

24.04.07

Deliverable DJ5.2.2,2: GÉANT2 Authorisation and Authentication Infrastructure (AAI) Architecture – second edition



Deliverable DJ5.2.2,2

| | |
|------------------------|---|
| Contractual Date: | 31/12/06 |
| Actual Date: | 24/04/07 |
| Contract Number: | 511082 |
| Instrument type: | Integrated Infrastructure Initiative (I3) |
| Activity: | JRA5 |
| Work Item: | WI2 |
| Nature of Deliverable: | R (Report) |
| Dissemination Level | PU (Public) |
| Lead Partner | RedIRIS |
| Document Code | GN2-07-024 |

Authors: D. R. Lopez (RedIRIS, main editor), R. Castro (RedIRIS), B. Kerver (SURFnet), T. Lenggenhager (SWITCH), M. Linden (CSC), I. Melve (UNINETT), M. Milinovic (Srce/CARNet), J. Rauschenbach (DFN), M. Stanica (DFN), K. Wierenga (SURFnet), S. Winter (RESTENA), H. Ziemek (DFN)
With contributions from M. Molina (Dante, JRA1) and other GN2 activities

Abstract

The objective of this deliverable is to define the architecture of eduGAIN, the GÉANT2 Authorisation and Authentication Infrastructure, its internal and external interfaces, and the structure of its components.

Table of Contents

| | | |
|---|---|----|
| 0 | Executive Summary | iv |
| 1 | Introduction | 1 |
| 2 | Architectural Overview | 2 |
| | 2.1 Components | 2 |
| | 2.2 Policy and Trust | 6 |
| | 2.3 Interactions | 7 |
| 3 | eduGAIN Operational Definition | 9 |
| | 3.1 Connection Scenarios | 9 |
| | 3.2 Operational Definition | 15 |
| 4 | Protocol and Bindings | 27 |
| | 4.1 Authentication Protocol | 28 |
| | 4.2 Attribute Exchange Protocol | 30 |
| | 4.3 Authorisation Protocol | 32 |
| | 4.4 Metadata Protocol | 33 |
| 5 | Generic Use Case | 34 |
| 6 | Security and Privacy Considerations | 38 |
| | 6.1 General Considerations | 38 |
| | 6.2 Bridging Elements play a special Role | 38 |
| | 6.3 Credentials and Third Parties | 40 |
| | 6.4 Attribute Release | 40 |
| 7 | Conclusions | 41 |
| 8 | References | 42 |
| 9 | Acronyms | 44 |

Table of Figures

| | |
|--|----|
| Figure 2-1: Basic components of eduGAIN and their relationships | 3 |
| Figure 2-2: Local Federation Adaptor mediating between an established infrastructure and eduGAIN | 4 |
| Figure 2-3: Local Adaptor | 5 |
| Figure 2-4: Trust links among the different eduGAIN components | 7 |
| Figure 3-1: Configuration and time-flow diagrams for LFA usage at both domains | 11 |
| Figure 3-2: Configuration and time-flow diagrams for LFA usage at Home and LAs at Remote | 12 |
| Figure 3-3: Configuration and time-flow diagrams for LAs at Home and LFA usage at Remote | 13 |
| Figure 3-4: Configuration and time-flow diagrams for LAs at both domains | 14 |
| Figure 4-1: Schematic overview of an abstract AA operation | 27 |
| Figure 4-2: Authentication functional components | 29 |
| Figure 4-3: Generalised attribute exchange | 30 |
| Figure 5-1: The generic eduGAIN use case | 36 |

0 Executive Summary

The architecture design document DJ5.2.2 describes the eduGAIN (this name was created for the GÉANT Authentication and Authorisation Infrastructure too be provided by JRA5) main components, its interworking and the operations defined. The eduGAIN operations set the basis for interconnecting European academic users with ubiquitous networked services. A formal specification of the operations and the parameters is appended providing a basis for the implementation of the prototype version. This is a more fine-grained description compared to the initial plan for the document, but of great value, not only as a discussion of the architecture but also as an intermediate step to an installation and the provision of test software.

The basic components of eduGAIN (**G**ÉANT **A**uthorisation **I**nfrastructure for the research and **e**ducation community) are described in the chapter 2 “Architectural Overview”. The design follows the requirements defined in DJ5.2.1, serving as a federator of already established federation based AAls (the local federations).

The eduGAIN architecture ensures safe and trustworthy communication between the resource owner (remote domain) and the users home institution (home domain) belonging to different local federations. This is achieved through the Metadata Service (MDS), the Private Key Infrastructure (PKI) and a set of naming conventions for its architecture components. The interface to the domains is supported by the Bridging Elements (BE). The Bridging Elements map the eduGAIN protocols and profiles to those of the local federations. Although the BE may be a centralised component in the local federation, it is however expected that federation software will be adapted to become ‘eduGAIN aware’ by adding BE functionality, thus enabling direct communication between AAI elements belonging to different local federations.

A basic set of interactions is defined, specified and outlined in the possible scenarios. The deliverable provides the full picture of the interactions taking place in any possible configuration, presented in a component based figure and a time-flow chart for each possible combination. The eduGAIN services are presented including all parameters foreseen in the AAI related communication as an overview, and additionally explaining the mapping to a formal specification language and profiles.

| | |
|---------------------|------------|
| Project: | GN2 |
| Deliverable Number: | DJ5.2.2bis |
| Date of Issue: | 24/04/07 |
| EC Contract No.: | 511082 |
| Document Code: | GN2-07-024 |

1 Introduction

This document presents the results of the design process of the architecture for **eduGAIN**¹, the GÉANT2 AAI, started with first ideas on an architecture blueprint November 2004 in the JRA5 meeting in Amsterdam. The design of an inter-federation architecture is clearly a research area, and therefore terms can be changed to better reflect the intended function. As a result of that the terminology is not always in line with the one used in the requirements document (DJ5.2.1, *Documentation on AAI Requirements*). An updated version of the DJ5.1.1 (*JRA5 Glossary of Terms*) was provided as [DJ5.1.1,2] reflected the current state.

The basic idea behind the design is to draft the architecture in a way that interoperability with existing AAI solutions is strongly supported. The AAI federation-aware software to which eduGAIN will provide interfaces are at least Shibboleth (Internet2), PAPI (RedIRIS), Liberty Alliance/FEIDE (UNINETT) and A-Select (SURFnet). Designing and implementing a full AAI is a non-trivial operation that typically involves many man-years of effort. Given the fact that a number of possible federation-aware software packages are available, eduGAIN assumes the existence of an established local federation (see reference section for federation-aware software). Therefore, solutions for identity management are outside the scope of this design and must be provided by local means.

¹ The group to give the infrastructure a proper single name has coined this term. It is composed of the “edu” prefix (commonly associated to academic and research environments in the Internet worldwide) and the GAIN acronym, standing for *GÉANT Authorisation Infrastructure*.

2 Architectural Overview

The purpose of this chapter is to provide a general sketch of the whole architecture of the eduGAIN infrastructure, describing its components and the basic interactions among them. This architecture tries to satisfy the requirements expressed both by the explicit statements made in the requirement document and the requests coming from the potential user JRAs and SAs.

More specifically, the eduGAIN architecture intends to offer full authentication and authorisation services by acting as a connector of pre-existing AA infrastructures either at a national or international level. Along this document we will refer to them as *established infrastructures*. Thus eduGAIN will provide the superstructure for federating the established infrastructures.

The bridging elements between the local and the eduGAIN infrastructures may be:

- Located at a central point in the local infrastructure, thus aggregating the trust, or
- Provided at the institutional or group level.

In more specific terms, eduGAIN will establish the trust links that enable the infrastructure components to directly interact and provide the interfaces to route and translate all their interactions

2.1 Components

To define the basic components of the architecture let us assume the following general scenario: an attempt of using a resource is made by a user, the resource will request an authorisation decision using the attributes of the user, as provided by an identity repository to which the user will have to be (or has previously been) identified. In the general case, the resource and the identity repository are located in different (security, administrative) domains that we will call respectively (according to the user's view) *remote* and *home*.

| | |
|---------------------|------------|
| Project: | GN2 |
| Deliverable Number: | DJ5.2.2bis |
| Date of Issue: | 24/04/07 |
| EC Contract No.: | 511082 |
| Document Code: | GN2-07-024 |

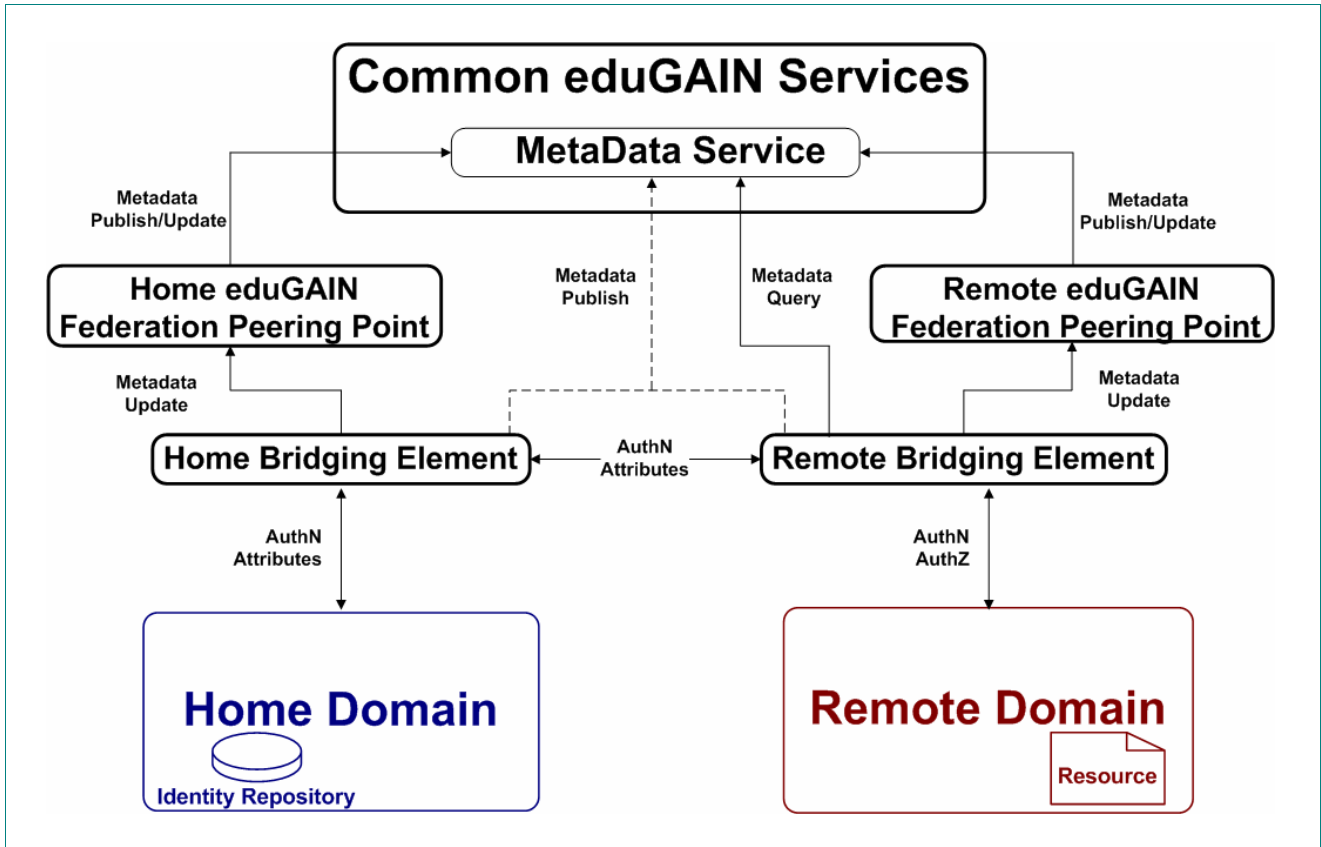


Figure 2-1: Basic components of eduGAIN and their relationships

The proposed architecture is based on a set of components, which use the trust links established among them by the eduGAIN federation schema. These components, as depicted in the above diagram, are:

The **eduGAIN Federation Peering Points**¹ (FPP, available at both domains), in charge of publishing metadata about a federation through the MDS (see below). For each federation connected to eduGAIN there is exactly one FPP, which is dynamically informed of the state and changes of all BEs within its federation. The FPP plays thus 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 potential changes.

The **Common eduGAIN services**, offering general services requested by (and only by) the federations' Bridging Element(s) to accomplish their tasks. Each of these services offers a corresponding interface to the Bridging Elements according to the protocols and profiles defined by the eduGAIN architecture. This version of the architecture only defines one of these services: the **Metadata Service**, in charge of providing metadata

¹ Might be renamed into Federation Publishing Points in later documents

| | |
|---------------------|------------|
| Project: | GN2 |
| Deliverable Number: | DJ5.2.2bis |
| Date of Issue: | 24/04/07 |
| EC Contract No.: | 511082 |
| Document Code: | GN2-07-024 |

about eduGAIN interfaces, and mainly used locating the appropriate identity repository at the home domain, through the *MDSInterface*. Other services, like a certificate verification service, a notary service or a common logging service, can be added in the future.

The **Bridging Elements** (BE), in charge of establishing appropriate trust links among AAI components and user applications and to adapt syntax, semantics and procedures used by established infrastructures and individual sites. BEs are the objects (and subjects) of trust when crossing federation or site limits. The eduGAIN trust will be maintained among these elements (as they are eduGAIN-aware, the MDS and other possible common services will know about them and will be able to assess on them). The internal trust of each BE with respect to its local federation/elements 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 local 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), in the latter case of a Local Adaptor (LA).

