03.02.09

Deliverable DJ5.4.1,2: Advanced Technologies Overview, Second Edition

Deliverable DJ5.4.1,2

Contractual Date:	30/09/08
Actual Date:	03/02/09
Contract Number:	511082
Instrument type:	Integrated Infrastructure Initiative (I3)
Activity:	JRA5
Work Item:	WI13
Nature of Deliverable:	R
Dissemination Level	PU
Lead Partner	DFN
Document Code	GN2-08-243

Authors: T. Lenggenhager (SWITCH), S. Winter (RESTENA), T. Wolniewicz (UMK), D. Lopez (RedIRIS), S. Neinert (USTUTT), J. Rauschenbach (DFN), A. Solberg (UNINETT), I. Thomson (DANTE), JRA5

Abstract

This Deliverable assesses the different advanced technologies investigated to determine their actual or potential usefulness to the Roaming and AAI activities, both in the remainder of GÉANT2 and potentially for the middleware of GÉANT3.

Table of Contents

Table of Figures

0 Executive Summary

DJ5.4.1 “Advanced Technologies Overview” provided a review of the then-recent technical developments in the JRA5 area that were not yet part of the project plan. The intention was not only to provide an overview of these items but also to evaluate them with regard to their applicability to the remaining portion of the GÉANT2 project.

This second edition of the “Advanced Technologies Overview” is based on issues arising from practical experience (service or pilot service) and future extension plans. The evaluation will also cover the advances’ usefulness in the upcoming GÉANT3 project as well as for the remainder of GÉANT2.

The review covers:

- The problem of preserving user privacy in eduroam, how it is currently handled and what potential solutions can be implemented.
- The problems concerned with the internationalisation of user names in eduroam (for example using language-specific characters such as ü or ê in user names), how it is currently mitigated and potential solutions.
- The use of dynamic server discovery with RadSec in eduroam, and what issues need to be pursued.
- A model which will give a user a means to construct services according to their needs (composable services).
- Metadata service enhancements in eduGAIN.
- Issues relating to level of assurance (LoA) and single log-out in Single Sign-On (SSO) solutions.

1 Introduction

There were three major infrastructure and service related developments in Joint Research Activity 5 (JRA5) of GN2:

- Service level infrastructure (eduroam). Note that as far as eduroam is concerned, operational items belong to Service Activity 5 (SA5) of GN2, while research-related elements remain in JRA5's jurisdiction.
- Pilot infrastructure; eduGAIN (which may become a service in GN3).
- Experimental infrastructure for unified Single Sign-On (uSSO).

The first edition of this deliverable (DJ5.4.1, GN2-07-142v3 "Advanced Technologies Overview") described technology developments for the above areas, some of which are now integrated in the infrastructure (RadSec in eduroam, Shibboleth 2 and SAML 2 in eduGAIN, and NAS-SAML and RadSec in uSSO). Most of the other developments described in DJ5.4.1 may not be implemented but are still of interest, and may prove valuable in GN3.

However, there are a number of new ideas and recently specified extensions, mainly based on practical experiences or recently detected shortcomings, which are discussed in this deliverable. Some of these are currently part of the JRA5 project for the remainder of GN2, while others are suggested to be progressed in GN3.

Finally, it is worth noting that although this deliverable focuses on the usefulness of these developments for the JRA5 activity, the items described could easily be useful to those activities in the middleware area of GN3.

2 Roaming

2.1 Persistent Privacy-Preserving User Identification in eduroam

2.1.1 Problem Description

The eduroam trust model allows the user to be anonymous to the service provider (SP). This preserves the privacy of the user, but as described in the use-cases below there are reasons for wanting persistent user identities. Therefore, the problem is how to provide persistent user identification in eduroam while also preserving privacy.

The current eduroam trust model enables privacy by allowing anonymous use of the service at the service provider (SP) side. A user can obfuscate their actual identity by supplying a pseudonym or empty string as their identity's local component. This is possible because EAP authentication can be divided between the so-called “outer identity” (used for request routing purposes, which only needs a valid realm component) and the so-called “inner identity” (which needs to contain the exact and correct user identification, but is only visible to the identity provider (IdP) during authentication).

Note that this privacy mechanism is not available to wireless roaming consortia that are not based on IEEE 802.1X and EAP. This gives eduroam an advantage over these services.

Example:

A user's true identity is “william.gates@restena.lu”. For their eduroam login, the outer identity “@restena.lu”, is used, leaving the local part of the login blank. The SP will see the RADIUS User-Name is “@restena.lu” and can subsequently route the request through the eduroam RADIUS hierarchy.

The SP does not see the true identity of the user because this ID is only transmitted inside the TLS tunnel that the user establishes to his IdP.

The IdP can see the true identity and verify the user's credentials with this information:

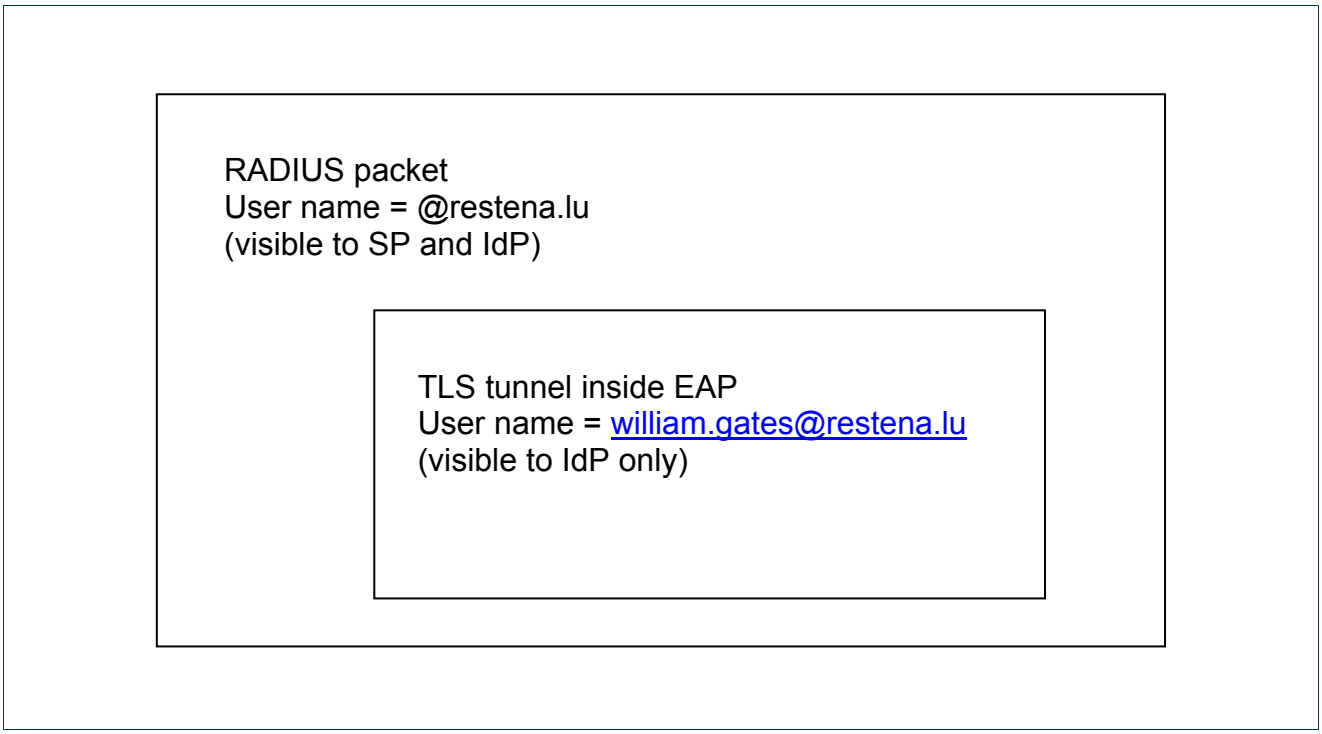

Figure 2.1: Different outer identity and inner identity

In some cases it may be desirable for a service provider to uniquely identify the same user persistently across sessions. On the other hand, privacy of the user should not be neglected.

