing and retrieving metadata

The MDS deals with the metadata information model as defined in the SAML 2.0 Metadata specification [SAMLMD]. To lookup, search for, and publish metadata eduGAIN uses the REST architectural model based on HTTP exchanges. REST fits the MDS model well and is simple. It also has the benefit of being compatible with other systems using HTTP to retrieve metadata from a location stored in DNS [SAMLMD], and at the same time adding support for sophisticated searching and publishing.

Every federation participating in eduGAIN **MUST** publish via the MDS metadata related to its local interfaces, such as identity providers, attribute authorities, and service providers. These interfaces are generally subordinated to a BE and therefore the published metadata concerns one or more BEs from the local federation, with their associated interface descriptions. Typical metadata **MUST** include the component identifiers and location (contact URLs) of the corresponding elements, and **SHOULD** include additional information such as attributes supported or required by the specified interfaces.

This information is published in the form of SAML 2.0 XML documents having as root either an `EntityDescriptor` element (associated with one BE) or an `EntitiesDescriptor` element (associated to several BEs). Publication can take place from a centralized point within a federation (the FPP) or from BEs that have been authorized to publish their own metadata at the MDS. In the latter case, the root of the SAML 2.0 document is an `EntitiesDescriptor` containing data about one or more BEs from the local federation, while in the first case the root is an `EntityDescriptor` carrying the metadata associated to the publishing BE.

All published metadata are stored in a database at the MDS and can be retrieved upon inquiry from remote BEs during the trust establishment process. An interrogating BE may need for instance to locate the identity provider associated to a user from a different federation and therefore issues a metadata query based on certain information obtained from the user. The MDS response will consist of a SAML 2.0 document containing the required information about the user's home institution, or of an error message in case the search did not return a useful result.

4.2 eduGAIN Profiles

4.2.1 Web Single Sign On (SSO) Profile

A BE (the *remote BE*, or R-BE in what follows) willing to authenticate users belonging to another participating federation through Web SSO **MUST** constrain the users to connect to the corresponding Web interfaces in

order to get their appropriate identity data by means of a HTTP redirection. This redirection MUST be done using the procedures described here. These procedures are essentially equivalent to the Shibboleth Web SSO Browser/POST profile (as described in [SAMLBind] and [ShibArch]), mapping the eduGAIN parameters to HTTP constructs as defined below.

The HTTP request to the user's home BE (H-BE in what follows) interface MUST use the GET method and MUST contain the following URL-encoded query string parameters:

- `providerId` The unique identifier of the requesting R-BE. It MUST be coded according to the structure defined for BE identifiers in the guidelines of section 3.2.
- `shire` The HTTP interface at the R-BE to which to deliver the authentication response as described below.
- `target` A value specified by the R-BE to be returned in the TARGET form element of the authentication response. It SHOULD be opaque to the H-BE, but it MAY be the URL of the resource intended to be accessed at the R-BE side.

The query string MAY contain the following optional parameter:

- `time` The current time, in seconds elapsed since midnight, January 1st, 1970, as a string of up to 10 base10 digits.

4.2.1.1 Processing Rules

The H-BE MUST process the supplied request and either return an error response to the user agent or attempt to fulfil the request according to the local procedures. If an error occurs, the H-BE MUST return a SAML <Response> in accordance with the SAML Browser/POST profile and coded according to the rules described for the SAML mapping of eduGAIN `AuthenticationResponse` with error results.

The `shire` parameter is used as the return interface at which the HTTP POST method will be invoked, including the HTTP form-encoded data containing the response, and is also the value placed in the `Recipient` attribute of the SAML <Response> element. The `target` parameter MUST be used as the value of the TARGET form element whether or not an error has occurred.

The `providerId` parameter MAY be used by the H-BE (and/or the home local federation components) to customize the processing of the request based on their knowledge of a relationship with the service provider. Such customization might include, but is not limited to, the format of the principal's identifier to be returned in the assertion(s), the credential to use while signing the <Response> message, and the set of attributes to include in the authentication assertion, if any.

Finally, the `time` parameter MAY be used as an indicator of the freshness of the request so that replayed requests, such as might be triggered by navigation of a user agent's history list, can be detected. The `time` parameter MUST NOT be used as part of any security measures.

4.2.2 Automated Client Profile

This profile is indicated for the cases of software not directly operated by humans in the moment when it has to engage in an authentication or authorization interaction. This category includes elements such as daemons, autonomous servers, programs subject to automatically scheduled execution, etc.