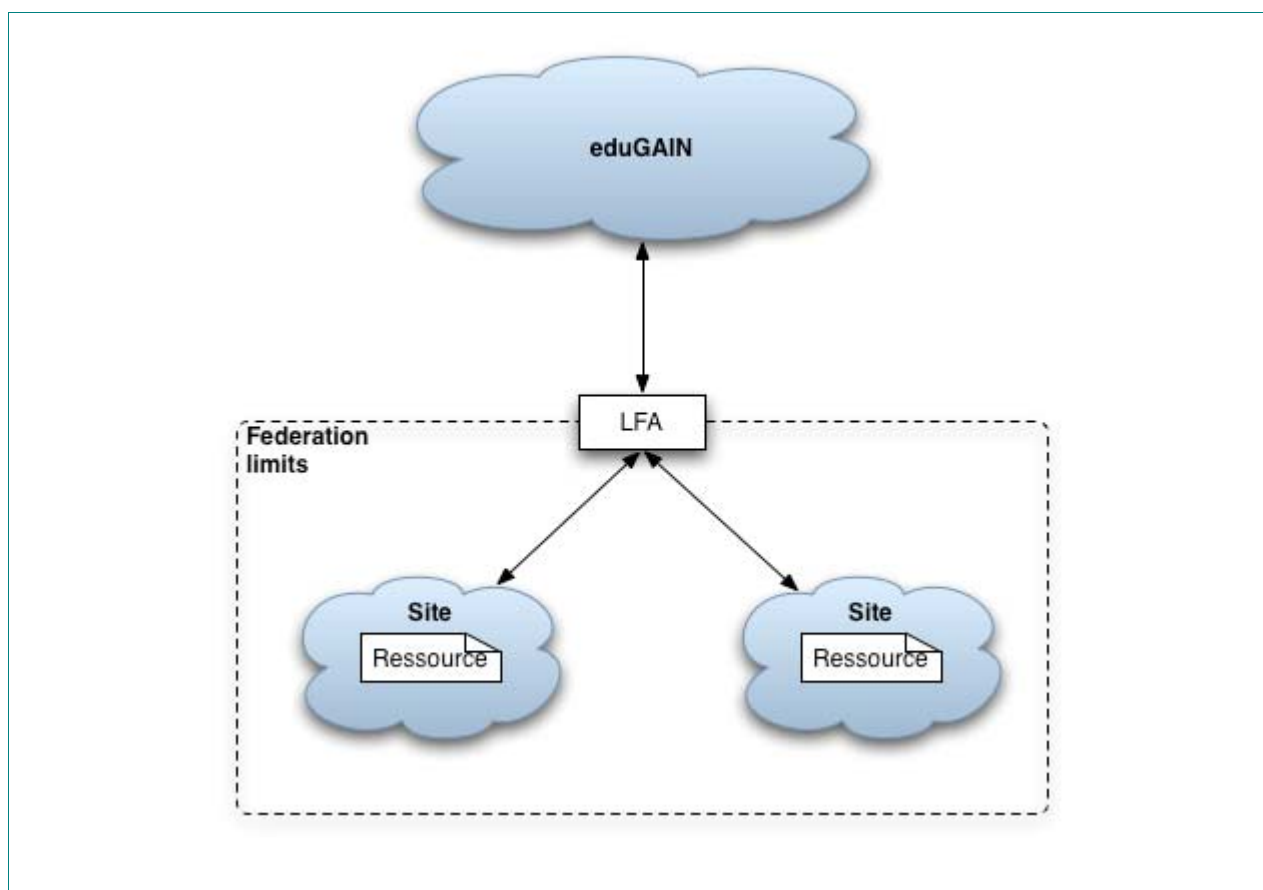


Figure 2-2: Local Federation Adaptor mediating between an established infrastructure and eduGAIN

| | |
|---------------------|------------|
| Project: | GN2 |
| Deliverable Number: | DJ5.2.2bis |
| Date of Issue: | 24/04/07 |
| EC Contract No.: | 511082 |
| Document Code: | GN2-07-024 |

When established infrastructures are in place but the AAI elements are not eduGAIN aware, a **Local Federation Adaptor** (LFA¹) is used to adapt the established infrastructure own protocols/profiles/procedures to the eduGAIN counterparts. Although it is not required by the architecture, one LFA instance (we refer here to a single architectural instance: replication because of dependability reasons is out of the scope of this document) per established infrastructure seems to be the more feasible approach, offering a single interaction point to eduGAIN, while the internal procedures of the national or group-wide infrastructure are preserved. Individual federation components can continue using their procedures and common federation services, as in the case of the WAYF (roughly equivalent to a basic use of the Metadata Service proposed here) used in the Shibboleth federation software.

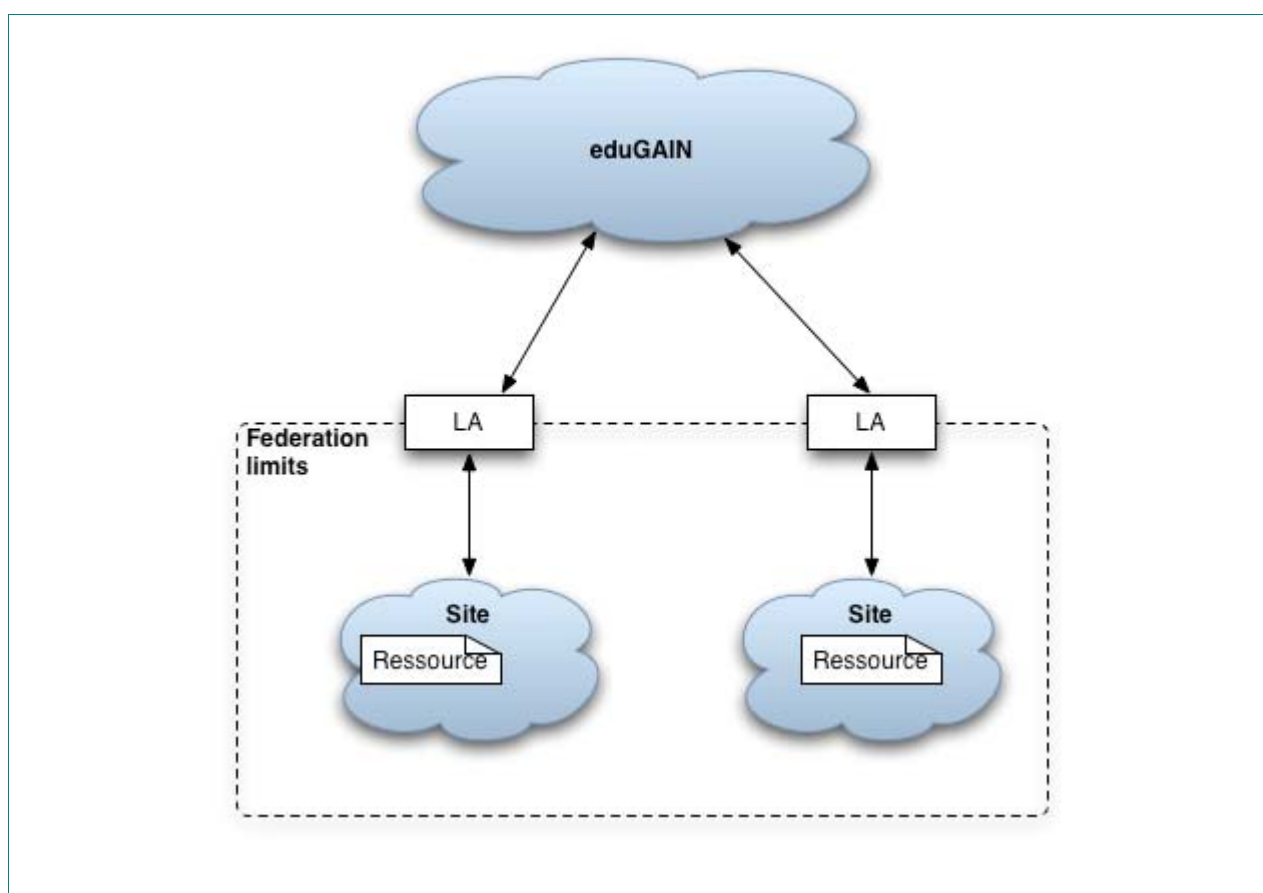


Figure 2-3: Local Adaptor

The LFA will be in charge of adapting syntaxes (and possibly attribute semantics) and of preserving state when required by the established infrastructure (for example, in the case of HTTP redirects, as commonly used in Web-based infrastructures). Therefore the LFA (and the LA) is very much related to the local AAI. However,

¹ This component was originally referred to as *Local Federation Connector* in the previous deliverable on AAI requirements. However, a more detailed analysis of the required features of this component has made us to change its name, with the objective of better describing its functionality.

| | |
|---------------------|------------|
| Project: | GN2 |
| Deliverable Number: | DJ5.2.2bis |
| Date of Issue: | 24/04/07 |
| EC Contract No.: | 511082 |
| Document Code: | GN2-07-024 |

since it is imperative that a local federation is in place JRA5 will in any case deliver the necessary functionality to interface with the federation-aware software that is used in the participating NRENs: A-Select, FEIDE, PAPI and Shibboleth.

It is to be expected that in the beginning, federations will have a single connection point with eduGAIN. It is however reasonable to expect that in due time the local federation software will be adopted to use the protocols developed within this project. It makes sense to allow these eduGAIN-aware components to directly interact with it. The single connection point approach may introduce single points of failure as well as traffic bottlenecks. The model should therefore allow eduGAIN-aware elements of two different local federations to directly interact with each other, using eduGAIN just to establish the trust between the two and to locate each other. If individual components of the established infrastructure are allowed to directly interact within eduGAIN, a **Local Adaptor** (LA) will be used to perform these interactions. The LA is very similar to the LFA, but doesn't aggregate trust for the whole federation, and may in fact even interface to a single host.

2.2 Policy and Trust

Federations, home domains, individual users, remote domains and service providers may each have individual policies that need to interoperate for confederated AAI to work properly. One example is attribute release policies, where more experience must be gained in testing.

Trust establishment may differ from the information flow across eduGAIN components. Existing building blocks in deployed AAI federations and deployed PKI infrastructure are used for establishing trust. The eduGAIN trust fabric is specified in more detail in the AAI cookbook [DJ5.2.3]. In some cases, trust fabric establishment depends on the service, as in the case of wireless access and RADIUS usage.

The implementation of policies is outside of the scope of this document.

| | |
|---------------------|------------|
| Project: | GN2 |
| Deliverable Number: | DJ5.2.2bis |
| Date of Issue: | 24/04/07 |
| EC Contract No.: | 511082 |
| Document Code: | GN2-07-024 |

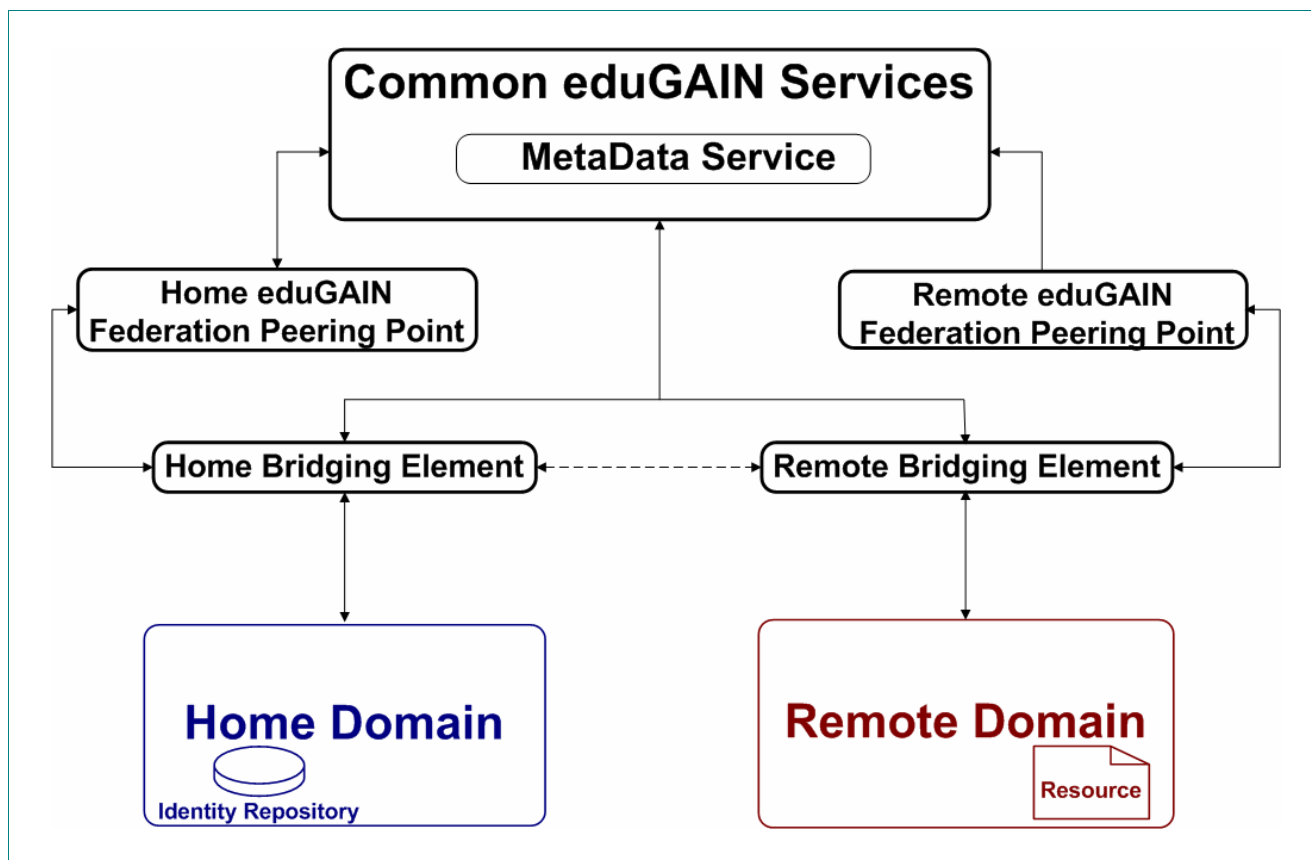


Figure 2-4: Trust links among the different eduGAIN components

2.3 Interactions

The interactions depicted in this architectural overview are intended to describe a high-level operational view of the required data exchanges. They should not be interpreted as individual connections and/or operations. Each of the interactions described here is refined in terms of formal operations further in this architectural specification. The considered interactions are:

- Authentication of the user (in the Home Domain), initiated by Service Provider (in the Remote Domain)
- Attribute request, initiated by Service Provider for information about the user
- Authorisation for the service, based on information related to the user.
- The Metadata Service is used by Federation Peering Points to locate the appropriate interfaces where the above interactions can be performed and establish trust between them.