The following use cases give further examples of the problem:

Use case 1: Timely reaction to network incidents

When an incident requiring immediate action is observed, the SP eduroam administrator has to block a particular user from accessing the network. Starting ordinary eduroam procedures and blocking the user at the IdP side will, most likely, take too much time. Hence the only available option may be blocking the entire IdP realm, which in certain situations may cause problems for large numbers of users, and sometimes (for example during conferences) may be unacceptable.

Use case 2: Reaction to minor incidents

Some violations of SP local rules may be questionable to the IdP as to their cause or reason. The IdP has no means to block a user in a context of a given SP. If the nature of the incident is unknown or vague, the decision to block the user from using eduroam as a whole may seem too harsh. In such cases it would be desirable to provide the SP with the potential to block a user locally.

Use case 3. Identifying and reacting to the overuse of guest access

Users could use institutional wireless networks to build permanent Internet links from home, which may be against the local regulations of the institution. If such links are created by users affiliated with the institution then

their activity can be monitored and action taken; for example, if data transfers exceed acceptable values then QoS restrictions can be introduced. If the institution acts as an eduroam SP users from other eduroam institutions could use its network in this way. However, with anonymous identity and the possibility of frequent change of MAC addresses, the SP may have a problem identifying such infringements and an even bigger problem reacting to them.

Use Case 4. Collecting guest usage statistics at SP

The SP, as the owner of the network, has a legitimate interest in knowing how many guest users are using its network. Using information from the Calling-Station-Id RADIUS attribute, the SP can count the number of devices but not the actual users. Some users may be changing the MAC address of their device, which would add even more confusion to the statistics.

If the statistics indicate to the SP that they should block users, there can be problems.

For example, if the SP blocks the identity “@nerd.com” from using the network, the user in question can change the outer identity at their discretion (for example, into “big@nerd.com”, “annoying@nerd.com”, or even to a different and legitimate user's name (like “stefan@nerd.com”) to avoid the lockout).

An alternative way of blocking would be to block the MAC address of the user's device. However, MAC addresses can be easily changed on modern networking hardware, so this approach may prove ineffective. A user can always disguise themselves to the SP as a different user by changing the tuple (outer identity, MAC address) to different values simultaneously.

Similarly if the SP wants to count network usage, it can only be done at a “per realm” resolution. Therefore usage policy abuses are impossible to spot.

2.1.2 Current mitigation of the problem

The eduroam service description provides basic and adequate measures to mitigate most abuse cases without the need for deployment of new technology. There are currently two ways to mitigate the problem:

- There is a requirement to keep logs of authentication sessions at the IdP side, created with an exact time source. Using this information, it is possible for the SP to report the abuse (of an anonymous user) to the IdP, which is responsible for the offending realm. The IdP can then identify the user and take adequate measures (for example, disabling the user to prevent future logins). This approach has the drawback that a cross-campus, and possibly cross-federation, communication overhead is generated. This approach does not work in real-time.
- If the above step was not successful (e.g. IdP personnel could not be reached) or takes too long, an alternative is to block all users from an entire realm. Since the offending user can only change the local part of their login (the correct realm information is needed to route the request to the IdP in the first place), the realm is a sufficient indicator to the SP. This approach has the drawback that innocent users from the same realm will also be locked out by this measure. This approach does work in real-time.

2.1.3 Potential solutions

2.1.3.1 *Chargeable User Identity*

[RFC4372] Chargeable User Identity (CUI) is defined as a RADIUS attribute with the semantics for conveying a unique but opaque identification string that is persistently attributed to exactly one user. The semantics of this attribute equate to what is known (in the educational AAI world) as the attribute:

urn:mace:dir:attribute-def:eduPersonTargetedId [eduPerson2008].

Note that in the currently deployed RADIUS hierarchy it is difficult to issue different CUI values per user per eduroam SP (one value of CUI per user can be issued for all eduroam SPs). This conforms to the definition of urn:mace:dir:attribute-def:eduPersonTargetedId when considering the whole of eduroam SPs being one cohesive “group of service providers”. However, it is potentially possible to send different CUI attributes to different eduroam Service Providers by generating a unique CUI value based on (not trustworthy) Network Access Server (NAS) identity indications.

The underlying concept of using this attribute is that a NAS (Access Point or Switch) on the SP side requests a chargeable user identity when it sends the Access-Request towards the IdP. The IdP then recognises this attribute request and returns the attribute. The NAS subsequently uses the opaque handle for all future communication about this user session (for example in RADIUS accounting messages).

An initial trial of CUI showed several technical hurdles to its use:

- There is almost no support for the CUI attribute in typical NASs. This problem could be mitigated by the SP's RADIUS server. It could inject the request for the CUI attribute on behalf of the NAS. After consulting one of the authors of the RFC in question, most deployments of this RFC do not rely on NAS behaviour but inject the CUI request and evaluate the CUI attribute content on the SP-side RADIUS server level.

  Tests performed in GN2-JRA5 (by the Polish team) have shown that CUI injection into Radius messages, both Access-Request and Access-Accept in response to proper CUI requests, can be easily configured in FreeRADIUS. This is quite sufficient for handling severe abuse cases. Proper proxying of requests containing the CUI attribute was also confirmed for a number of popular Radius servers. In order to build accounting support based on CUI, the RADIUS server needs to do the NAS's job of maintaining state of the user session, a stateful session surveillance mechanism would need to be built into an otherwise stateless server.