Each automated client has to hold an X.509 certificate signed by any CA acceptable to the eduGAIN BE in the federation it belongs to (that will be referred as the H-BE). The client SHALL use the certificate to retrieve a valid eduGAIN authentication response from the H-BE, and the H-BE SHALL use the TLS connection to implicitly authenticate the client. Note that the client MUST know in advance the location of the H-BE (stored in the client configuration or by means of a trusted third party out of the scope of this profile).

The H-BE will then validate that the certificate used by the client is a legal certificate to be used for an automated client and that it is acceptable within its realm/federation.

The client SHALL use the authentication response as the identity material to be sent along with any request that requires explicit eduGAIN-based authorisation. The service receiving such a request, prior to taking the appropriate authorisation decision, MAY:

- Accept the received authentication response and base its decision on it, or
- Request a renewal of the authentication response to the H-BE, and/or
- Use the (received or renewed) authentication response data to retrieve further attributes from the H-BE.

5 The eduGAIN Trust Fabric

A trust model is required in order to allow each eduGAIN component to assess the identity of its peer(s) during any interaction. This section describes this trust model, including the validation strategies for connections and signatures that must be followed. The trust establishment process will be enabled by means of using TLS connections for each eduGAIN interaction and including XML-Sig digital signatures for the appropriate protocol elements and assertions.

eduGAIN inter-component trust will be supported by a Public Key Infrastructure (PKI) based on X.509 certificates. It will be rooted at a specific-purpose Certification Authority (CA) created and maintained within the project. This root CA will be referred to as the **eduGAINCA**. The self-signed certificate of the eduGAINCA SHALL be the only mandatory root of trust for any eduGAIN component. Other roots of trust MAY be included by certain components for specific applications of the eduGAIN infrastructure. Validation procedures applied by eduGAIN components MUST support the existence of multiple roots of trust.

5.1 PKI Structure

The eduGAINCA SHALL only issue certificates to other CAs, and these subordinated CAs will in turn be responsible for issuing certificates to the individual components. In general, each subordinate CA will correspond to a participating federation inside the eduGAIN infrastructure and SHALL issue certificates to (and exclusively to) the BEs inside that federation.