The requests and responses exchanged over the eduGAIN infrastructure are:

AuthNReq: A request for authentication data about the subject requesting a service. It may imply a redirection to the home domain in order to perform a direct exchange of credentials.

AuthNResp: A response containing authentication data according to a previously issued *AuthNReq*. It must contain an anonymised handle to allow further attribute requests, and may contain any other attributes about the entity that originated the request, according to service and privacy policies.

MetadataLookup: A request for metadata concerning the BE where a particular AA request can be satisfied. It must contain the concerned BE's identifier (name), and that of the federation to which it belongs.

MetadataSearch: A request for metadata in order to determine the BE where a particular AA request can be satisfied. It must include whatever information needed to simplify the discovery process of the home domain of the subject, in the form of specific attribute-value pairs called *Home Locators*. It may require direct user interaction.

MetadataPublish: A HTTP POST request in order to publish at the MDS metadata about one or more BEs.

MetadataResponse: A response containing the requested metadata about an eduGAIN component, or an error message in case no matches were found.

AttrReq: A request for attributes pertaining to a certain subject. It must contain the handle corresponding to the subject, a list of attribute identifiers requested, and a resource/service identifier.

AttrResp: A response containing attributes according to a previously issued *AttrReq*. It must contain a list of attribute/value(s) pairs and/or reasons for not disclosing certain values or attributes, according to service and privacy policies.

AuthZReq: A request for an authorisation decision. It must contain the resource identifier and the attributes collected for the requesting entity, and may include references to the policies to be applied.

AuthZResp: A response containing an authorisation decision according to a previously issued *AuthZReq*. It must contain a Boolean value specifying the decision and may include data to be used by the resource for whatever purpose (personalisation, for example) and/or an explanation about the returned decision, according to service and privacy policies.

| | |
|---------------------|------------|
| Project: | GN2 |
| Deliverable Number: | DJ5.2.2bis |
| Date of Issue: | 24/04/07 |
| EC Contract No.: | 511082 |
| Document Code: | GN2-07-024 |

3 eduGAIN Operational Definition

This chapter explores the scenarios that arise from the interactions among the components described in the previous section and introduces the operational specification for eduGAIN, following a Service Oriented Architecture (SOA) approach: The services offered (internal or externally) by eduGAIN are listed and, for each service, a list of the available operations and their parameters is included.

3.1 Connection Scenarios

This section presents several diagrams to illustrate how eduGAIN components interact, according to the definitions made in 2.3. A pair of diagrams (architectural and time-flow) is included for each of the configurations that can arise from combining (either at the home or the remote domains) the different BEs described in 2.1. The interactions in the diagrams are ordered according to the following general sequence of events:

The user's attempt to use a resource first triggers actions that result in the user's authentication. Further actions can be attribute retrieval and authorisation decision (if requested by the resource), possibly requiring further attribute retrieval. *The minimum scenario that can occur is that a resource only uses the authentication service for a user and allows or denies resource usage simply based on the fact whether or not the user can authenticate (the eduroam network admission decision is such a scenario).* Other scenarios might include attribute retrieval, after the authentication, but no authorisation afterwards (if the mere presence of an attribute is already sufficient for the resource). In most scenarios however, all three phases take place.

The user's authentication and attribute retrieval require accessing an Identity Repository, located in the user's home domain, which can provide this data. The authorisation decision is to be made by an Authorisation Engine.

The diagrams follow a colour code in which:

| | |
|---------------------|------------|
| Project: | GN2 |
| Deliverable Number: | DJ5.2.2bis |
| Date of Issue: | 24/04/07 |
| EC Contract No.: | 511082 |
| Document Code: | GN2-07-024 |

- Red is used to show the remote domain with its BEs and internal elements, and the interactions using its own protocols/procedures.
- Blue is used to show the home domain with its BEs and internal elements, and the interactions using its own protocols/procedures.
- Green is used to show the eduGAIN components, interfaces, services, and interactions.

Let us first consider the case when established infrastructures exist at the home as well as at the remote domain but the federation software is not eduGAIN-aware. This means that at both domains LFAs are used. After that the two cases in which only one of the domains use eduGAIN-aware federation software are presented, and finally, the case in which both domains use eduGAIN-aware federation software.

3.1.1 LFA usage at both domains

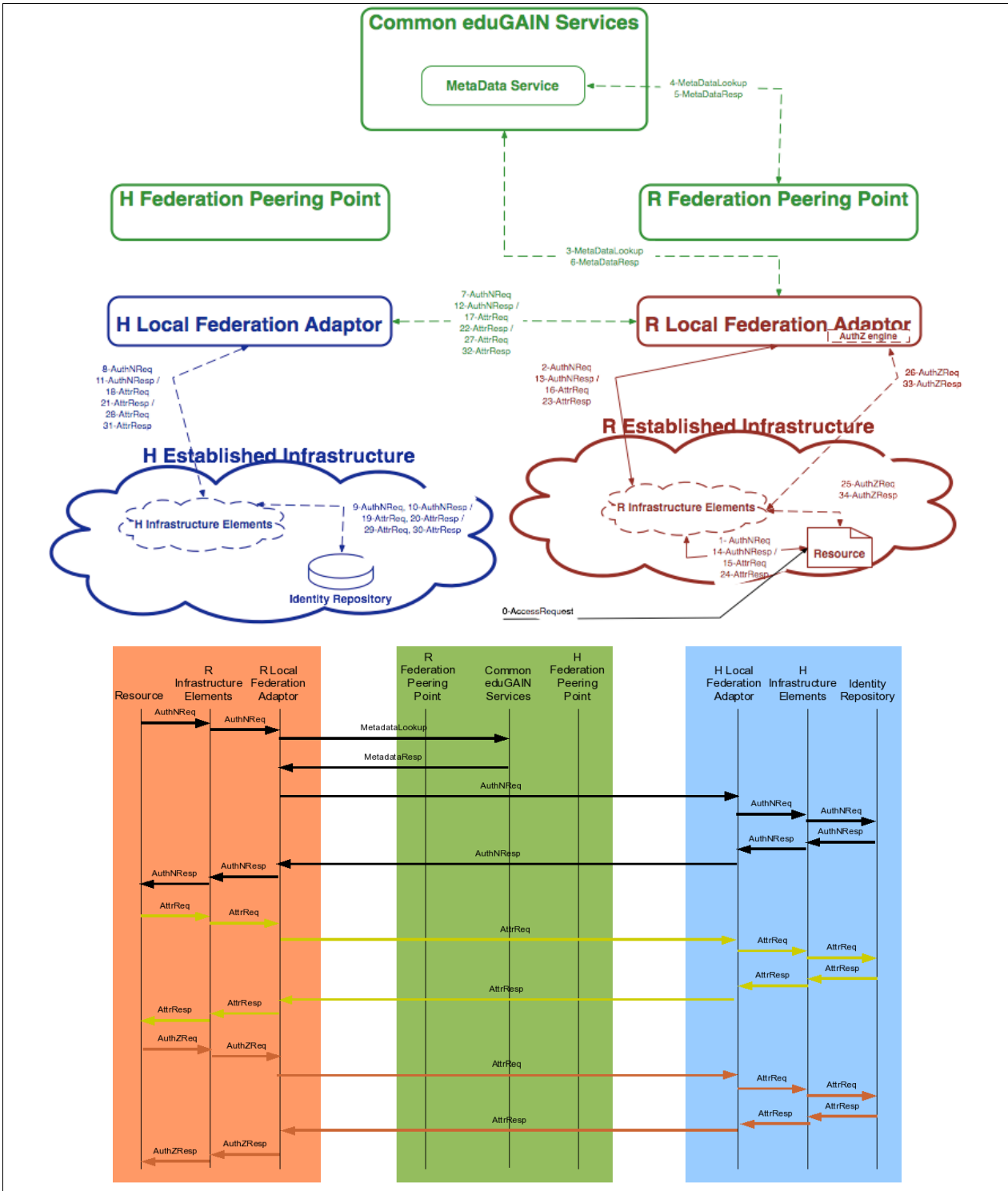


Figure 3-1: Configuration and time-flow diagrams for LFA usage at both domains

Project: GN2
 Deliverable Number: DJ5.2.2bis
 Date of Issue: 24/04/07
 EC Contract No.: 511082
 Document Code: GN2-07-024

3.1.2 LFA usage at Home. LAs at Remote

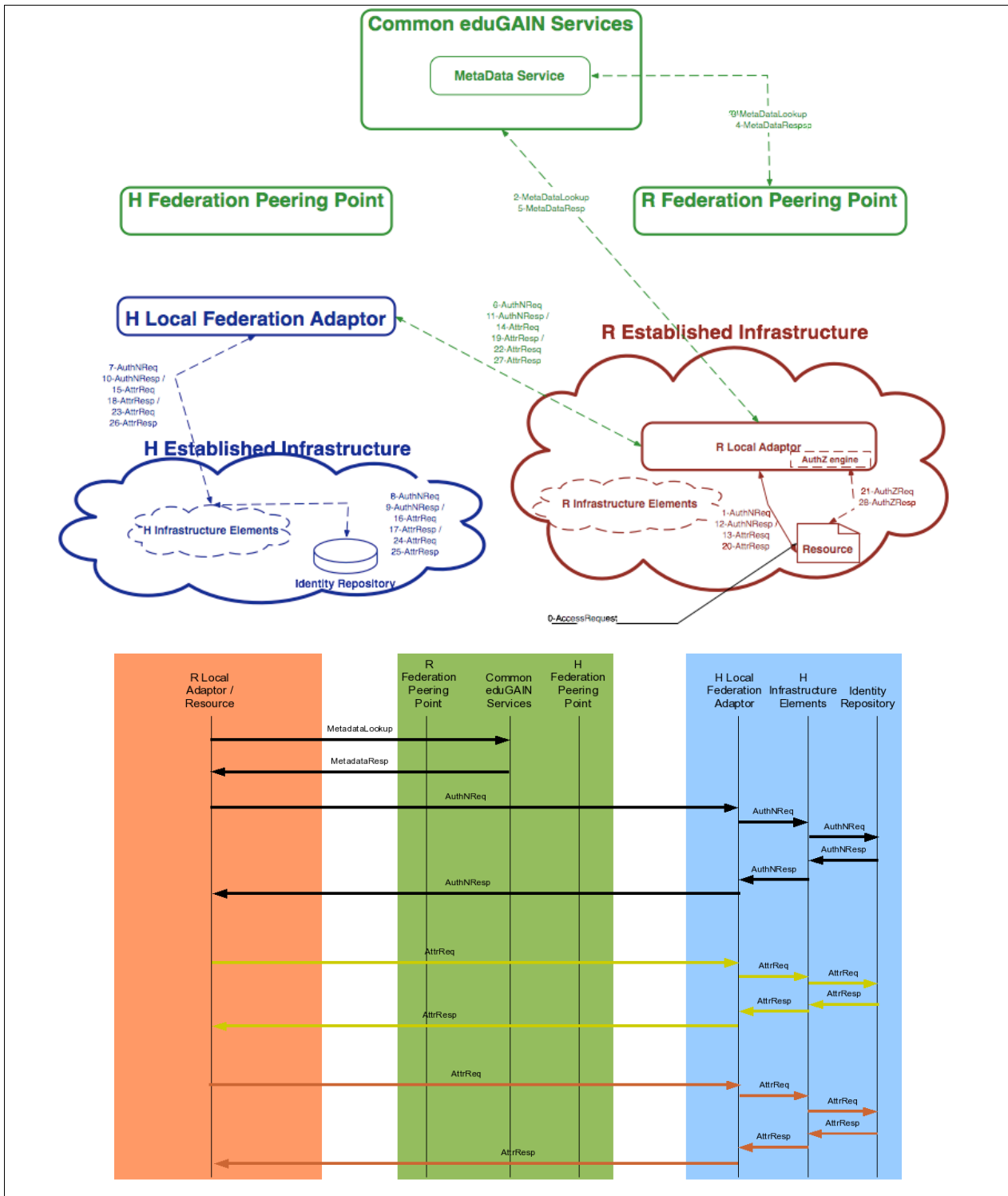


Figure 3-2: Configuration and time-flow diagrams for LFA usage at Home and LAs at Remote