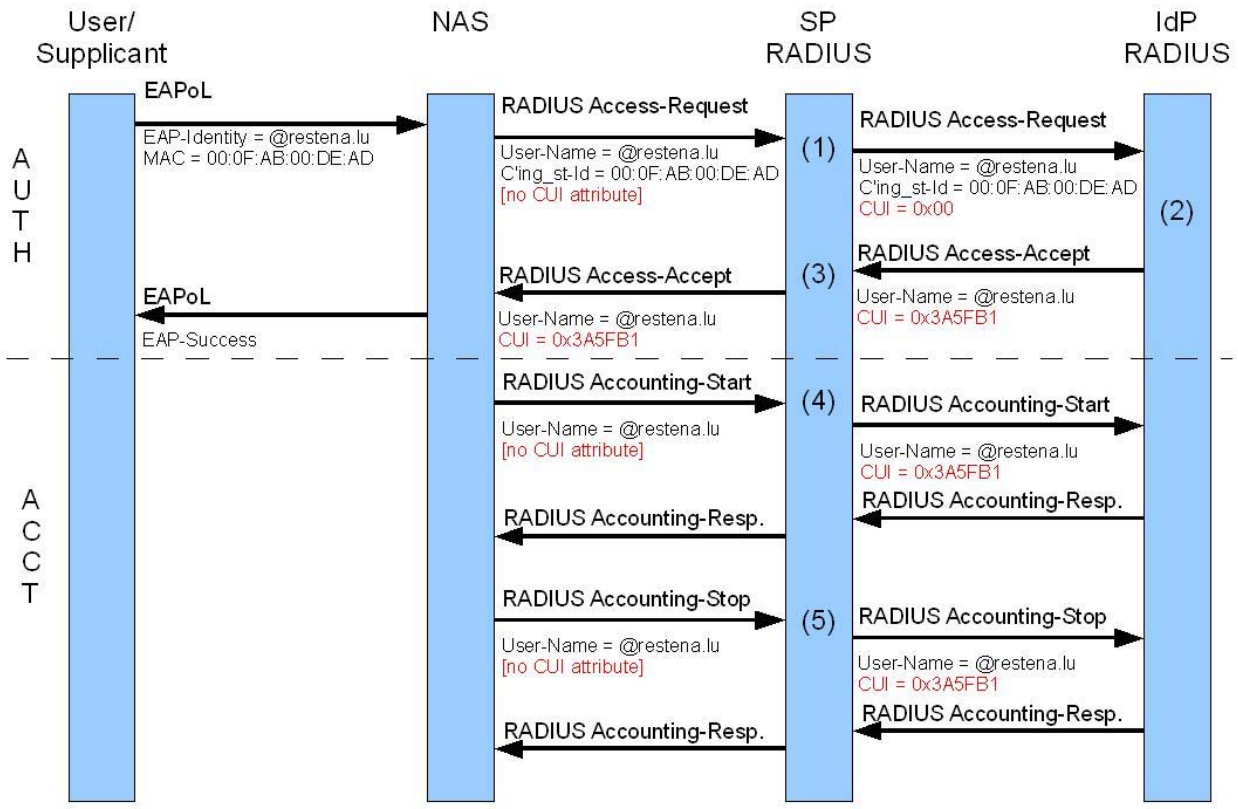

Figure 2.2:: Message flow regarding CUI injection

(1) SP RADIUS adds to local CUI DB: new session, CUI requested from IdP
(2) IdP RADIUS creates CUI value based on *inner* identity, either valid per SP RADIUS or globally
(3) SP RADIUS updates local CUI DB: session has CUI 0x3A5FB1
(4) SP RADIUS retrieves CUI from local CUI DB, adds to Accounting-Request
(5) SP RADIUS retrieves CUI from local CUI DB, adds to Accounting-Request, deletes session from local CUI DB

In order to simplify testing, CUI functionality has been added by JRA5 to the popular eapol_test tool (part of the wpa_supplicant package). This extension is contained in wpa_supplicant since release 0.6.4.

Consultation with the main author of FreeRADIUS led to a possible way forward regarding implementation of this feature by using an SQL backend to store the state information and a set of sophisticated SQL queries during authentication and accounting. Very probably, no actual code changes in the server would be necessary. Investigations regarding this topic are underway in GN2-JRA5.

- Not all IdPs will support CUI immediately. When an SP requests CUI, but the reply lacks the CUI attribute, the RFC document leaves it to the choice of the SP policy to reject or allow the user session.

In eduroam, it must be ensured that the absence of a CUI attribute does not lead to users not being able to log in. I.e. it requires a notice in the next iteration of the eduroam service description to mandate that CUI is a “soft” attribute, which is desirable, but not required.

- It is unknown whether the CUI string can be considered personally identifiable information. In the current eduroam architecture, a user can make their identity and whereabouts known only to the IdP (by altering outer identity and MAC address for every login, as described). While the CUI in itself does not allow an SP to reveal the true identity of the user (it is opaque for the SP), several cooperating SPs might derive a mobility profile for the user because they share the same CUI attribute for the same user. Using different CUI values per eduroam SP may help alleviate the problem of traceability for users. On the other hand, in the typical case users do not change MAC addresses of their mobile devices and hence are exposed to an identical threat to their privacy. It is noteworthy that in a potential future dynamic discovery RadSec setup, it may become possible to use different CUIs per eduroam SP, so that it becomes possible to present different user handles to different eduroam SPs, which eliminates this problem.
- The IdP is responsible for the generation of the content of the CUI attribute. The amount of opaqueness, which goes into this content, is undefined. Some IdPs might simply copy the true inner identity of the user into the SP-visible CUI attribute. In this case, privacy for the user is lost. It is important to make IdPs aware of their responsibility and make them create truly opaque handles for their users.
- CUI values must always be suffixed with the realm they belong to. Otherwise, two IdPs may independently generate the same opaque handle, but the SP might consider them belonging to the same user. Since the SP has no control over the attribute content sent by the IdP, it should autonomously and automatically add the realm to its session database backend that it has received and use the new suffixed value for internal processing.

2.1.3.2 *Overloading User-Name attribute*

Instead of using the CUI attribute, the User-Name attribute could be overloaded.

This could be achieved by sending an opaque value for the attribute User-Name in the Access-Accept message in the answer from the IdP to the SP. The attribute with the new content then traverses the hierarchy up to the NAS, which then ideally uses this new value of User-Name for all later communication related to this session.

Unfortunately, the NAS behaviour regarding which instance of the User-Name attribute is used varies between vendors. Some equipment uses the initial User-Name to report session information instead of using the returned one. Since it may or may not use the identifier which the IdP sent back, processing of accounting information for the IdP might possibly get confused because of mismatching identifiers.

Secondly, there is no protocol-inherent control over this extra use of the attribute. In particular, there is no way for an SP to signal that it would like a CUI-shaped User-Name attribute back in return, and there is no way for an SP to be sure whether the User-Name returned by the IdP is a CUI (and can thus be reliably used to identify the user permanently) or whether it is just an ordinary User-Name with no special significance.

2.1.3.3 *Enforcing equal inner and outer identity*

Yet another approach would be to forbid any divergence between inner and outer identity. This would require a MUST NOT requirement in the next iteration of the eduroam service definition. However, there are no technical means to enforce this requirement throughout the infrastructure, as the only entity that can check whether or not the two are identical is the IdP itself. As a consequence, there is no way to audit whether the IdP configuration is correct, and the therefore the process will likely not be reliable.

This solution also completely eradicates the possibility of user privacy in eduroam, since the true inner identity of the user will be revealed to the entire infrastructure. It might subject eduroam to the European Directive on treatment of personally identifiable information. Since user email address is typically used as the user identity, exposure of this attribute could be considered a serious privacy violation.