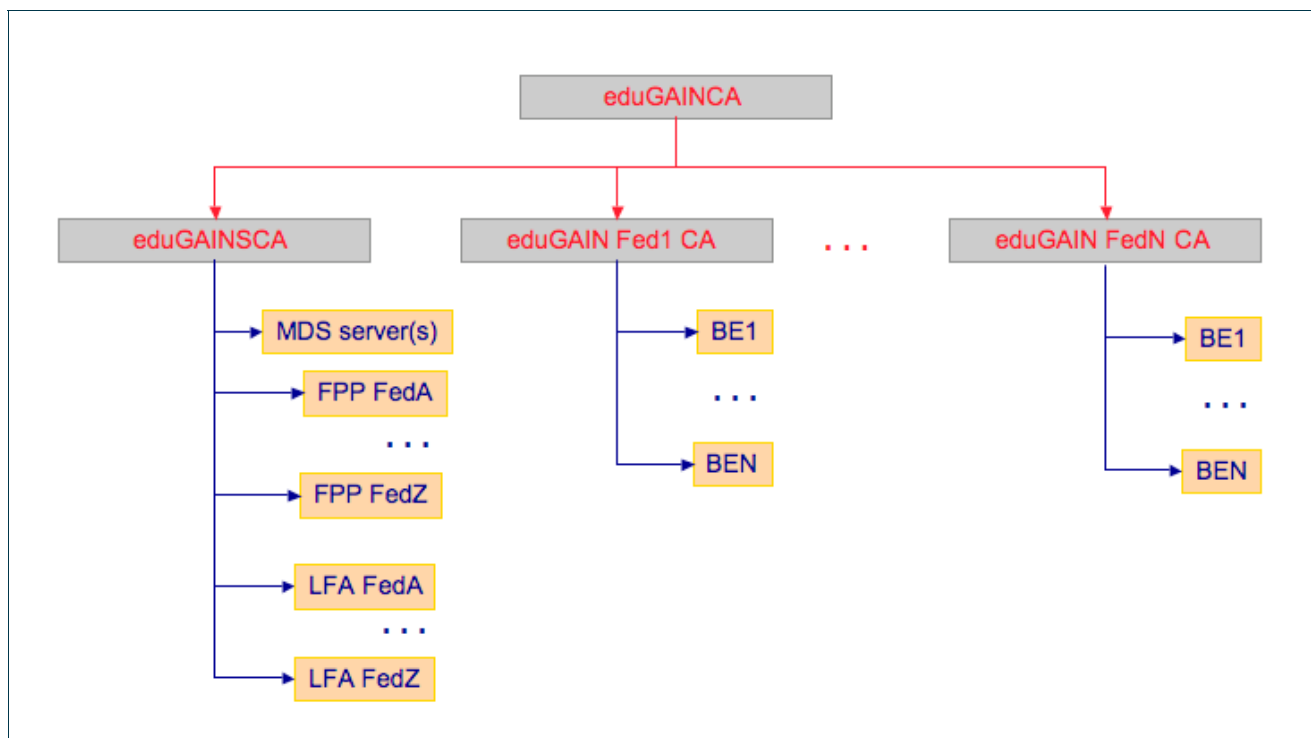


Figure 5.1: Structure of the eduGAIN PKI

As the only exception to this federation-wide CA structure, there will exist a subordinated CA for common eduGAIN components (named **eduGAINSCA**) with the function of:

- Issuing certificate(s) for the MDS server(s).
- Issuing certificates for the FPPs of each participating federation.
- Issuing certificates for the BEs of those federations that are not able or willing to run their own federation eduGAIN subordinate CA.

The eduGAINSCA will provide a set of separately managed Registration Authorities (RA), one for dealing with the two first cases above, and one for each of the participating federations using it.

5.2 Certificate Profiles

All certificates issued by any CA within the eduGAIN PKI MUST conform to the Internet PKI profile (PKIX) for X.509 certificates as defined by RFC 3280 [RFC3280]. These CAs SHALL only issue X.509 v3 certificates.

Certificate serial numbers SHALL NOT consist of sequential numbers according to the time of issuance. It is RECOMMENDED to use pseudo-random generated Universal Unique Identifiers (UUIDs).

The extensions to the X.509 v3 certificate that MUST be present in the certificates issued within the eduGAIN PKI will be:

Project:	GN2
Deliverable Number:	DJ5.2.3,1
Date of Issue:	25/09/06
EC Contract No.:	511082
Document Code:	GN2-06-236v5

- For eduGAIN component certificates:
 - Basic Constraints (critical): `ca: false`
 - Subject Key Identifier: `<hash>`
 - Authority Key Identifier: `<hash>`
 - Key Usage (critical): `digitalSignature, nonRepudiation, KeyEncipherment, dataEncipherment`
 - Extended Key Usage: `serverAuth, clientAuth, emailProtection, codeSigning, timeStamping.`
 - CRL Distribution Points: `<URI>`
 - Certificate Policies: `<OID>`
 - Subject Alternate Name: The appropriate eduGAIN component identifier(s), stored by means of the `uniformResourceIdentifier` field, as defined by RFC 3280.

- For CA certificates:
 - Basic Constraints (critical): `ca: true`
 - Subject Key Identifier: `<hash>`
 - Authority Key Identifier: `<keyID>`
 - Key Usage (critical): `digitalSignature, nonRepudiation, KeyCertSign, cRLSign`
 - Extended Key Usage: `timeStamping`
 - CRL Distribution Points: `<URI>`
 - Certificate Policies: `<OID>`

The OIDs for algorithms used for signatures of certificates issued by the CAs within the eduGAIN PKI MUST be as follows:

- Hash function: `id-sha1 1.3.14.3.2.26`
- Encryption: `rsaEncryption 1.2.840.113549.1.1.1`
- Signature: `sha1WithRSAEncryption 1.2.840.113549.1.1.5`

Each entity MUST have a unique and unambiguous Distinguished Name (DN) in all the certificates issued to the same entity by its correspondent CA. The DN MUST be structured as defined by ITU-T Standards Recommendation X.501.