Project: GN2
 Deliverable Number: DJ5.2.2bis
 Date of Issue: 24/04/07
 EC Contract No.: 511082
 Document Code: GN2-07-024

3.1.3 LAs at Home. LFA usage at Remote

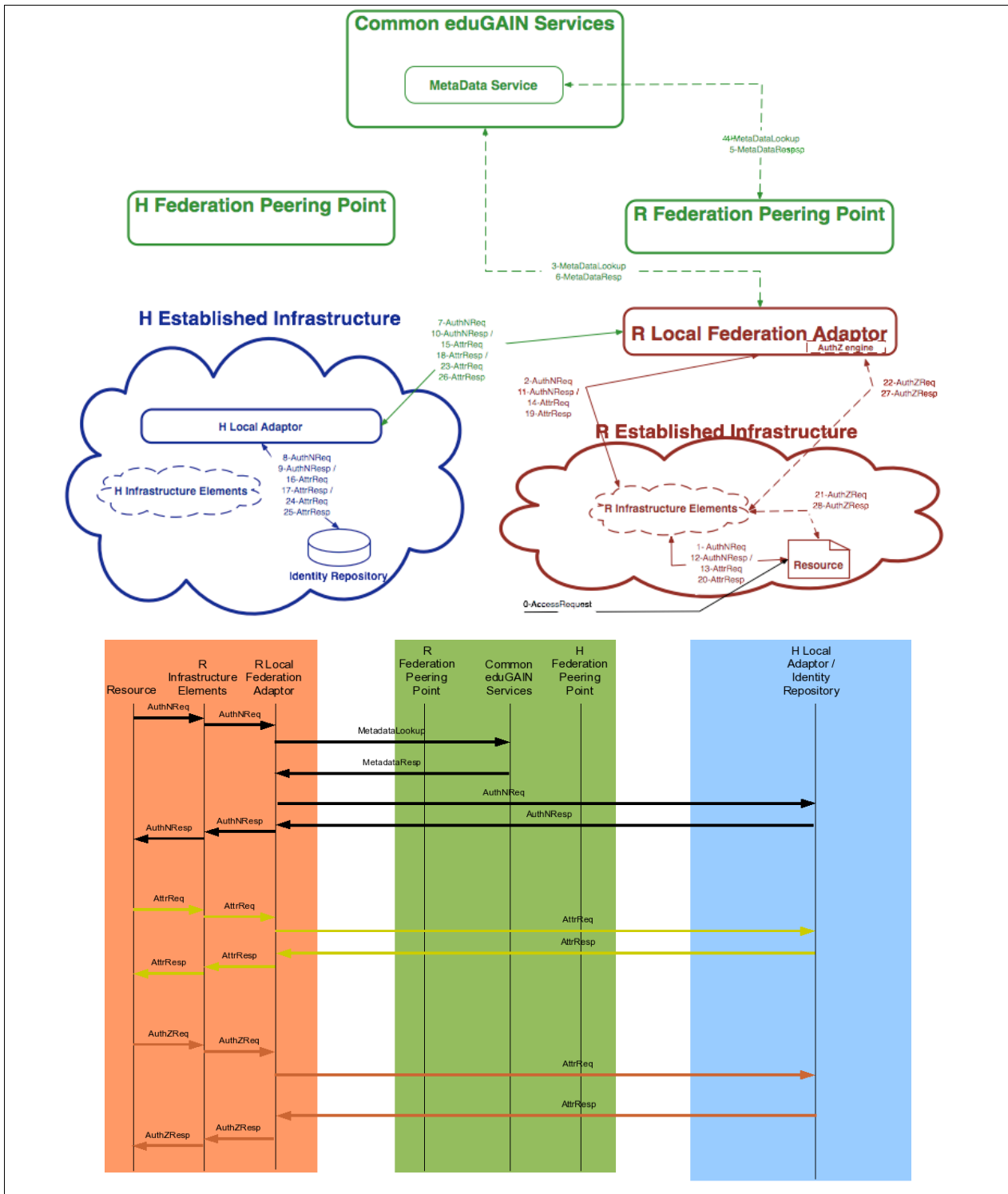


Figure 3-3: Configuration and time-flow diagrams for LAs at Home and LFA usage at Remote

Project: GN2
 Deliverable Number: DJ5.2.2bis
 Date of Issue: 24/04/07
 EC Contract No.: 511082
 Document Code: GN2-07-024

3.1.4 LAs at both domains

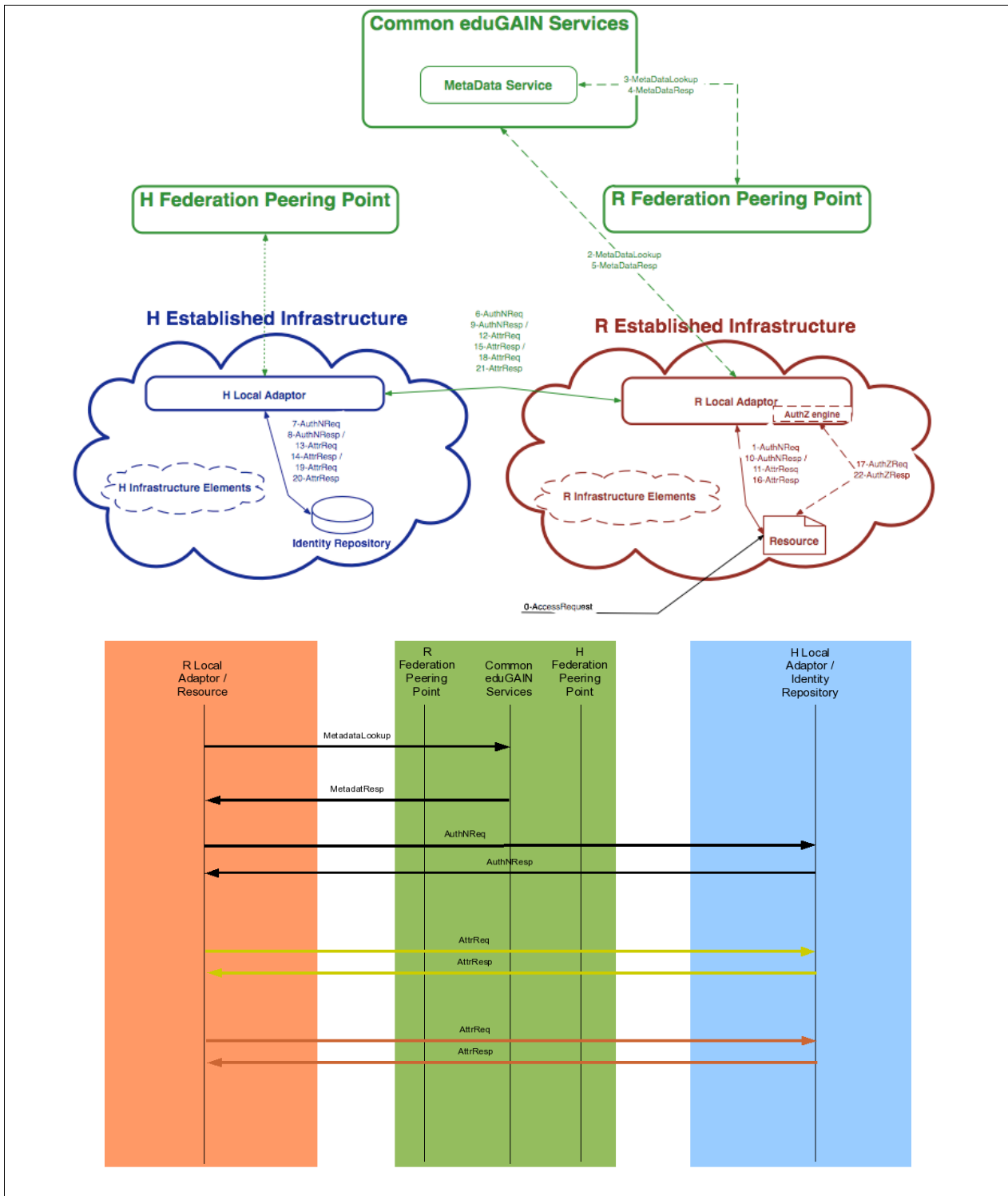


Figure 3-4: Configuration and time-flow diagrams for LAs at both domains

Project: GN2
 Deliverable Number: DJ5.2.2bis
 Date of Issue: 24/04/07
 EC Contract No.: 511082
 Document Code: GN2-07-024

3.2 Operational Definition

All operations will be carried out using a secure channel with mutual authentication (for the requester and responder). The typical example is a TLS channel with client and server authentication. Thus, each component of the architecture, upon reception of a request or a response, will be able to:

- Check the validity of the credentials presented by its peer (trust chain, non-revocation, etc.)
- Decide according to the values of these credentials, whether the peer is acceptable or not for the intended interaction.

Since the credentials of the requesting element will be provided through a secure channel, the operation arguments will not include any of them, although any of the components may decide to:

- Reply with an error response (indicating "Invalid credentials" to a certain request).
- Ignore an invalid request (logging an "Invalid credentials" event if required).

For the establishment of the required secure channels, a PKI is needed. JRA5 will investigate on the availability of existing appropriate CA, and, if required, provide an own root CA for these purposes.

In the here defined operations, state information is maintained by the end points (requester and responder), i.e. the traversed components do not need to keep track of this information. The end points explicitly exchange the required data, e.g. data for mapping responses and requests or for establishing cache mechanisms.

To enable Single Sign-on, initial requests from already authenticated users to resources may contain AA arguments to be used (either directly by the resource or by the infrastructure in which it is integrated) in performing the corresponding interactions. These arguments can be:

- A list of attributes and the corresponding values.
- An authentication reference or handle, to be used for retrieving attributes for the requester.
- A (set of) home locator(s), to be used in establishing the appropriate interfaces to request data from.
- Any kind of cache reference, to be passed to the appropriate component of eduGAIN.

Arguments must include all required timestamps and signatures to make them acceptable for eduGAIN and the intended components. The particular mechanisms for passing these arguments will depend on the precise protocol to be used in access requests (HTTP cookies, HTTP GET parameters, HTTP POST arguments, specific elements in XML files inside SOAP messages, etc.) and must be defined by the profile binding of each individual service. The translation of these arguments into actual eduGAIN operation arguments is up to the

| | |
|---------------------|------------|
| Project: | GN2 |
| Deliverable Number: | DJ5.2.2bis |
| Date of Issue: | 24/04/07 |
| EC Contract No.: | 511082 |
| Document Code: | GN2-07-024 |

service implementation. The eduGAIN operations will provide placeholders for accepting and returning values for these arguments.

Attributes included in operation parameters must comply with the following rules:

- Each attribute is defined by a name and a *namespace*. The namespace defines the scope of the attribute, i.e. the domain in which the same understanding of semantics exists, corresponding to the attribute name.
- The value of each attribute can be:
 - A simple value
 - A multiple value of one single type (multi-valued attribute)
 - Formally encoded according to structured frameworks, like the URN encodings defined by [SCHAC].

3.2.1 Authentication Service

The Authentication Service is in charge of verifying the identity of the entity (user, application...) requesting the access to a certain resource integrated in eduGAIN. In what follows, we will use the term *principal* to refer to this entity. The eduGAIN Authentication Service is not intended to perform credential exchange¹, but to establish that the principal has been properly identified (by an authentication authority), and it is possible to retrieve attributes about this principal using the eduGAIN infrastructure.

3.2.1.1 Authentication Request

The **AuthenticationRequest** contains the following parameters:

- **RequestID**: A reference (internally generated by the requester) for this operation to be used in the **InResponseTo** field of the corresponding response in further interactions regarding the request.
- **ProducerID**: The component identifier for the producer of the request.
- **ConsumerID**: The component identifier for the consumer of the request.
- **Resource**: The URI of the resource whose access request originated the operation.
- **AuthenticationType**: An identification of the protocol used in the authentication.

¹ Concrete credential exchange methods can be recommended or mandated by specific eduGAIN profiles.

| | |
|---------------------|------------|
| Project: | GN2 |
| Deliverable Number: | DJ5.2.2bis |
| Date of Issue: | 24/04/07 |
| EC Contract No.: | 511082 |
| Document Code: | GN2-07-024 |

- **[AuthenticatingPrincipal]**¹: A string (simple or in URN format) that identifies the principal [username/object] that is requesting the authentication.
- **[AuthenticationMethod]**: A URI that identifies a authentication method (e.g. password, PKI) to be used to authenticate the subject.
- **[HomeLocators]**: A list of data that will be used by the Metadata Service to evaluate the request. Each individual locator consists of an attribute-value pair. A very commonly used locator is the NAI (*Network Access Identifier*).
- **[HomeSite]**: The home site of the user where the request is going to be solved. This value corresponds to the **ConnectTo** value in the **MetadataLookup** operation.
- **[Shire]**: A URI that identifies the interface where the response to this request will be consumed.
- **[Referred]**: The URI from where the requester has been redirected to this service.
- **[CacheReference]**: An optional parameter including a **ResponseID** argument (see below) of a previous **AuthenticationResponse**.