2.1.3.4 *Transporting eduPersonTargetedId via DAMe*

It is also possible to use the “Deploying Authorisation Mechanisms for eduroam” (DAMe) architecture to convey an eduPersonTargetedId attribute directly between IdP and SP server. Unfortunately, the DAMe toolset is not yet in production use within eduroam and thus doesn't offer a solution in the short term. As the use of DAMe becomes more wide-spread, this option should be reconsidered.

2.1.4 Conclusions

Out of the approaches outlined above, the Chargeable-User-Identity attribute appears to be the best solution. The basic functionality is ready for implementation in the eduroam community and is now present in the production eduroam service of the Nicolaus Copernicus University in Poland. However, it does contain a considerable number of challenges that must be overcome in order to achieve full accounting support. It is suggested to implement the SQL-based session tracking in FreeRADIUS and use it for a small-scale field testing.

According to reports from IETF participants, CUI has seen few to no deployment outside the GSM community. eduroam, as the largest IEEE 802.1X based roaming consortium, would be spearheading this approach

All testing of this new feature will need to be subject to backwards compatibility to ensure that non-CUI-aware IdPs or SPs are not left behind and no service disruption is introduced. For cases where either the SP or IdP do not support CUI, and thus a unique user identification is not possible, it should be noted that the fallback mechanism as described in 2.1.2 “Current mitigation of the problem” can always be used.

2.2 Internationalisation of user names

2.2.1 Problem description

Historically, domain names and e-mail addresses in the internet have been restricted to a small set of characters within the ASCII standard. Domain names form the basis for eduroam realms, and e-mail addresses are typically used as login names for eduroam users.

Recent developments in the IETF (the relevant standardisation body for internet domain names and e-mail) have led to the possibility of using a plethora of new non-ASCII characters for domain names and e-mail addresses.

Even though no institutions have expressed an interest in using non-ASCII realms and login user names in eduroam yet, it can be anticipated that such a request will eventually happen.

In order to use internationalised user identifiers, appropriate support in the networking protocols that comprise eduroam need to be in place: IEEE 802.1X (support in both supplicants and authenticators), EAP, RADIUS, and the various EAP methods that are in use in eduroam.

The situation with regards to support of non-ASCII identifiers turned out to be varied and ambiguous. The analysis below describes how this situation has arisen, and gives examples of possibly observable encodings of the example fictitious user identity **jürgen@müller.lu**. Note that where the encoding uses multi-byte or non-displayable characters, the notation [0x....] for their hexadecimal values is used.

2.2.2 Protocol analysis

An initial analysis of the protocols involved and a practice test, both conducted by RESTENA, revealed that it is currently not possible to use non-ASCII characters for eduroam login names in a predictable, reliable way. Even though most of the individual protocols involved foresee some amount of internationalisation support, the interfaces between the protocols are not properly defined and cannot interoperate with each other sufficiently. A summary of the findings for each of the involved protocols is given below.

2.2.2.1 *RADIUS/RadSec and Diameter*

Both of these protocols carry their authentication payload in attributes. The attribute that contains the user's login name (called “User-Name”) is a string. It is mandatory that this be encoded as Unicode characters in UTF-8 encoding (see [RFC2865], section 5.1). This means of transportation is safe for internationalised user names. However, end-user devices do not directly interface to a RADIUS/RadSec or Diameter infrastructure. User interaction involves the intermediate IEEE 802.1X entity “authenticator”, which in practice is a wireless access point.

Expected encoding of example case: **j[0xC3BC]rgen@m[0xC3BC]ller.lu**

2.2.2.2 *IEEE 802.1X*

Authenticator

The authenticator is in charge of crafting a RADIUS/RadSec/Diameter packet with the input as provided by the IEEE 802.1X supplicant. It is obliged to make a literal copy of the supplicant's identity indication ("EAP-Reponse/Identity") into the RADIUS et. al. User-Name attribute. It is important to note that the authenticator has no influence over the data sent by the supplicant. In particular, if the EAP-Response/Identity is for any reason not encoded in UTF-8, the authenticator has no authority to change the input encoding. Nor has the authenticator any chance of finding out about the encoding of the input in the first place because there is no character set information in the EAP-Response/Identity.

Supplicant

The IEEE 802.1X supplicant's role is to craft EAP packets according to the user's input and send them on the ISO/OSI layer 2 to the authenticator in EAPoL/EAPoW frames. The EAPoL/EAPoW framing does not mandate any character encoding and simply acts as a transparent container for EAP payload.

Possible encodings for our example case are:

j[0xC3BC]rgen@m[0xC3BC]ller.lu (if supplicant uses UTF-8)

j[0xFC]rgen@m[0xFC]ller.lu (if supplicant uses Latin-1 or Windows ANSI encoding)

Note: Many other encodings are possible.

2.2.2.3 *EAP*

EAP Identities

The Extensible Authentication Protocol (EAP) [RFC3748] does not make any restrictions regarding the encoding of EAP identities. As a consequence, when a supplicant crafts an EAP-Response/Identity packet for IEEE 802.1X, it can use an arbitrary encoding. The use of UTF-8 encoding is one of many possibilities. In particular, it is possible to use an encoding which is incompatible with UTF-8. In this case, an authenticator is forced to generate a malformed RADIUS packet; it must copy non-UTF-8 input into a UTF-8-only output field.

This behaviour is an unintended consequence of the protocol design and has the potential to render non-ASCII user names completely unusable.

Network Access Identifiers (NAIs)

In eduroam, user names are not entirely arbitrary but follow the definitions of [RFC4282] about Network Access Identifiers (NAIs). This RFC defines, among others, the realm structuring where the local part of a user name is separated from the realm part with an infix "@".

Unfortunately, this RFC also contains some unintuitive design decisions for NAIs when it comes to internationalisation. One of those decisions is to require that non-ASCII realms are to be converted to an ASCII representation ("punycode"), as in the Domain Name System (DNS). However, there are no practical implementations of RADIUS or supplicants that expand non-ASCII names into their punycode representation.

Possible encodings for example case if punycode expansion is used:

j[0xC3BC]rgen@xn--mller-kva.lu (if expansion result is encoded in UTF-8)

j[0xFC]rgen@xn--mller-kva.lu (if expansion result is encoded in Latin-1 or Windows ANSI)

(many other encodings possible)

With current supplicants, there is also no way for a user to inform the supplicant that a user name input string is meant to be interpreted as a NAI – the presence of an @ sign is a hint, but does not automatically imply that the input is meant to be a NAI. Consequently, it is up to the supplicant to decide which encoding to use for the realm.

Summarising the state about request routing in IEEE 802.1X, it needs to be stressed that the user's (outer) identity that is used for request routing is indeterminate because it depends on local settings on the client's hardware, and can change across Operating System versions, Firmware revisions, and locale settings. That makes it hard to foresee which literal realm actually is sent on the wire, making it at least questionable which realm(s) need to be configured on the IdP and the home federation side to handle the user-input realm correctly.

2.2.2.4 *EAP methods*

Individual EAP methods have their own means of identifying the user. I.e.: the EAP-Response/Identity and the RADIUS User-Name fields are required for routing an authentication request to the IdP server and usually have no significance in the authentication process itself. The requirements for EAP methods do not mandate UTF-8 support.