The eduGAINCA SHALL have the DN:

```
DC=net, DC=geant, CN=eduGAINCA
```

The eduGAINSCA SHALL have the DN:

```
DC=net, DC=geant, CN=eduGAINSCA
```

Other subordinate CAs SHALL have the DN:

```
DC=net, DC=geant, CN=<WellKnownFederationName>
```

Where the value of `WellKnownFederationName` will be derived from the federation identifier within its community. Conflicts in naming SHOULD be mediated by the eduGAINCA operators.

eduGAIN components SHALL have the DN:

```
DC=net, DC=geant, O=<WellKnownFederationName>, CN=<FQDN>
```

Where the value of `FQDN` SHOULD correspond to the FQDN used by the eduGAIN interface(s) of the component. As a component may well provide eduGAIN interfaces through different network interfaces (corresponding to different FQDN), trust evaluation MUST NOT be based on the certificate subject DN.

5.3 Trust Validation Procedures

Trust validation MUST be performed by eduGAIN components according to a two-step procedure:

- The received certificate SHALL be evaluated to check whether the whole trust path correctly resolves to the eduGAIN root of trust.
- The eduGAIN component identifier contained in the Subject Alternate Name extension of the received certificate SHALL be evaluated against the metadata available for this interaction. It MUST match with the component identifier as stored in these metadata.

A failure in any of the verifications above SHALL cause a reject of the requested operation with a `TrustError` result.

This procedure implies that, for a proper trust evaluation, all metadata exchange through the MDS MUST contain the eduGAIN component identifiers applicable in each case.

5.4 TLS Validation

Unless otherwise specified in the corresponding profile, all connections between any two eduGAIN components MUST use TLS and perform two-way certificate validation (both initiator and responder) according to the procedures described in the previous section. Subject DNs of the peer validated certificates (and eduGAIN component identifiers as validated in step 2 above) MUST be included as part of the component logs, and trust paths for the validation SHOULD be included as part of the logs as well.

5.5 XML Signature Validation

XML Signatures MUST be used in the following SAML constructs:

- Assertions containing one SAML `AuthenticationStatement` and (optionally) several SAML `AttributeStatement` in response to an eduGAIN `AuthenticationRequest`.

XML Signatures SHOULD be used in the following SAML constructs:

- Assertions containing SAML `AttributeStatement` in response to an eduGAIN `AttributeRequest`.

Validation of the certificates associated with XML Signatures MUST follow the procedures described in section 5.3. Subject DNs of the issuing party of validated certificates (and eduGAIN component identifiers as validated in step 2) MUST be included as part of the component logs, and trust paths for the validation SHOULD be included as part of the logs as well.

Regarding trust validation, components inside non-SAML-enabled architectures connected through eduGAIN have no other alternative but to trust their BEs and FPPs without any further checking. However, when dealing with SAML-enabled SPs and/or IdPs that use XML signatures instead of (or in addition to) TLS-based trust, it could be possible to use additional checks, allowing end-to-end trust establishment at the price of reducing transparency and (possibly) scalability.

Anyway, since these end-to-end trust checks may be of interest in several use cases, the BEs SHOULD NOT strip the signatures received from the providers connected through them, but rather add their own signature when required.

6 The eduGAIN API

The current implementation of the eduGAIN API is made in Java, and it provides a set of common libraries for all eduGAIN components. The eduGAIN API structure follows a layered approach, as shown below.