3.2.1.2 AuthenticationResponse

The **AuthenticationResponse** contains the following parameters:

- **ResponseID**: A reference (internally generated by the responder) for this operation to be used in further interactions regarding the request.
- **ProducerID**: The component identifier for the producer of the response.
- **ConsumerID**: The component identifier for the consumer of the response.
- **NotBefore**: The date after which this response will be valid.
- **NotAfter**: The date after which this response will no longer be valid.
- **InResponseTo**: A reference to the **RequestID** contained in the input message of the operation.
- **Result**: The result for the request. Possible values are:
 - **Accepted**: Authentication accepted. Following parameters will include a handle (identification string) for the subject, to be used in further interactions.

¹ Parameters / results in [brackets] are optional.

| | |
|---------------------|------------|
| Project: | GN2 |
| Deliverable Number: | DJ5.2.2bis |
| Date of Issue: | 24/04/07 |
| EC Contract No.: | 511082 |
| Document Code: | GN2-07-024 |

- **ConnectTo**: Following parameters will include the interfaces where further AA requests can be satisfied.
- **RedirectUserTo**: Following parameters will include the interfaces which the requester can contact directly in order to get more information for evaluating authentication data.
- **InsufficientData**: The data provided in the request is not sufficient for discovering the appropriate interfaces and a direct interaction with the user is not feasible.
- **InvalidCredentials**: The result of the credential evaluation at the service does not allow providing any other response.
- **UnknownHomeSite**: The interfaces provided in the request are not valid
- **Fault**: An error occurred. The reason (and optionally additional information) will be provided by the below mentioned **errorReason** and **errorMessage** parameters.
- **[SubjectHandle]**: An identification string returned upon successful authentication.
- **[AttributeValueList]**: A list of elements, each of them corresponding to one attribute that the identity source is willing to deliver upon successful authentication and containing the following elements:
 - **AttributeNamespace**: The namespace corresponding to the attribute name.
 - **AttributeName**: The name of the requested attribute this element is providing data for.
 - **AttributeValue**: The value of the requested attribute (if policy allows it). Multi-valued attributes must be represented by several of these elements.
- **[Interfaces]**: It contains a list of URIs relevant with the **ConnectTo** or **RedirectUserTo** where the appropriate operations (according to the result) can be invoked. If more than one URI is provided, they must be considered functionally equivalent and the order in which they appear inside the argument will not be considered relevant.
- **[AdditionalData]**: Any other data the Authentication Service is willing to include. Possible purposes of these data are to provide additional logging and diagnostic information, or information relevant to the requesting resource, and any assumptions on their format is out of scope of this document.

In case of errors (**Result = Fault**), the **AuthenticationResponse** contains the following parameters:

- **errorReason**: One of the possible error types taken into account in the system:
 - **TrustError**: Error checking the authenticity of the message

| | |
|---------------------|------------|
| Project: | GN2 |
| Deliverable Number: | DJ5.2.2bis |
| Date of Issue: | 24/04/07 |
| EC Contract No.: | 511082 |
| Document Code: | GN2-07-024 |

- **MalformedMessage:** Error in the structure or codification of the message
- **ExpiredMessage:** The validity time of the content of the message has expired.
- **IncompatibleVersions:** The version of the message is incompatible with the version of the service.
- **errorMessage:** Information that gives extra information about the error.

3.2.2 Metadata Service

The Metadata Service is in charge of providing the BE with the information where authentication and attribute data can be obtained in order to make a decision on a certain request, or where the authorisation decision can be made.

The local federations are responsible for publishing updated information about their BEs to the Metadata Service. There are three basic operations available and a common response for them.

3.2.2.1 MetadataLookup

This operation is performed when sufficient information for uniquely identifying the BE is known to the requesting entity. The **MetadataLookup** contains the following parameters:

- **FederationID:** The identifier (name) of the federation to which the concerned BE belongs.
- **EntityID:** The identifier (name) of the BE subject of the metadata request.

3.2.2.2 MetadataSearch

When the requesting entity does not know the entity identifier of the bridging element it wants metadata for, it must perform a search using those available parameters. We will refer to the search parameters as *Home Locators* (HL). Which HLs are supported depends on the eduGAIN profile being implemented. Though other kinds of HLs can be defined by different profiles (and even bi-lateral agreements), the following classes of HLs MUST be implemented by any MDS server:

homeDomain: A domain name indicating the user's home location.

URN: A value encoded in URN format. Intended for eduGAIN component identifiers, although other users (like particular entitlement values) may be applicable.

| | |
|---------------------|------------|
| Project: | GN2 |
| Deliverable Number: | DJ5.2.2bis |
| Date of Issue: | 24/04/07 |
| EC Contract No.: | 511082 |
| Document Code: | GN2-07-024 |

netID: An e-mail-like identifier in the form `localPart@FQDN`, though not necessarily a valid e-mail address of any kind.

The **MetadataSearch** contains the following parameters:

- **HomeLocators**: A list containing the identifier and actual value of the HLs to be searched for.
- **[FederationID]**: The identifier (name) of the federation to which the concerned BE belongs.

3.2.2.3 *MetadataPublish*

The corresponding Federation Peering Point publishes Metadata for eduGAIN BEs to make it available through the MDS. The **MetadataPublish** operation is used for this purpose and contains the following parameters:

- **FederationID**: The identifier (name) of the federation to which the concerned BE belongs.
- **EntityID**: The identifier (name) of the BE subject of the metadata request.
- **Metadata**: An XML document containing the identifiers and actual values of the metadata fields being published

3.2.2.4 *MetadataResponse*

The **MetadataResponse** contains the following parameters:

- **Result**: The result for the request. Possible values are:
 - **Accepted**: Request accepted. An HTTP success code (200) is returned by the MDS, and the body of the HTTP response message will contain the requested metadata.
 - **InvalidCredentials**: The result of the credential evaluation at the service does not allow providing any other response. The MDS returns an error code and an explanatory error message.
 - **Fault**: An error occurred. The reason (and optionally additional information) will be provided by the below mentioned **errorReason** and **errorMessage** parameters.
- **[Metadata]**: The metadata contains the necessary information to contact the home location(s). Depending on the request, the metadata section may include information for one or more home locations. For each home location a set of interfaces is given. Information about what kind of services is available on each interface can be extracted from the metadata. Services can be one of those defined in the different eduGAIN profiles, one of the predefined in SAML Metadata [SAMLMD], or a future eduGAIN specific service not yet defined as (for example) logging and diagnostics services.

| | |
|---------------------|------------|
| Project: | GN2 |
| Deliverable Number: | DJ5.2.2bis |
| Date of Issue: | 24/04/07 |
| EC Contract No.: | 511082 |
| Document Code: | GN2-07-024 |

In case of errors (**Result = Fault**), the **MetadataResponse** contains the following parameters:

- **errorReason**: One of the possible error types taken into account in the system:
 - **TrustError**: Error checking the authenticity of the message
 - **InvalidQueryParameters**: one or more of the provided query parameters are invalid or not supported.
 - **MalformedMessage**: Error in the structure or codification of the message
 - **ExpiredMessage**: The validity time of the content of the message has expired.
 - **IncompatibleVersions**: The version of the message is incompatible with the version of the service.
- **errorMessage**: Information that gives extra information about the error.

3.2.3 Attribute Exchange Service

The Attribute Exchange Service is in charge of providing (additional) attributes about a principal.

3.2.3.1 *AttributeRequest*

The **AttributeRequest** contains the following parameters:

- **RequestID**: A reference (internally generated by the requester) for this operation to be used in further interactions regarding the request.
- **ProducerID**: The component identifier for the producer of the request.
- **ConsumerID**: The component identifier for the consumer of the request.
- **Resource**: The URI of the resource whose access request originated the operation.
- **SubjectHandle**: The identification string obtained upon the last successful authentication of the subject requesting access to the resource.
- **[AttributeNameList]**: A list of the attribute names whose values are requested. If no **AttributeNameList** is included in a request, all “available” (those that can be released according to the applicable policy) attributes are requested.

- [**HomeSite**]: The home site of the user where the request is going to be solved. This value corresponds to the **ConnectTo** value in the **MetadataLookup** operation.
- [**Referred**]: The URI from where the requester has been redirected to this service.
- [**CacheReference**]: An optional parameter including a **ResponseID** argument (see below) of a previous **AttributeResponse**.

3.2.3.2 *AttributeResponse*

The **AttributeResponse** contains the following parameters:

- **ResponseID**: A reference (internally generated by the responder) for this operation to be used in further interactions regarding the request.
- **ProducerID**: The component identifier for the producer of the response.
- **ConsumerID**: The component identifier for the consumer of the response.
- **NotBefore**: The date after which this response will be valid.
- **NotAfter**: The date after which this response will no longer be valid.
- **InResponseTo**: A reference to the **RequestID** contained in the input message of the operation.
- **Result**: The result for the request. Possible values are:
 - **Accepted**: Operation accepted. Following parameters will provide data about the requested attributes.
 - **ConnectTo**: Following parameters will include the interfaces where further AA requests can be satisfied.
 - **InvalidCredentials**: The result of the credential evaluation at the service does not allow providing any other response.
 - **UnknownHomeSite**: The interfaces provided in the request are not valid
 - **Fault**: An error occurred. The reason (and optionally additional information) will be provided by the below mentioned **errorReason** and **errorMessage** parameters.
- **AttributeValueList**: A list of elements, each of them corresponding to one of the requested attributes and containing the following elements:

| | |
|---------------------|------------|
| Project: | GN2 |
| Deliverable Number: | DJ5.2.2bis |
| Date of Issue: | 24/04/07 |
| EC Contract No.: | 511082 |
| Document Code: | GN2-07-024 |

- **AttributeNameSpace**: The namespace corresponding to the attribute name.
- **AttributeName**: The name of the requested attribute this element is providing data for.
- **AttributeValue**: The value of the requested attribute (if policy allows it). Multi-valued attributes must be represented by several of these elements.
- **[SubjectHandle]**: An identification string associated with the entity whose attributes are sent.
- **[Interfaces]**: This optional argument must only be included in the case of a **ConnectTo** result. It shall contain a list of URIs where the appropriate operations (according to the result) can be invoked. If more than one URI is provided, they must be considered functionally equivalent and the order in which they appear inside the argument will not be considered relevant.
- **[AdditionalData]**: Any other data the identity provider is willing to include. Possible purposes of these data are to provide additional logging and diagnostic information, or information relevant to the requesting resource, and any assumptions on their format is out of scope of this document.

In case of errors (**Result = Fault**), the **AttributeResponse** contains the following parameters:

- **errorReason**: One of the possible error types taken into account in the system:
 - **TrustError**: Error checking the authenticity of the message
 - **MalformedMessage**: Error in the structure or codification of the message
 - **ExpiredMessage**: The validity time of the content of the message has expired.
 - **IncompatibleVersions**: The version of the message is incompatible with the version of the service.
- **errorMessage**: Information that gives extra information about the error.

3.2.4 Authorisation Service

The Authorisation Service is used to decide whether a certain action is allowed on a certain resource, given a certain set of circumstances: identity and attributes of the requester and any other circumstances like applicable policy, time constraints, etc. It is provided through the BE the resource belongs to, and (as in the case of any other service), can be located using the MDS through the corresponding Federation Peering Point.

| | |
|---------------------|------------|
| Project: | GN2 |
| Deliverable Number: | DJ5.2.2bis |
| Date of Issue: | 24/04/07 |
| EC Contract No.: | 511082 |
| Document Code: | GN2-07-024 |

3.2.4.1 *AuthorizationRequest*

The **AuthorizationRequest** contains the following parameters:

- **RequestID**: A reference (internally generated by the requester) for this operation to be used in further interactions regarding the request.
- **ProducerID**: The component identifier for the producer of the request.
- **ConsumerID**: The component identifier for the consumer of the request.
- **Resource**: The URI of the resource the authorisation is requested for.
- **Action**: An identifier for the action the requester is willing to perform on the resource. A string from a controlled vocabulary¹ and/or a formally defined URN can be used as value.
- **AttributeValueList**: A list of the attributes (as established by the authentication procedures) provided by and/or obtained for the element requesting the authorisation. Each individual element consists of an attribute name-value pair.
- **[AttributeAuthority]**: An optional URI of an eduGAIN instance that can help in making the authorisation decision by providing additional attributes.
- **[SubjectHandle]**: The identification string obtained upon the last successful authentication of the subject requesting access to the resource.
- **[PolicyReference]**: An optional list of policy references to be applied in the decision. As above, the values can be strings from a controlled vocabulary and/or formally defined URNs.
- **[CacheReference]**: An optional parameter including a **ResponseID** argument (see below) of a previous **AuthorizationResponse**.

3.2.4.2 *AuthorizationResponse*

The **AuthorizationResponse** contains the following parameters:

- **ResponseID**: A reference (internally generated by the responder) for this operation to be used in further interactions regarding the request.