For the typical EAP methods in use in eduroam (EAP-TTLS, PEAP-MSCHAPv2, EAP-TLS), the actual credentials are transmitted in a TLS tunnel directly from the supplicant to the home IdP server and are guaranteed to be integrity-protected in the transmission process between those two. As a consequence, the only two parties needing to agree on a character encoding are the supplicant and the IdP RADIUS server. The encodings for EAP methods are defined per method.

EAP-TLS

EAP-TLS has an identity indication within a X.509 client certificate in either the Subject or SubjectAltName fields. As per [RFC5280] section 7.1, these fields can be encoded as a UTF-8 string, and it appears that if a different encoding is used, there is a signalling mechanism which encoding was chosen. As a consequence, transmitting and if needed transcribing between encoded strings should be possible protocol-wise.

EAP-TTLS

The EAP-TTLS protocol uses a set of Diameter attributes within the TLS tunnel to communicate the user's identity. Since Diameter string attributes are per definition UTF-8 encoded strings, transmission of non-ASCII characters is no problem.

PEAP-MSCHAPv2

The MSCHAPv2 protocol [RFC2759] defines the user name field to be in ASCII (see chapter 4 of this RFC). There is no support in the protocol for non-ASCII characters.

2.2.3 Future actions

The issue of incoherent support for international character sets has been brought to the attention of the appropriate working groups within the IETF. An action plan to remedy the situation has been proposed and consists of:

- Update of RFC4282 (NAI) to not recommend punycode conversion.
- Update of RFC3748 (EAP) to mandate UTF-8 encoding of usernames, passwords.
- Update of RFC2759 (MSCHAPv2) to force UTF-8 username encoding.

After these updates, vendors and implementers need to be encouraged to implement the changes. Apparently, after uncovering the internationalisation issues there was a broad support from deployers outside of the IETF to perform any needed changes promptly. An initial commitment was given from Microsoft staff to change their EAP and MS-CHAPv2 implementations for UTF-8 conformity on short notice (possibly even before the RFCs themselves are updated).

2.2.4 Conclusions

The situation regarding internationalised user names in eduroam (and most other instances of authentication protocol deployments) is not looking very good. Right now, the attempt of using internationalised user names or realm names in eduroam needs to be heavily discouraged.

This situation is likely to change as soon as the corresponding specifications and implementations have been updated. The time frame until this has happened appears to be rather large and is certainly not less than a year.

The process within the IETF should be followed closely and, where necessary, be influenced actively to achieve proper results. The amount of influence that can be taken regarding actual field implementations of internationalisation support is limited, but such influence should be sought wherever possible.

2.3 Dynamic server discovery with RadSec

2.3.1 Introduction

As discussed in previous deliverables, the use of RadSec brings numerous technical benefits over RADIUS. Key benefits include reliable transport of authentication messages (reducing the probability of aborted authentications), better mitigation of IP packet fragmentation (which sometimes leads to hard-to-diagnose problems in combination with EAP-TLS), confidentiality improvements (no disclosure of AAA information to anyone but RadSec servers), and the possibility to include ad-hoc service provider locations (authenticating and authorising the ad-hoc SP with an eduroam Service Provider certificate). All these benefits realise with the use of RadSec in a static way (replacing the RADIUS hierarchy 1:1).

There are additional benefits when using RadSec in dynamic discovery mode. These benefits include less disclosure of AAA information to intermediate proxies (with the potential of eliminating intermediate proxies altogether) and shorter authentication times (by reducing the amount of required AAA server processing and forwarding time).

Technical tests have been conducted with one implementation of RadSec (Radiator) near the beginning of the GN2 project already. The recent developments regarding the RadSec specification have yet to be tested though, amongst them dynamic discovery with radsecproxy and interoperability between the various implementation.

2.3.2 Procedure to retrieve IdP information from DNS

There is currently no standard storage model for the realm resolution information. The RadSec internet draft quotes the original Radiator DNS discovery algorithm in a non-normative appendix, which takes the user's realm as an input to a series of DNS lookups. Investigation of this algorithm has shown that the specification contains a few operational challenges.

1. The algorithm is identical to the resolution algorithm of RFC3588 (Diameter). It includes A and AAAA DNS lookups to underscore constructs, which is seen as suboptimal by DNS operational people. At IETF73, it was consequently decided to sanitise the lookup algorithm to use SRV lookups instead of A/AAAA lookups where possible.
2. The algorithm does not take internationalised domain names into account. The topic of internationalised domain names was largely ignored throughout the AAA space so far. At IETF73, it was decided to update the lookup algorithm to properly cater for internationalised domain names.

2.3.3 Field test

A field test of the dynamic discovery features is planned in the final months of the GN2 project.

This test will use a preliminary version of the revised DNS discovery algorithm, which includes a fallback to SRV lookups if previous NAPTR lookups were unsuccessful. The same algorithm will also be submitted for IETF publication. The algorithm is described below:

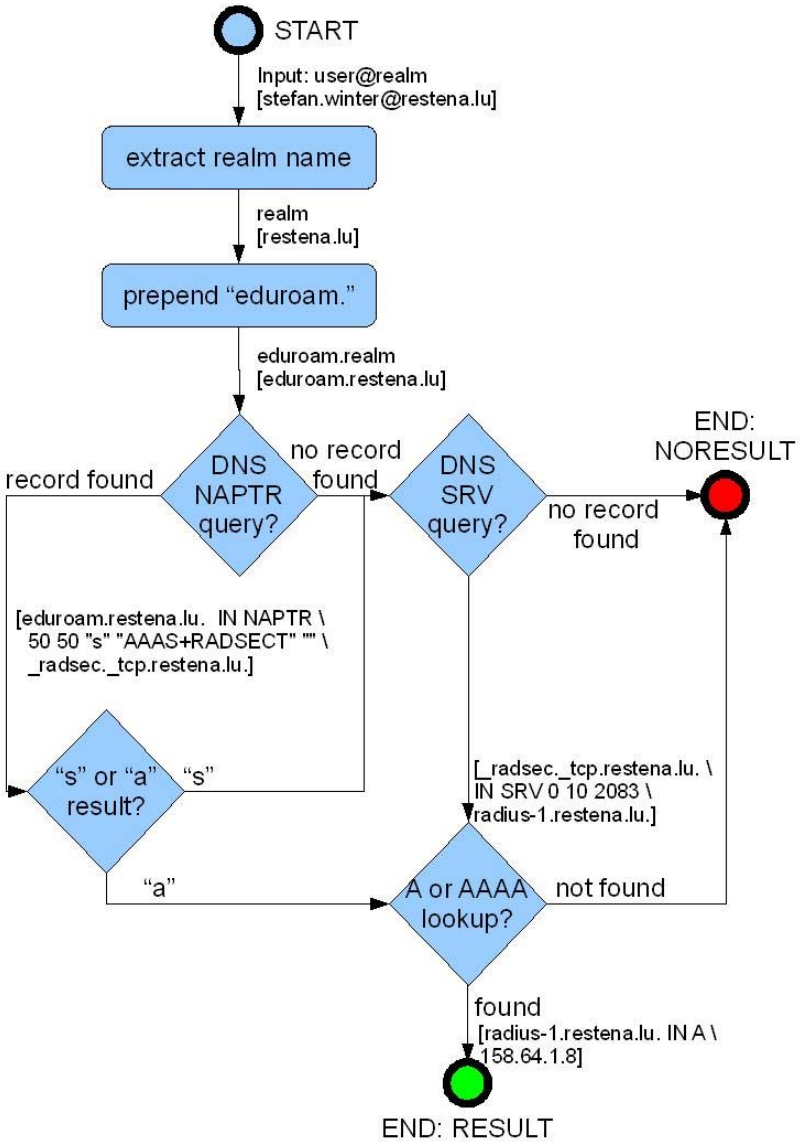

Figure 2.3: Dynamic DNS Discovery

Step	Example
Extract realm from username	stefan.winter@restena.lu → restena.lu
Prepend “eduroam.” to realm name	restena.lu → eduroam.restena.lu
Lookup NAPTR for service AAAS+RADSECT in DNS if found, follow NAPTR lookup chain up to IP address and port of most preferable server for realm. Connect to this server. END.	
Lookup SRV for _radsec._tcp.realm if found, follow SRV results up to IP address and port of most preferable server for realm. Connect to this server. END.	
Give up. END.	

The test should have as little impact as possible on the production operation of the servers that act as test realms. A test scheme has been developed that marks the realms that are to be discovered, especially: where the production username and realm is “foo@example.com”. The parallel dynamically discoverable user name and realm are “foo@example.com.dynamic”. Such user names can only be used at locations which participate in the field test (i.e. on access points that are connected to a RadSec server that is configured to handle these realms with dynamic discovery) because the realm name is not routable on other servers. Participating RADIUS servers only need one extra realm stanza that matches all realms ending in “.dynamic”, leaving the rest of the server configuration untouched.

Upon receipt of a login request of “foo@example.com.dynamic” the server needs to be configured to strip the trailing “.dynamic” and use the remainder of the username as input to the algorithm above.

The servers participating in the field test can thus still service all production realms via the RADIUS hierarchy as normal.

After the field test has ended, production use of dynamic discovery can be enabled globally by replacing the regular expression for „.dynamic“ with an unconditional expression. (i.e. for all realms).

2.3.4 Issues to be investigated

In preparation of an initial deployment of dynamic discovery, several issues have been raised which need to be addressed in the future:

2.3.4.1 *Proxies may be needed to implement federation policy*

Some federations may limit international roaming to a subset of their users. The determination of whether to allow international roaming would usually be implemented on a Federation Level RADIUS Server (FLRS) since this is the proxy that maintains the international uplink connection. When dynamic discovery is in place, and a

SP can directly contact an IdP, such policy decisions become ineffective because the FLRS is not queried during the authentication.

One possible mitigation of this problem is to make dynamic discovery entries always point to an FLRS, and not direct to an IdP. In this case, the FLRS proxy is still in the loop to make policy decisions. On the other hand, mandating a proxy connection somewhat defeats the purpose of introducing dynamic discovery.

Another possible mitigation is to re-examine the federation policy to see whether such restrictions on international roaming are actually (still) needed.

2.3.4.2 *Proxies may be needed to sanitise authentication requests*

There have been several cases where the roaming experience of users was suboptimal because IdPs were sending bogus authorisation information inadvertently to an SP. The SP failed to filter out such information and instead blindly applied the authorisation level to the user session. Specifically, VLAN assignments from the IdP side were sent where it was undue, and SP access points assigned the user to the wrong VLAN. With an FLRS, authentication sessions can be automatically filtered so that such misinformation will not wreak havoc on a user session, even if both the IdP is misconfigured (to send the information) and the SP is misconfigured (to not discard the information). With dynamic discovery, this additional filtering option is eliminated, and the responsibility for the authorisation information rests entirely with the IdP and SP in question.

One possible way to mitigate this problem is proper configuration at SP and IdP locations.

2.3.4.3 *Do direct connections expose locality information about the user?*

In the current proxy hierarchy, IdPs do not receive a lot of information about the whereabouts of their users while they are roaming. The RADIUS attributes NAS-Identifier and NAS-IP-Address may give a somewhat vague view on the whereabouts, but this information can be filtered out within the proxy hierarchy or could be insignificant in the first place (e.g. NAS-IP-Address containing a RFC1918 IP address). When the SP initiates a direct connection to the IdP, the IdP may learn about the whereabouts of its users by examining the source IP address of the incoming connection. This may make it easier to create mobility profiles for a user.

2.3.5 Conclusions

The RadSec dynamic discovery mode is supported at least by the Radiator and the radsecproxy implementations. A small testbed was created to gather some experience with these new features, especially with radsecproxy and the DNS discovery algorithm. A production level usage may be introduced in GN3 after solving the pending policy issues.

3 AAI related technologies - current and future development items

3.1 Composable services

GÉANT2 has laid the groundwork for the generalisation of service confederation to any possible network service, enabling inter-domain collaboration. Future development (as already envisaged by initiatives like IPsphere; http://www.ipsphereforum.org/) will enable seamless collaboration in a global environment by means of a ductile service framework. This framework must be close to the users in terms of direct management, and be able to be scaled up both in width (different communities, different countries, and so on) and depth (to any number of users).

In GÉANT2, several network services have been exposed to users through Web Services interfaces. For example, perfSONAR has provided means for accessing and sharing network measurements, and AutoBAHN has developed mechanisms for controlling bandwidth and circuit allocation. These initiatives have demonstrated that it is possible to provide multi-domain access to network capabilities using open service interfaces.

The logical progression of this work implies the development of a service framework able to consistently allocate individual services, offering them to authorised network users. The availability of such a framework would allow network administrators and user communities to define complex services from already existing services.

This framework implies a set of successive layers of trust links (from campus to regional networks, to NRENs to inter-domain layers) as the only possible solution offering this combination of local control of resources and global scalability.

In the context of this framework, the complex services mentioned above could lead to interfaces that, for example, allow:

- Setting up a lightpath to a data source plus an encrypted multicast connection to distribute the data in several domains, through ad-hoc key distribution,

- Booking a certain amount of storage at several sites plus a dedicated circuit from the source of data to the repositories at a certain time and for a certain interval.
- Establishing an ad-hoc layer 2 VPN including one (or several) eduroam connection(s).
- Retrieving performance data by making the network accept a specifically created probe.

All this clearly requires identity management, but also a uniform messaging framework and registration, and (meta) directory services, plus specific user interfaces.

The Enterprise Service Bus (ESB) is the current service-oriented paradigm for building integrated enterprise systems out of heterogeneous and highly distributed components. This provides an abstraction layer that guarantees the communication among the different participating services without establishing specific connections among them. Services register at the bus and send and receive messages through it. The bus takes care of the appropriate routing and the possible adaptations required.

The idea of providing a consistent service interface to network capabilities in a multi-domain environment has led to a new concept that was originated (to the best of our knowledge) in the GÉANT community: the Multi-Domain ESB. This is an extension of the ESB, deployed in a multi-domain environment, and able to offer an integration layer for services, so they can collaborate and being composed to build more complex, user-driven collaborative services.