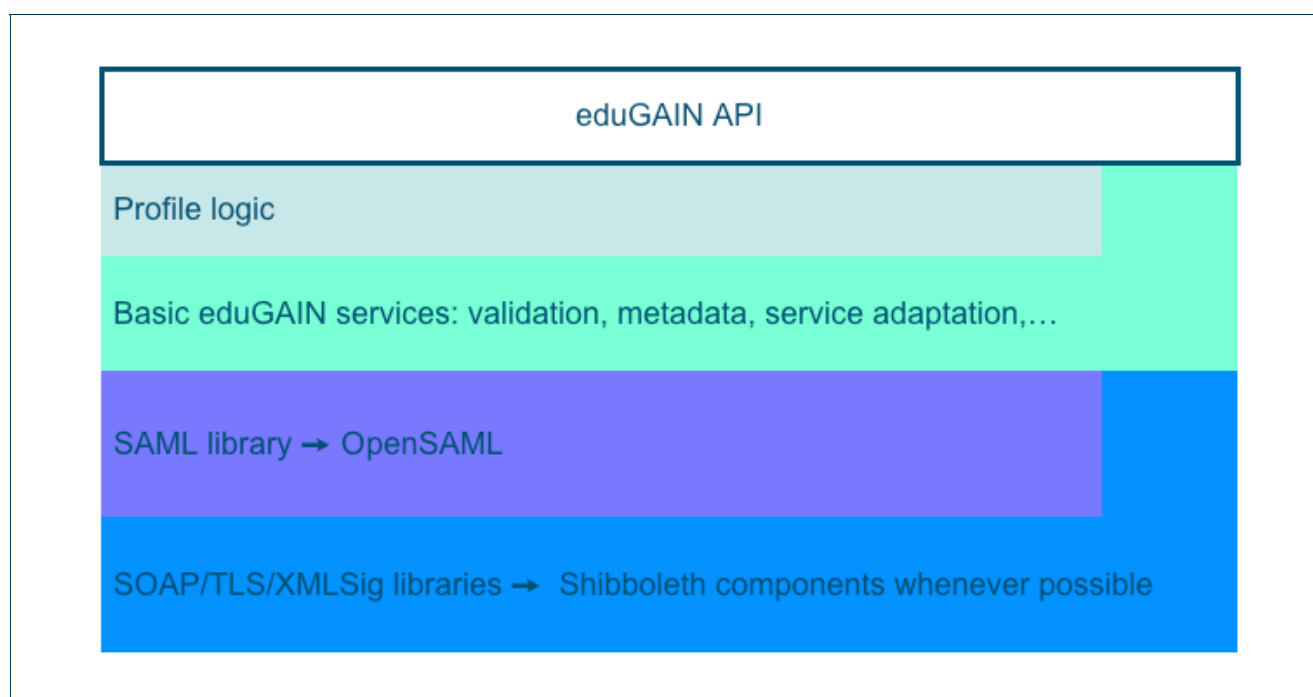


Figure 6.1: Layer diagram for the eduGAIN API

The basic layer includes all the basic support libraries for the different protocols that eduGAIN uses, as required by the current profiles and the binding libraries.

The layer above provides basic binding services for the protocols that implement the eduGAIN service definition. As the currently available binding is based on SAML, the eduGAIN API makes use of [OpenSAML].

The basic eduGAIN services are grouped into three Java packages:

1. **eduGAINVal**, providing the services required to perform the eduGAIN trust validation procedures, and to prepare the exchanged material according to those procedures.
2. **eduGAINMeta**, providing the services required to publish and query eduGAIN metadata.
3. **eduGAINBase**, providing the implementation of the eduGAIN abstract service definition plus specific interfaces for each of the defined profiles.

Profile logic is provided by specific packages inside eduGAINBase, including the *vanilla* eduGAIN profile corresponding to SAML over SOAP over HTTPS, according to the mapping defined by the general eduGAIN architecture specification.

A normal eduGAIN component should only need to interface with eduGAINBase, though any of the above listed services can be accessed if required. The interface is based on the use of specific classes modelling the abstract interface described in [GN2DJ522] and provided by the eduGAINBase package.

The eduGAIN API provides a general abstraction layer for authentication and authorisation operations, that is usable both by components directly woven into the eduGAIN trust fabric and by other elements within their internal trust domains, inside the participating federations or even locally.

7 A Checklist for Connecting to eduGAIN

This section details a set of steps to connect a set of resources and/or identity repositories to the eduGAIN confederated infrastructure. It is not intended to be an exhaustive user guide, but to provide a comprehensive list of the steps that must be accomplished:

- Establish your own identity federation. You must select a federating software (either free or commercial) and establish your internal policy and procedures. eduGAIN is federation agnostic, as long as the appropriate BEs are in place for a given federating solution and procedures does not break the eduGAIN confederation policy. BEs produced along the development of eduGAIN are available for A-Select, PAPI, Shibboleth and the Sun Federation Suite, although many others may exist. If you need more help, the [REFEDS] group is a good starting point.
- Obtain a federation identifier from the eduGAIN Naming Registry. All component identifiers of the elements connecting your federation to eduGAIN must be derived from this identifier according to the guidelines in section 3.
- Entangle your federation in the eduGAIN trust fabric. This can be done either by accrediting your federation CA to be signed by the eduGAINCA, or by applying for certificates through eduGAINSCA. In a first stage, we recommend you start with the latter and consider the possibility of running your own federation CA later on.
- Design the structure of your eduGAIN connection. We recommend starting with a highly centralised schema, with a single FPP and a single BE acting as Local Federation Adaptor as the unique points of contact between your federation and eduGAIN. As you get more experienced, you will be able to decide a more distributed and flexible setup.
- Validate your eduGAIN connection by means of the eduGAIN Validation Facility. This is a set of testing components that will help you in tuning your components to what eduGAIN expects from them. It is mandatory to pass the validation tests before a federation can connect to the production eduGAIN infrastructure.