¹ Actions and policy references as described here are meaningful in the corresponding application context, and thus are transparent to the eduGAIN infrastructure itself. User applications may establish their own sets of them, by means of multi- or bi-lateral agreements, requirements in the framework of policy management authorities, or even official standards where applicable.

| | |
|---------------------|------------|
| Project: | GN2 |
| Deliverable Number: | DJ5.2.2bis |
| Date of Issue: | 24/04/07 |
| EC Contract No.: | 511082 |
| Document Code: | GN2-07-024 |

- **ProducerID**: The component identifier for the producer of the response.
- **ConsumerID**: The component identifier for the consumer of the response.
- **NotBefore**: The date after which this response will be valid.
- **NotAfter**: The date after which this response will no longer be valid.
- **InResponseTo**: A reference to the **RequestID** contained in the input message of the operation.
- **Result**: The result of the request. Possible values are:
 - **Accept**: The request is authorised.
 - **ConnectTo**: Following parameters will include the interfaces where further AA requests can be satisfied.
 - **Deny**: The request is denied.
 - **InvalidCredentials**: The result of the credential evaluation at the service does not allow providing any other response.
 - **Fault**: An error occurred. The reason (and optionally additional information) will be provided by the below mentioned **errorReason** and **errorMessage** parameters.
- **[Interfaces]**: This optional argument must only be included in the case of a **ConnectTo** result. It shall contain a list of URIs where the appropriate operations (according to the result) can be invoked. If more than one URI is provided, they must be considered functionally equivalent and the order in which they appear inside the argument will not be considered relevant.
- **[AdditionalData]**: Any other data related to the authorisation decision. Possible purposes of these data are to provide additional logging and diagnostic information, or information relevant to the requesting resource, and any assumptions on their format is out of scope of this document.

In case of errors (**Result = Fault**), the **AuthorizationResponse** contains the following parameters:

- **errorReason**: One of the possible error types taken into account in the system:
 - **TrustError**: Error checking the authenticity of the message
 - **MalformedMessage**: Error in the structure or codification of the message

- **ExpiredMessage:** The validity time of the content of the message has expired.
- **IncompatibleVersions:** The version of the message is incompatible with the version of the service.
- **errorMessage:** Information that gives extra information about the error.

4 Protocol and Bindings

In order to perform authentication and authorisation interactions, the Security Assertion Mark-up Language (SAML), in combination with any acceptable transport over a secured channel, is in wide use and already provides large parts of the required functionality described in this document (cf. Figure 4-1) [W3CREC-soap12-part2]. SAML is a set of standards well suited to the eduGAIN tasks. Therefore, the definitions in this chapter about the 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. Some of the eduGAIN functionality requires extensions to SAML1.1, or the use of SAML2.0. At implementation time federation-aware software may have implemented SAML 2.0, which may be reflected in the use of SAML 2.0 in eduGAIN as well. These definitions constitute the **Basic eduGAIN Profile**, which will be refined and extended to other profiles in the profile detailed specification document that constitutes, with this one, the architectural specification of eduGAIN.

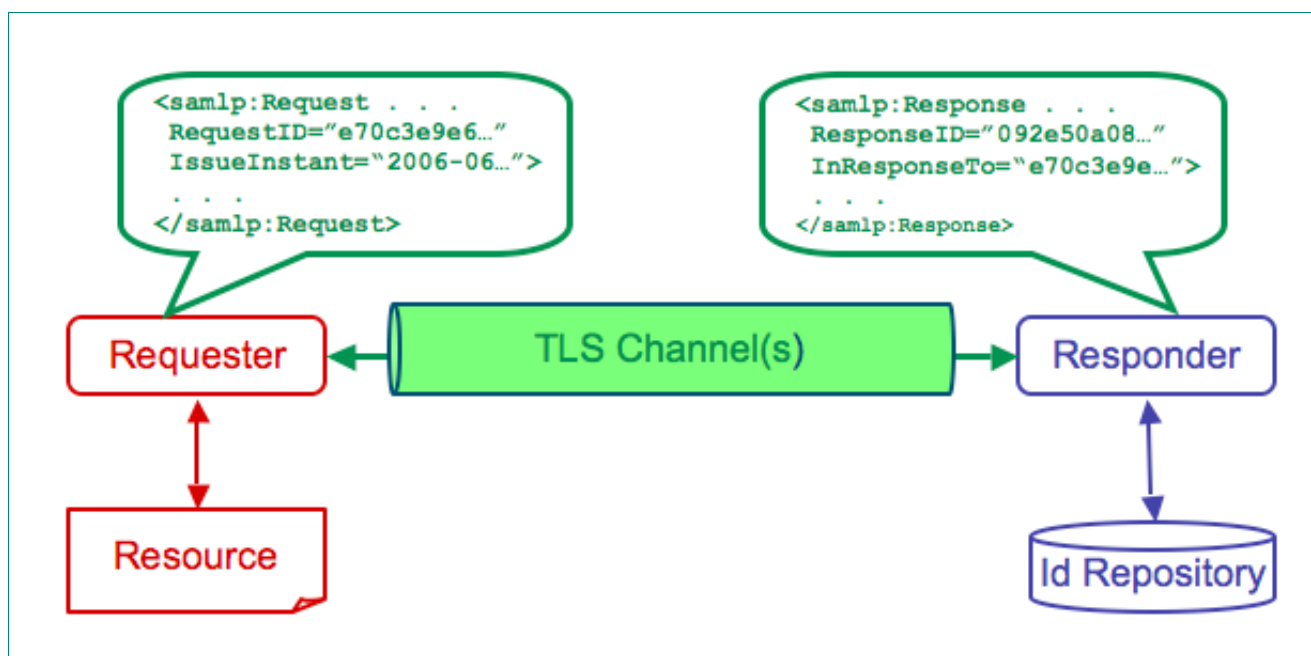


Figure 4-1: Schematic overview of an abstract AA operation

eduGAIN abstract messages, as described in the previous section, are mapped by the eduGAIN Basic Profile into SAML `Response` and `Request` elements, according to the general rules described in the detailed profile specification.

4.1 Authentication Protocol

Authentication requests and responses deal with users or systems (principals) accessing resources. Commonly a request to access a resource is started with the authentication initialisation phase. Depending on (local) implementation of authentication at the Home Organisation (Authentication Authority) of the authenticating principal and the kind of authentication, one or several interactions might take place and (re-) authentication might be handled as a single session: single sign-on.

eduGAIN is a superstructure linking federations and therefore independent of local Identity Management procedures. It is expected that the local federations will implement the necessary translations from the local federation specifics to the eduGAIN infrastructure. The authentication protocol is implemented as a request/response pair: **AuthenticationRequest/AuthenticationResponse**.

The **AuthenticationRequest** needs to be able to support several authentication-methods, for example RADIUS and authentication by WebISO systems like A-Select or PAPI. It is desirable to map the **AuthenticationRequest** to existing SAML elements. This can be achieved by profiling the existing SAML elements `AuthenticationQuery` and `AuthenticatioStatement`.

To be able to understand how the actual authentication is handled within the BE, a functional description of this federation component is given. From the authentication point of view the BE consist of three (functional) components:

| | |
|---------------------|------------|
| Project: | GN2 |
| Deliverable Number: | DJ5.2.2bis |
| Date of Issue: | 24/04/07 |
| EC Contract No.: | 511082 |
| Document Code: | GN2-07-024 |

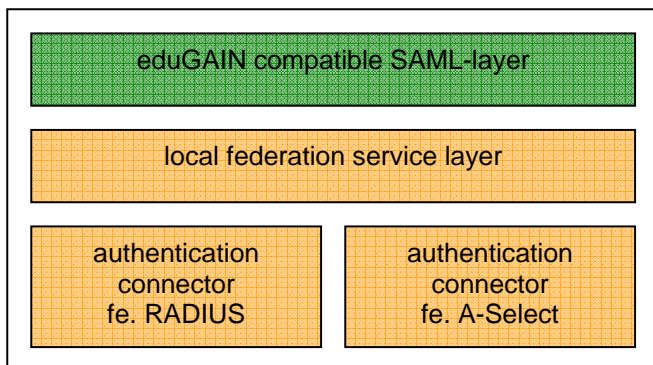


Figure 4-2: Authentication functional components

The BE's eduGAIN compatible SAML-layer is in general supposed to handle *all* communication with the eduGAIN federation concerning authentication (and attribute gathering and authorisation). This layer handles the **AuthenticationRequest**, in other words: it will respond with an `AuthenticationStatement` if an `AuthenticationQuery` is received.

Depending on local implementation of the BE the local federation service can either be a lightweight transparent (protocol) gateway or a full service implementation of a (national) federation like A-Select, FEIDE, PAPI or Shibboleth. It is up to the local federation service layer to communicate with or redirect to the (legacy) backend that handle the actual authentication. As explained before, this document only handles the eduGAIN compatible SAML-layer and describes the elements that can be *used* in the local federation service layer.

In order to keep implementation simple it is *desirable* to keep the authentication phase stateless (for the eduGAIN components). If state handling is required, this will have to be solved by either the local federation service layer or the authentication backend of the Authentication Authority.

The authentication flows between any two (legacy) authentication backends go through the Remote and Home Bridging Elements (BE-R/BE-H). Note that the MDS request and response are not actually part of the authentication flow and are described as a separate binding in the following section.

The SAML element `AuthenticationQuery` is used to make the query: "What assertions containing authentication statements are available for this subject". This means, the query requires that the actual authentication has been handled already (in another process). The eduGAIN Authentication Service however can be used in two scenarios: either authenticate a principal or validate an (previously established) authentication.

For the first scenario it is needed to be able to add additional information, such as authentication material and originating URI to the request so that an Authentication Authority in the Home Organisation can actually authenticate the request. An example would be to add the RADIUS authentication credentials encapsulated in EAP as blob to the request. This can be achieved by adding `SubjectConfirmationData` to the request.

| | |
|---------------------|------------|
| Project: | GN2 |
| Deliverable Number: | DJ5.2.2bis |
| Date of Issue: | 24/04/07 |
| EC Contract No.: | 511082 |
| Document Code: | GN2-07-024 |

The second scenario, validating an previously established authentication is often the case with Web ISO. The user wanting to access the resource is not authenticated (no ticket from the resource) and unknown within the federation (no ticket from the federation) of the resource. The (remote) federation finds out which home organisation the user belongs to (MDS) and redirects them (with an *AuthenticationQuery*) to their Home Bridging Element. Depending on the national implementation of the Home Organization, a local federation might be in place (a ticket is available) or the user is directed to the Authentication Authority that proves their identity and redirects them. Finally the H-BE *states* that the user is authenticated and sends the user back towards the R-BE that, based on the eduGAIN Federation, *can* rely on the established authentication and set a local federation ticket. For this flow based on redirects, additional parameters are required as well.

4.2 Attribute Exchange Protocol

In the eduGAIN architecture concept, attributes are generally managed by the user's home organisations and stored in local *Identity Repositories*. Attribute Requests are usually issued by an entity "on" or very close to the service in order to request information about individuals trying to access the resources - we will use the term *Attribute Requestor* (AR) for those entities in the following sections. In this connection, attributes (usually as name/value pairs) are used to describe properties (and, related to them, rights) of individuals, and therefore can be used to control the access of users to requested resources, but they may as well be used to depict additional information (of any type) about the individuals, e.g. age, address or billing information (as described in [DJ521]).

Attribute Requests, possibly originating from various ARs located somewhere in the network, have finally to be directed towards the responsible *Attribute Authority* (AAu) at the user's home organisation. Furthermore, if a local AA infrastructure conceptually supports a manual control of attribute releases by the user, the inter-institutional/inter-architectural communication via eduGAIN also has to support the required interaction with the user in the attribute retrieval phase to implement this control.

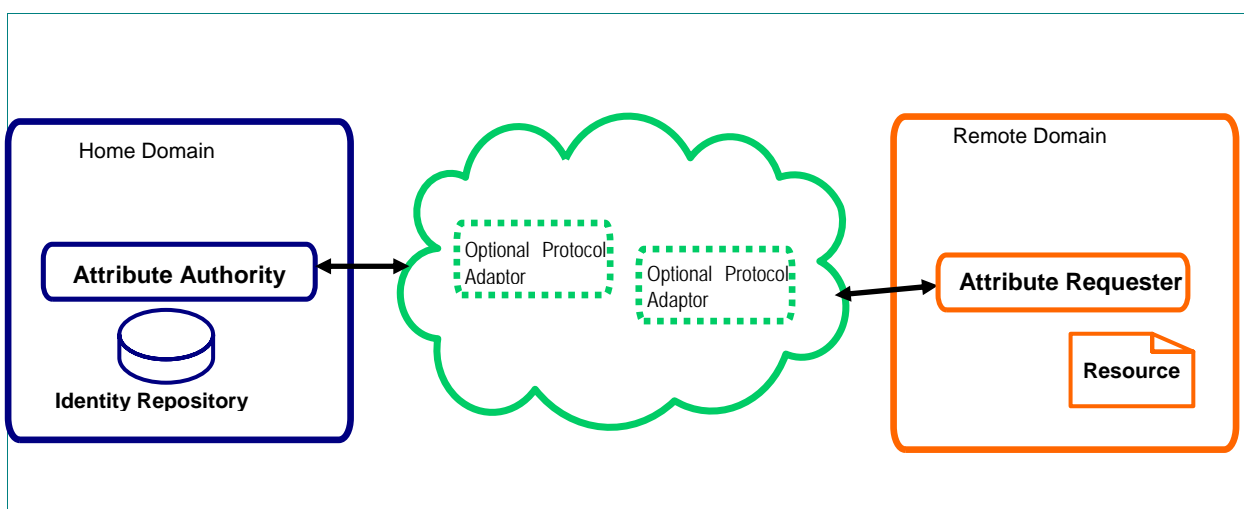


Figure 4-3: Generalised attribute exchange

| | |
|---------------------|------------|
| Project: | GN2 |
| Deliverable Number: | DJ5.2.2bis |
| Date of Issue: | 24/04/07 |
| EC Contract No.: | 511082 |
| Document Code: | GN2-07-024 |

As described in chapter 3, several varying integration stages between existing local AA infrastructures and the common eduGAIN infrastructure may exist at the different institutions, reflecting all possible combinations of eduGAIN-aware and -non-aware local AA subsystems installed at the sites. Concerning the handling and usage of attributes, it might occur that a local AA infrastructure does not use (and therefore not support) a certain number of required attributes for the federated AA Infrastructure to work as expected. Consequently, a specific processing (adaptation/conversion) of Attribute Requests, Responses, and even attribute semantics has to take place in order to support the cross-institutional and cross-architectural exchange of user-related data between different AA technologies. Especially in cases where the number and/or semantics of certain attributes provided from the Home Organisation is not sufficient to base direct authorisation decisions on, the above-operating AAI has to provide the required functions for the translation and/or reorganisation of available attributes and corresponding semantics. This processing has to be executed by the eduGAIN architecture; it will be implemented by the *Local Federation Adaptors* described in section 2.1. On the other hand, if a local AA subsystem requires interaction with the user at any stage of the protocol sequence (like e.g. the above-mentioned affirmation of attribute releases), the eduGAIN architecture has to support this by providing the necessary communication channels to the user.