In the context of a MD-ESB, identity-related technologies must be considered in the broad sense of identifying and locating all entities associated with the network, ranging from final users to service endpoints. Several identity technologies play a key role in four different and complementary aspects:

- Identifying entities to their peers in any interaction. The specific profiles developed for eduGAIN-enabled services constitute an excellent first step for this.
- Describing entities and providing means to locate them. This implies the ability to route requests to the appropriate interfaces and to establish and verify their properties. Ontologies, publish-subscribe interfaces and data-oriented message routing as offered by OM2 (http://www.openmetadir.org/) seem a valid approach to these tasks.
- Establishing an appropriate framework for service composition from the user perspective, providing authentication, authorisation (and accounting, if required) at the portals and/or tools to be deployed. In this respect, the results of OPUCE (http://www.opuce.tid.es/) provide a very interesting starting point as an intuitive user interface for service composition.
- Providing access, creation and management of identity data as a service by itself, to be integrated with the rest of composable network services available through the composition framework. In this respect, schemas for dynamic service creation and deployment like OSGi (http://www.osgi.org/) and the IPsphere model mentioned above must be considered.

3.2 Metadata Service enhancements

3.2.1 Metadata Signing concept in eduGAIN

3.2.1.1 *Metadata Trust Links and Revocation in eduGAIN*

Metadata constitute the backbone of any identity federation, describing their participants and providing the basic elements onto which to build trust links. In the confederation model provided by eduGAIN, formal metadata definition plays an even more important role since participants from different federations do not share any common a priori knowledge or implicit trust assumptions when they started an interaction. This is why metadata publishing and distribution have been among the main goals of eduGAIN.

In particular, assessing the trust of the metadata to be used to establish a link to a certain eduGAIN component constitutes the key issue when any identity data exchange is about to be initiated. The trust evaluation procedures must be obviously based on the eduGAIN trust fabric, and satisfy requirements in what relates to their scalability and their ability to be integrated with the local federation procedures. Furthermore, two additional properties are highly desirable:

- The possibility of re-distributing and caching, so connections to the central metadata distribution points that constitute the eduGAIN Meta-Data Service (MDS) need not to be in real time. This way, the confederation system does not rely on a single point (or set of points) of failure.
- Simple and immediate revocation of published metadata, so responses to security incidents are swiftly propagated.

The following sections briefly present the aspects of eduGAIN metadata structure that are relevant to trust evaluation, and introduce two approaches for these procedures. The first one is to be applied in version 1 of the eduGAIN infrastructure (due for imminent release at the time of writing). The second is to be introduced during the eduGAIN pilot phase, and is intended to allow the application of the new proposals on dynamic metadata.

3.2.1.2 *General eduGAIN Metadata Structure*

Per each participating federation, one or several Federation Peering Points (FPP) are allowed to upload metadata to the MDS, according to the federation's and eduGAIN policies. According to the eduGAIN trust structure, each one of these FPPs has a valid eduGAIN certificate issued by a CA accredited to the eduGAIN Truststore.

Each one of these FPPs manages an `EntitiesDescriptor` element inside the whole eduGAIN metadata. Inside this descriptor, an `EntityDescriptor` element describes each one of the eduGAIN components under the FPP responsibility. Components submit their `EntityDescriptor` elements to their FPP, which validates them according to its publishing policies prior to building a new `EntitiesDescriptor` and uploading it to the MDS.

Each `EntityDescriptor` provides all the required elements for establishing the trust on it, so they can be individually cached, distributed and re-used according to local rules. This implies that an `EntityDescriptor` has to carry a digital signature, verifiable using the eduGAIN trust fabric.

Furthermore, every eduGAIN `EntityDescriptor` MUST contain a digital certificate that will be used to evaluate trust in the interactions of the corresponding component with the rest of the eduGAIN infrastructure. In order to simplify crypto-evaluation at the individual components, and to allow for the usage of eduGAIN metadata in other infrastructures, it is common practice that this certificate is self-signed.

3.2.1.3 *Basic Metadata Signature (eduGAIN 1.0)*

Each `EntityDescriptor` managed by an FPP is signed with the FPP's eduGAIN private key. When a metadata element is read, its integrity can be established by verifying the digital signature of the elements. An element will yield a correct verification result whenever:

- Its digital signature verifies.
- The FPP key has not been revoked and included in the eduGAIN Truststore revocation list.

This implies that the only ways to revoke a certain metadata element are:

- Make it delete by the FPP at the MDS
- Revoke the FPP key, thus revoking not only the element but the whole metadata set it is part of.

And, moreover, this means that the MDS acts as the only trusted source possible of metadata and that a component downloading them must always use TLS and check MDS certificates.

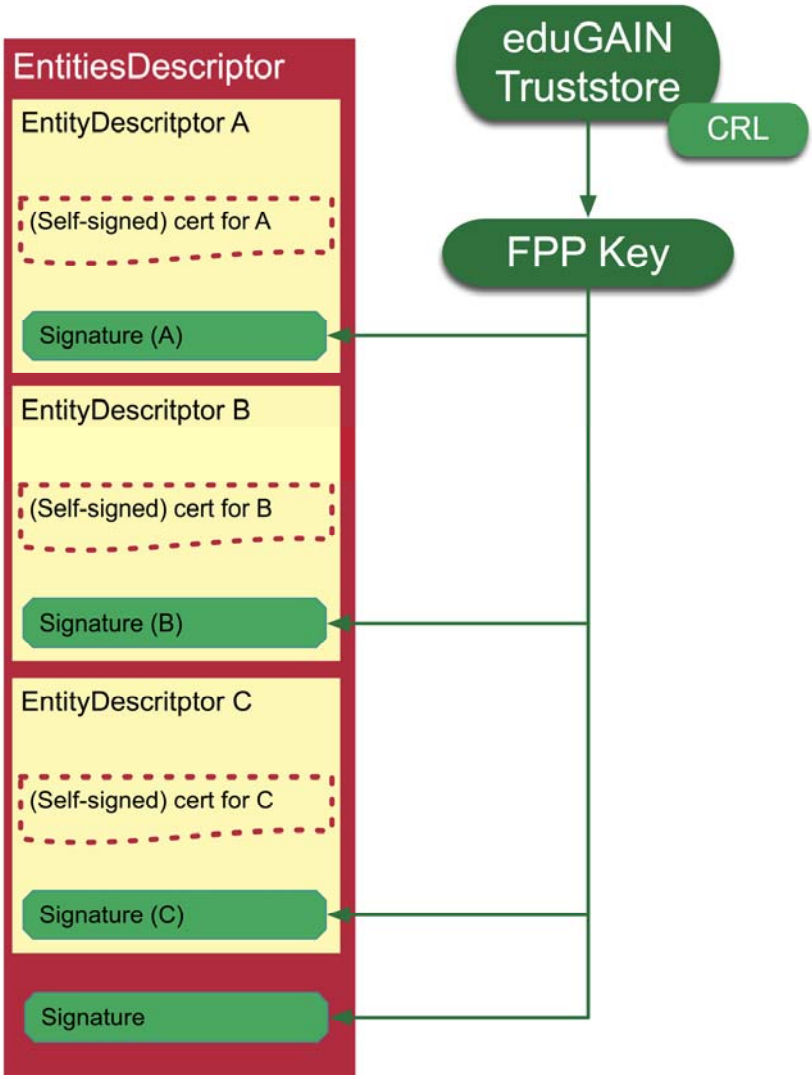

Figure 3.1Metadata signature construction

3.2.1.4 *Independently Revocable Metadata*

Independent metadata revocation requires the infrastructure being able to individually invalidate a certain signature without requesting the incumbent FPP to contact the MDS or killing its own trust by revoking its signing key.