- Collaborate with the rest of eduGAIN participants in maintaining and enhancing the infrastructure. eduGAIN is a collaborative project that needs your support. Community resources are available at the eduGAIN wiki [JRA5WIKI].
- Be good and may eduGAIN be with you...

8 Conclusions

The “eduGAIN cookbook” provides the essential information about eduGAIN technological procedures and requirements that needs to be understood and applied by the federations wishing to participate. The core concepts forming the base of the eduGAIN architecture are introduced, together with a more detailed presentation of its components and their corresponding roles. The interactions between these components in order to ensure safe and trustworthy communication between local federations are explained within the framework of particular use cases, forming the eduGAIN profiles and protocols.

Special attention needs to be directed towards the eduGAIN trust fabric, as its correct implementation by all participants is an essential condition in the good functioning of the confederation infrastructure. Together with this information, the document aims to cover the necessary steps to be followed by local federations in order to successfully participate in eduGAIN.

This is an early version of the cookbook and, as practical experience is still not available, intended to help early users. It will be refined, extended and corrected in a later stage of the project as version 2 of DJ5.2.3.

9 References

- [GN2DJ522]** D. Lopez, R. Castro, B. Kerver, T. Lenggenhager, I. Melve, M. Milinovic, J. Rauschenbach, K. Wierenga, S. Winter, H. Ziemek et al. GÉANT2 Authentication and Authorisation Infrastructure (AAI) Architecture. GÉANT2 Deliverable DJ5.2.2. October 2005.
<http://www.geant2.net/upload/pdf/GN2-05-192v6.pdf>
- [JRA5Wiki]** Collaboration site for the GÉANT2 JRA5 participants, <http://www.rediris.es/jra5wiki/>
- [OpenSAML]** OpenSAML 1.1/2.0 - an Open Source Security Assertion Markup Language implementation.
<http://www.opensaml.org/>
- [REFEDS]** REFEDS: Research and Education Federations. Group under the auspices of the TERENA Technical Programme on Middleware. <http://www.terena.nl/activities/refeds/>
- [REST]** Representational State Transfer,
http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm
- [RFC2119]** S. Bradner. Key words for use in RFCs to Indicate Requirement Levels. Internet Best Current Practice, IETF. March 1997.
- [RFC3280]** R. Housley et al., Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, IETF, April 2002.
- [SAML11]** E. Maler et al. Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V1.1. OASIS Standard, September 2003.
<http://www.oasis-open.org/committees/download.php/3406/oasis-sstc-saml-core-1.1.pdf>
- [SAML20]** S. Cantor et al. Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS Standard, March 2005.
<http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf>

- [SAMLBind]** E. Maler et al. Bindings and Profiles for the OASIS Security Assertion Markup Language (SAML). OASIS Standard, September 2003.
<http://www.oasis-open.org/committees/download.php/3405/oasis-sstc-saml-bindings-profiles-1.1.pdf>
- [SAMLMD]** S. Cantor (editor). Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS Standard, March 2005.
<http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf>
- [SCHAC]** J. Masa (editor). SCHAC Attribute Definitions for Individual Data, May 2006.
<http://www.terena.nl/activities/tf-emc2/docs/schac/schac-schema-IAD-rel1.pdf>
- [ShibArch]** S. Cantor (editor). Shibboleth Architecture. Protocols and Profiles.10 September 2005.
<http://shibboleth.internet2.edu/docs/draft-mace-shibboleth-arch-protocols-200509.pdf>

10 Acronyms

AA	Attribute Authority
AAI	Authentication and Authorisation Infrastructure
API	Application Programming Interface
BE	Bridging Element
CA	Certificate Authority
DN	Distinguished Name
FPP	Federation Peering Point
FQDN	Fully Qualified Domain Name
IdP	Identity Provider
LA	Local Adapter
LFA	Local Federation Adapter
MDS	Metadata Service
PKI	Public Key Infrastructure
REST	Representational State Transfer
SAML	Security Assertion Mark-up Language
SCHAC	Schema for ACademia
SOAP	Simple Object Access Protocol
SP	Service Provider
TLS	Transport Layer Security
URL	Unified Resource Locator
URN	Unified Resource Names
XML	eXtensible Mark-up Language