4.2.1 The Attribute Exchange Operation

As stated in section 3.2.3, the Attribute Exchange operation, from the conceptual viewpoint, consists of a single request/response message pair. However, depending on the number of translating Bridging Elements involved in the communication between the Attribute Requestor at the resource and the Attribute Authority at the user's HI, (e.g. for translating between different protocols), the number of message translations performed by Bridging Elements may vary. This is mirrored in a number of different possible message sequences (possibly associated with different protocols) between the entities involved in the communication depending on the number of 'hops' between them.

- The most simple amongst the possible scenarios is the *direct* retrieval of attributes from an eduGAIN-compliant Attribute Authority (connected via a Local Adapter) performed by an eduGAIN-compliant Attribute Requester (also connected via a Local Adapter). These two entities can communicate directly with each other using the Attribute Exchange profile defined below, without any further translation of protocols and semantics. As mentioned above, a single Attribute Exchange operation therewith consists of exactly two messages, the *Attribute Request* and the *Attribute Response* message.
- If one or two Local Federation Adaptors are involved in order to connect local (non-eduGAIN-compliant) AAI to eduGAIN, protocol and message transformations are performed. Independent of the protocols and messages used between the local AAI and the Local Federation Adaptors, the profile used between the eduGAIN-compliant Bridging Elements is the same as mentioned above (and defined below). As this document focuses on the eduGAIN architecture, the procedures described here and the corresponding profile (stating technical details) only refer to the communication between eduGAIN-compliant entities. These build the basis for all subsequent scenarios, which can be seen as extensions of the basic scenario by adding one or more Bridging Elements. While protocols between non-eduGAIN-compliant entities (e.g. a Shibboleth Service Provider or a PAPI PoA) and translators may be proprietary, the communication between eduGAIN-compliant entities always uses the Attribute Exchange Profile depicted here.

| | |
|---------------------|------------|
| Project: | GN2 |
| Deliverable Number: | DJ5.2.2bis |
| Date of Issue: | 24/04/07 |
| EC Contract No.: | 511082 |
| Document Code: | GN2-07-024 |

As stated in section 3.2, the attribute retrieval is performed after the initial authentication of the user is completed. Moreover, as described in that section, one of the results of a successful completed authentication procedure in the simple scenario (using two eduGAIN-aware peers) is a **ConnectTo** handle, returned to the authenticating peer, containing a pointer to the appropriate Attribute Authority of the user. With this address information, the AR is able to establish a direct connection to the AAu for requesting attributes.

The attribute exchange between an eduGAIN-compliant AR and an eduGAIN-compliant AAu is performed using SOAP/HTTP over a TLS connection. The requests are encoded as `AttributeQuery` elements inside a SAML 1.1 `Request` and the responses as `AttributeStatement` elements inside SAML 1.1 `Response`.

By using TLS, it is assured that the communication between the Attribute Requestor and the Attribute Authority at the user's Home Institution is secured and provides implicit mutual authentication, i.e. the authentication of the involved peers is already done on transport layer.

If a multi-hop communication is given (i.e. a chain of multiple TLS connections between different Bridging Elements is used), the transport layer does not provide direct end-to-end trust. Without using further trust mechanisms, e.g. digitally signing the end-to-end messages using the XML-Signature framework [XML-DSIG], the peers only have limited control over their trust differentiation, i.e. they implicitly trust all entities 'behind' their trusted peers (cf. chapter 7.2.2).

A description of the abstract parameters contained in the Attribute Requests and Responses is given in section 3.2.3. As stated above, these parameters will be encoded using SAML 1.1 elements and attributes.

4.3 Authorisation Protocol

The authorisation phase takes place after the authentication phase and optionally the attribute retrieval operation have been completed successfully. Even though a resource is not strictly obliged to use the authorisation interface (it could implement a minimal stand-alone decision module based only on authentication or attribute statements), it is strongly encouraged to use the interface in order to avoid duplication of work (i.e. building an own authorisation decision module) within resources.

The authorisation decision can be made in one atomic step, the operation **AuthorizationRequest**. Due to the atomicity of the operation it is desirable to keep the authorisation phase stateless, i.e. the Remote FPP does not keep information about authorisation decisions persistently between operations. It may, however, keep the result of previous authorisation decisions in a temporary storage (cache) to speed up further requests as long as the caching is unambiguous and does not lead to an inconsistent state, i.e. if the authorisation decision relies on external attribute sources this decision may not be cached and the external sources must be queried again on subsequent requests in order to stay in sync with the external attribute source.

As stated in section 3.2.4, the operation consists of a single request/response pair along with an error message. Requests originate from the resource and mainly include a list of attributes that apply to the requesting user (which were gathered in the attribute request phase) and a hint what the user intends to do with

| | |
|---------------------|------------|
| Project: | GN2 |
| Deliverable Number: | DJ5.2.2bis |
| Date of Issue: | 24/04/07 |
| EC Contract No.: | 511082 |
| Document Code: | GN2-07-024 |

the resource. In most use cases, only one single action needs to be requested by the resource. While it is supported to request authorisation for an arbitrary number of actions (by using multiple instances of the **Action** element, see below), it is advisable to request the simplest possible authorisation statement to keep size and complexity of the exchanged messages minimal. Further details about the exact structure of the request are contained in section 3.2.4.1.

The response to this request is a decision with only three possible outcomes, conveyed in the SAML element `AuthorizationDecisionStatement`: the user is either authorised to use the resource (`Permit`) or not (`Deny`), or the eduGAIN element that has processed the authorisation request could not decide about the request, in which case the answer is `Indeterminate`. In the case that more than one action has been requested it is possible that a `Permit` answer contains only a subset of the requested actions if only a subset of the requested actions is allowed. Since the response does not change the content of the requester's assertion, it needs not repeat the assertion itself but instead provides the request's `AssertionIDReference`. The fact that only a subset of the requested actions may be returned makes it a requirement for the requester to check which actions have been permitted after receiving the response. An indeterminate answer means the AAI instance is not the authoritative source to answer the request. In this case an optional `StatusMessage` element is added in the response to the resource to redirect the resource to the authoritative eduGAIN element (which effectively implements the abstract operation element **Interfaces**). A summary of the message parts of the response is given in section 3.2.4.2.

The SAML standard allows a great number of optional message elements, of which not all parts are needed by the eduGAIN authorisation interface. So, as a general rule all message parts of the SAML elements that are not explicitly mentioned hereafter are not used by the authorisation interface and will be ignored.

4.4 Metadata Protocol

The Metadata Service is used by the bridging elements to determine the home interfaces where an authentication or attribute request can be satisfied, and to establish trust among these interfaces and the requesting element. In the case where the appropriate home location is known and trust is established, this service is not used.

Even though the MDS holds all the metadata documents providing the trust foundation in the confederation, no significant trust SHOULD be assigned to the MDS itself. The MDS is trusted by the eduGAIN entities to forward metadata, but not to issue metadata. Consequently the MDS will never sign any metadata itself, but forward signatures from the publishers.

Since the MDS is only trusted to forward metadata, compromising the MDS will not allow injection of altered metadata documents. However compromising the MDS may deny publishers the possibility of performing updates, and hence allow distribution of old documents to the metadata consumers.

| | |
|---------------------|------------|
| Project: | GN2 |
| Deliverable Number: | DJ5.2.2bis |
| Date of Issue: | 24/04/07 |
| EC Contract No.: | 511082 |
| Document Code: | GN2-07-024 |

5 Generic Use Case

The purpose of this chapter is to illustrate the eduGAIN architecture and design with several real life examples, in which eduGAIN is used to provide the access to the networked resources beyond the limits of national or group-wide infrastructures.

Although there is a number of existing AA solutions, e.g. at institutional (campus) level, or at NREN level (spanning several institutions), all of which control users' access to resources based on authentication and authorisation information in their home institution HI, eduGAIN is also aimed to serve in the situations in which there is no local AAI. In these cases, specific BEs (acting as LAs) act as mediators between resources and eduGAIN.

In a typical use case eduGAIN acts as a superstructure providing the necessary interfaces between the corresponding infrastructures enabling the communication between the home institution HI and resource institution RI in order to resolve the user's request for the resource. Through its Metadata Service eduGAIN provides the information on where to send and receive requests across multiple federations and domains.

The use case discussed here is based on the assumptions already mentioned in the requirements document [DJ521]:

1. Any user U is given an appropriate digital identity by his home institution HI.
2. Digital identities issued by the HI are trusted and valid in a federation of participating institutions. This extends to the eduGAIN federation.
3. The control of the authorisation to access or operate on a resource R is decided (or delegated) by an Authorisation Service of the resource owner or service provider at the institution RI.
4. In particular, if U wants to access or operate on resource R, his digital identity has to be trusted by the resource owner or service provider in the institution RI.

The generic use case has evolved from the common situation where U wants to access or operate on a resource R. Here are the steps:

| | |
|---------------------|------------|
| Project: | GN2 |
| Deliverable Number: | DJ5.2.2bis |
| Date of Issue: | 24/04/07 |
| EC Contract No.: | 511082 |
| Document Code: | GN2-07-024 |

1. U requests access to R
2. R contacts R-LFA to acquire AuthN data
3. R-LFA prompts the user to give information about his home location
4. R-LFA sends the request to eduGAIN MDS to find the user's home location
5. eduGAIN MDS sends the appropriate response to R-LFA
6. R-LFA contacts the corresponding H-LFA
7. H-LFA redirects U to his HI AuthN service
8. U presents his credentials
9. HI AuthN service sends the AuthN response to H-LFA
10. H-LFA forwards the response to R-LFA
11. R-LFA forwards the response to the R
12. If needed R issues the request for additional attributes through R-LFA
13. R-LFA transforms (if necessary) and sends the request to H-LFA
14. H-LFA sends the request to the HI identity repository
15. H-LFA gets the requested information from the HI
16. H-LFA transforms (if necessary) and sends the response to R-LFA
17. R-LFA forwards the response to the R
18. R makes final decision about the request.

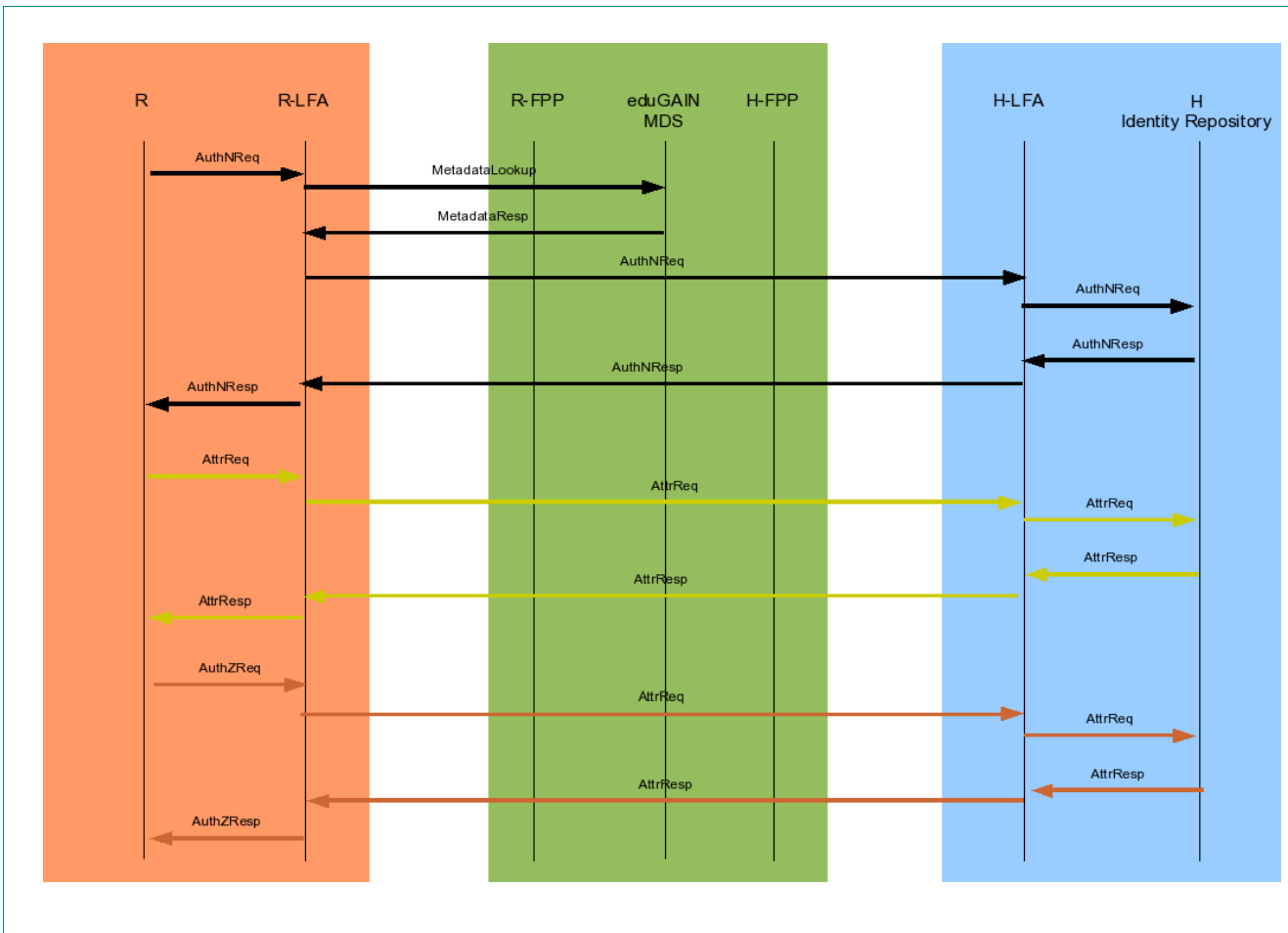


Figure 5-1: The generic eduGAIN use case

We have to point out that the above use case corresponds to the case in which federations using LFAs are established both at home and remote (resource) site. In case where LAs are allowed the traffic via LFA is eliminated. Also, in cases where there is no federation established specific LAs act instead of LFA.

Furthermore in the above case we present the common situation in which the resource R, after successful authentication, issues the attribute request (AttrReq) and makes the authorisation decision based on the values of the attributes received from the HI (AttrResp).

Authorisation decision can be left to the HI. In that case R will issue the authorisation request (AuthZReq) and react upon the received information (AuthZResp). In the most complicated situation R issues both the attribute request and authorisation request as described in section 3.1.

For better understanding of the whole architecture and its use we present its impact to the way the services are used and maintained:

| | |
|---------------------|------------|
| Project: | GN2 |
| Deliverable Number: | DJ5.2.2bis |
| Date of Issue: | 24/04/07 |
| EC Contract No.: | 511082 |
| Document Code: | GN2-07-024 |

- **From the end user's perspective**
 - There is no change in the way user presents their credentials or uses services local to their AAI.
 - There are no additional tools that must be used or installed at the user's machine.
 - The scope of the available services has been extended to the borders of the eduGAIN.
 - Some of the services available via eduGAIN may have different use policies, authentication and authorisation rules.

- **From the resource owner's / home institution's perspective**
 - There are no additional tools that must be used or installed to become a member of eduGAIN
 - For a resource owner: a user base may be enlarged to the borders of the eduGAIN
 - For a home institution: a request for information can come through the local AAI from AAI inside the eduGAIN
 - If a resource owner or home institution wishes to communicate directly with the other element inside eduGAIN (i.e. another resource owner or home institution in different AAI) the proper local adaptor (LA) must be developed, installed and maintained.

- **From the AAI (federation) administrator's perspective**
 - The proper local federation adaptor (LFA) must be developed, installed and maintained.
 - The federation peering point (FPP) must be installed and maintained as part of the eduGAIN infrastructure.

6 Security and Privacy Considerations

This section looks into security and privacy aspects to be considered for the authentication and authorization infrastructure eduGAIN. Security and privacy in an AAI relies partly on the protocols and technologies used within the AAI, but mainly it is governed by the AAI rules and policies, to which all participants have to agree and which they implement in organizational procedures they follow.

6.1 General Considerations

The primary role of eduGAIN is to provide an infrastructure to request, transfer and receive assertions about user authentication and authorizations. That includes information provided by the Home Institution to the Resource describing certain aspects of the authenticated user.

The processing of such assertions has to satisfy security goals as listed in chapter 2 of [BCP72] and privacy concerns. If the assertions carry information that relate to an identified or identifiable natural person, provisions of EU Data Protection Directive has to be taken into account

Since eduGAIN is designed to be Shibboleth compatible and Shibboleth is in principle a set of SAML 1.1 profiles, the general security and privacy considerations for SAML apply to Shibboleth and eduGAIN as well (see [ShibArch]). The document [SAMLSecure] covers in detail these aspects.

6.2 Bridging Elements play a special Role

Under security and privacy aspects, especially the Bridging Elements (BE, introduced in section 2.1) are of interest. These entities are in charge of the trust links between AAI components and user applications and they adapt syntax, semantics and procedures between infrastructures and individual sites. Seen from eduGAIN, a BE is the primary entity acting, hiding partially details of the components on which behalf it acts.

So participants in eduGAIN communicate with BEs operated by third parties. Each participant has to rely on the other BEs following the rules and policies of eduGAIN.

| | |
|---------------------|------------|
| Project: | GN2 |
| Deliverable Number: | DJ5.2.2bis |
| Date of Issue: | 24/04/07 |
| EC Contract No.: | 511082 |
| Document Code: | GN2-07-024 |

6.2.1 Rules and Policies of Federations

Provided a BE is a Local Adaptor (LA), it introduces only a limited set of entities into eduGAIN. For them, the BE has to guarantee that they follow the rules and policies of eduGAIN. They all have to adopt the eduGAIN rules and policies. A BE should not act on behalf of multiple entities using a single resource URI. That would prohibit eduGAIN entities to be able to distinguish them and would e.g. prohibit the possibility to provide different sets of attributes of a user to each of them.

If the BE is a Local Federation Adaptor (LFA), it integrates a whole established federation into eduGAIN. Adopting the eduGAIN rules and policies is no more that trivial since a larger set of entities is involved and perhaps not all of them are interested in communication with eduGAIN. The established federation will have to analyze its rules and policies and compare them with the ones of eduGAIN.

Provided the rules and policies of the established federation are stricter than the ones of eduGAIN, its integration into eduGAIN is no problem for eduGAIN. Whether participants of such an established federation are happy with extending it under these conditions remains to be seen. An established federation might only be willing to share less attributes with eduGAIN partners than within their own federation.

However, if the rules and policies of the established federation are looser than the ones of eduGAIN, it will not be possible to fully integrate it into eduGAIN. The BE will only be allowed to map participants into eduGAIN committing to the stricter eduGAIN rules. Further participants of the established federation would not be allowed to communicate with eduGAIN entities.

For established federations, the LFA looks appealing from an architectural point of view since only one entity needs to be introduced into the federation to become eduGAIN compatible. However, looking at the trust, the rules and policies it is based upon, finding exactly matching rules and policies in established federations and eduGAIN seems unlikely.

Based on the previous discussion one concludes:

- If all federations would have matching rules and policies with eduGAIN, one wouldn't need more than one federation.
- Having looser rules in eduGAIN than the established federations might be a problem for the established federations and will result in a less powerful eduGAIN.
- Having stricter rules in eduGAIN means an LFA can only map a subset of its federation into eduGAIN.

6.2.2 End-to-End Security

A BE breaks end-to-end security by adapting syntax, semantics and procedures. Assertion signatures from the communication partners will be checked by the BE and it will apply its own signature to the newly generated assertion. The two communicating entities have to fully trust the BE to behave properly.

6.3 Credentials and Third Parties

The **AuthenticationRequest** has an optional parameter **Credentials** (see 3.2.1.1), which allows a user to send their credentials back to the own Home Institution. Passing credentials via the network needs always attention, but passing them via third parties and not directly end-to-end is most critical. Therefore, in eduGAIN credentials, which are not delivered end-to-end, **MUST** be properly protected against any potential misuse. A third party shall never be able to gain knowledge from the stream of bits, which represents the protected credentials.

6.4 Attribute Release

Attributes about a user may be released as result parameter in **AuthenticationResponse** or in **AttributeResponse**. Whether attributes may be released to a Resource or not is up to the Home Institution, the user himself as well as data protection regulations which have to be taken into account.

Before a Home Institution releases attributes identifying an end user, he must be informed about which attributes are to be released, to whom, and for which purposes (Data protection directive, article 11). The end user consent for attribute release may also be necessary (article 7). These actions require interaction between the end user and his home institution either beforehand or, at the latest, at the time of attribute release. The architecture should make this possible.

| | |
|---------------------|------------|
| Project: | GN2 |
| Deliverable Number: | DJ5.2.2bis |
| Date of Issue: | 24/04/07 |
| EC Contract No.: | 511082 |
| Document Code: | GN2-07-024 |

7 Conclusions

This architecture definition covers all cases we currently see as a real scenario. The components and their interactions are described using a standardised syntax including all relevant attributes. This forms the basis of the next step, the implementation of eduGAIN as a prototype. A testbed will be set-up in the third project year to start functional tests of the implementation. Parts of the software will be provided for a first use in the performance-monitoring environment (JRA1 in GÉANT2). Feedback and experiences with the installation lead to this new version of the architecture document and conveyed into the second edition of the AAI cookbook the will be provided soon.

In this deliverable the formal description is still based on SAML1.1, a standard from 2003. We are aware of the fact that SAML2.0 is already specified, however for practical reasons (missing publicly usable implementation) we had to stay with the older version for now. It is planned to adapt the protocol definition when an implementation and test environment will be available.

| | |
|---------------------|------------|
| Project: | GN2 |
| Deliverable Number: | DJ5.2.2bis |
| Date of Issue: | 24/04/07 |
| EC Contract No.: | 511082 |
| Document Code: | GN2-07-024 |

8 References

- [ASFlows]** The A-Select Team. A-Select Functional Flows, June 2005
http://a-select.surfnet.nl/functional_flows.html
- [BCP72]** Rescorla, E. and B. Korver. Guidelines for Writing RFC Text on Security Considerations. BCP 72 RFC3552, July 2003.
<ftp://ftp.rfc-editor.org/in-notes/bcp/bcp72.txt>
- [DJ511,2]** JRA5 Glossary of Terms, GÉANT2 project deliverable DJ5.1.1,2
<http://intranet.geant2.net/server/show/conMediaFile.6254>
- [DJ521]** T. Wiberg, D. Lopez, M. Milinovic, J. Rauschenbach, K. Wierenga et al. Documentation on AAI Requirements, GÉANT2 project deliverable DJ5.2.1
- [DJ5.2.3]** Best Practice Guide – AAI Cookbook
http://www.geant2.net/upload/pdf/GN2-06-236v5-DJ5-2-3-1_Best_Practice_Guide-AAI_Cookbook_First_Edition.pdf
- [PAPIDef]** The PAPI Development Team. A Detailed Description of the PAPI Protocol. July 2005.
http://papi.rediris.es/doc/PAPI_Protocol_Detailed.pdf
- [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>
- [SAMLMD]** S. Cantor et al. 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>
- [SAMLSecure]** E. Maler et al. Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML). OASIS, September 2003.
<http://www.oasis-open.org/committees/download.php/3404/oasis-sstc-saml-sec-consider-1.1.pdf>

| | |
|---------------------|------------|
| Project: | GN2 |
| Deliverable Number: | DJ5.2.2bis |
| Date of Issue: | 24/04/07 |
| EC Contract No.: | 511082 |
| Document Code: | GN2-07-024 |

- [SCHAC]** TF-EMC2. SCHAC Attribute definitions for Individual Data.
<http://www.terena.org/activities/tf-emc2/docs/schac/schac-schema-IAD-1.3.0.pdf>
- [ShibArch]** S. Cantor (editor). Shibboleth Architecture, Working Draft 09, February 2005.
<http://shibboleth.internet2.edu/docs/draft-mace-shibboleth-arch-protocols-latest.pdf>
- [SOAP122]** SOAP Version 1.2 Part 2: Adjuncts (W3C Recommendation 24 June 2003),
<http://www.w3.org/TR/2003/REC-soap12-part2-20030624/>

9 Acronyms

In JRA5 used acronyms can be found in the JRA5 Glossary of Terms [DJ5.1.1,2]. Often used terms are listed below.

| | |
|---------------|--|
| AA | Authentication and Authorisation |
| AAI | Authentication and Authorisation Infrastructure |
| AR | Attribute Requester |
| AAu | Attribute Authority |
| AuthN | Authentication |
| AuthZ | Authorisation |
| BE | Bridging Element (includes LFA and LA) |
| CA | Certification Authority |
| FPP | Federation Peering Point |
| H-BE | Home Bridging Element |
| HI | Home Institution |
| HL | Home Locator |
| HO | Home Organization |
| LA | Local Adaptor |
| LFA | Local Federation Adaptor (formerly known as LFC, Local Federation Connector) |
| MDS | Metadata Service |
| RADIUS | Remote Authentication Dial In User Service |
| R-BE | Remote Bridging Element |
| RI | Remote (Resource) Institution |
| SAML | Security Assertion Markup Language |
| SOA | Service Oriented Architecture |
| SOAP | Simple Object Access Protocol |
| TLS | Transport Layer Security |
| URN | Uniform Resource Name |
| URI | Uniform Resource Identifier |
| WAYF | Where Are You From, Shibboleth-specific MDS-like service |
| XML | eXtensible Markup Language |