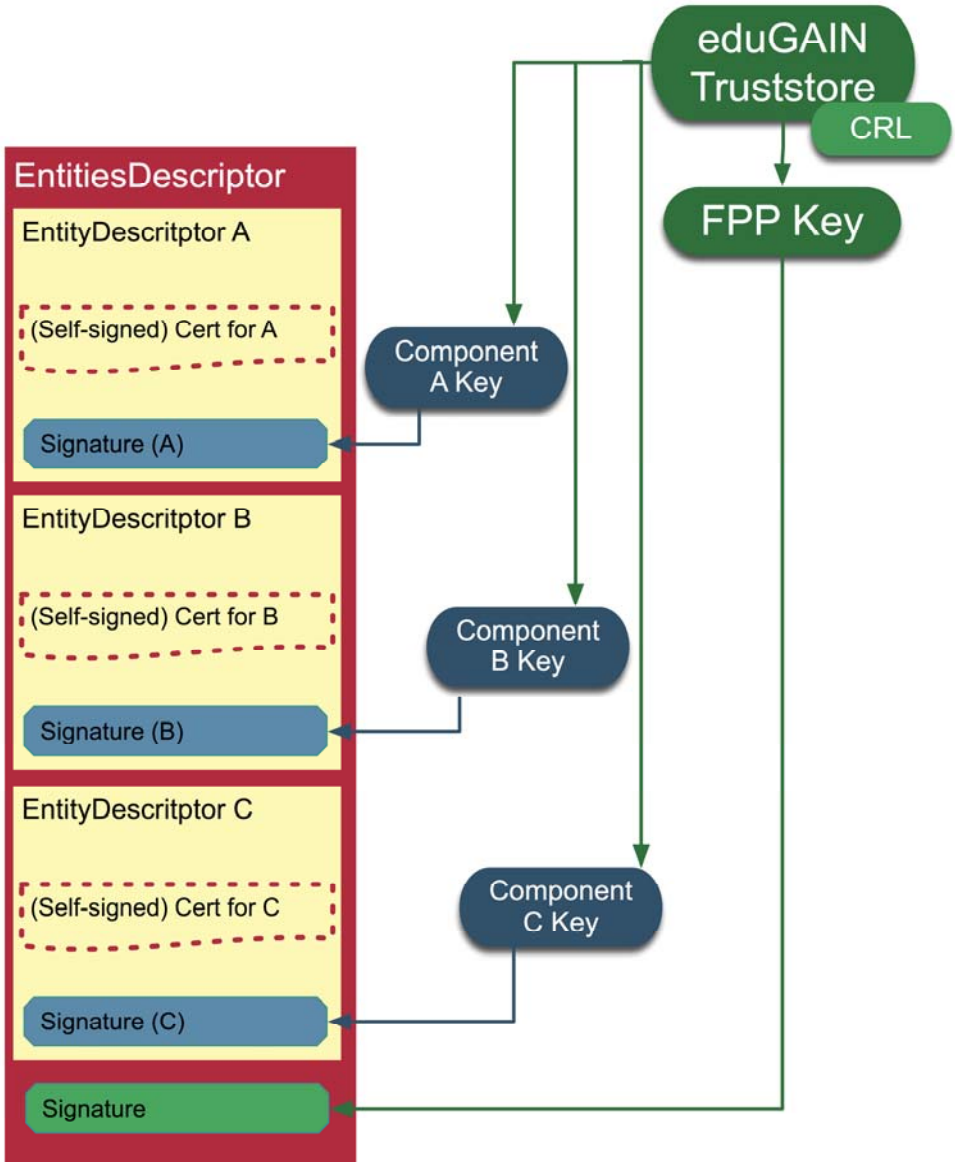

Figure 3.2 Signature key

The concept is that the normal signature verification process will produce evidence that the metadata element is currently considered valid. For this purpose, the simplest approach is to use the same revocation procedures already provided by public key technology and used in the verification of the metadata trust chain.

The only additional requirement to allow this implicit revocation checking is that each eduGAIN component must obtain a valid eduGAIN certificate and must use it for signing the `EntityDescriptor` describing the component. This certificate might be included in the `EntityDescriptor` as well, but it is not required as its main purpose is the signature of the `EntityDescriptor`. The FPP can (and should) make additional checks on the metadata being published, evaluating the association of a certain piece of metadata with the signing certificate in accordance with the federation's own signing policies.

To revoke a certain `EntityDescriptor`, an authorised party simply revokes the certificate for the specific metadata element. To keep metadata control in the hands of the FPP, it is necessary to allow that the revocation procedure can be started by either:

- The eduGAIN component the `EntityDescriptor` refers to, as the requester of the certificate.
- The FPP through which the `EntityDescriptor` is published, as responsible for the federation signing policies.
- The issuing CA of the certificate, as part of the eduGAIN trust fabric.

Whenever a verification procedure is run over a digital signature made with a revoked certificate, it will automatically fail, yielding the metadata element as invalid as it was immediately deleted at the MDS.

This way, the MDS is used for metadata publication and querying (including embedded WAYF support), and probably as last-resort metadata signer, but is relieved of its role of central and unique trusted point in the infrastructure. Metadata can be asynchronously downloaded, distributed by third parties, and cached while revocations can be performed almost in real time.

3.2.2 Metadata tagging

3.2.2.1 *What is a metadata tag?*

The standard SAML2 metadata schema [saml-metadata-2.0-os] allows much extensibility beyond the minimum information necessary for interoperation of SAML2 entities, i.e. IPs and SPs.

Adding a tag to a metadata entry allows that extensibility to be utilised. The UK Access and Management Federation was probably the first federation to use tags in metadata [ukfed-tech-spec]. Here is an example extracted from the UK metadata:

```
<EntityDescriptor entityID="...">
  <Extensions>
    <shibmeta:Scope regexp="false">typekey.iay.org.uk</shibmeta:Scope>
    <UKFederationMember/>
    <wayf:HideFromWAYF/>
    <AccountableUsers/>
  </Extensions>
  ...
</EntityDescriptor>
```

The tag <UKFederationMember/> states that the owner of the entity has agreed to be bound by the UK federation's Rules of Membership. The tag <wayf:HideFromWAYF/> instructs the Discovery Service (DS)

(formerly known as 'Where Are You From Service' (WAYF)) to hide this IdP entry from the list of IdPs the WAYF presents to the user, e.g. for IdP test entries. The tag `<AccountableUsers/>` gets added, when a federation member asserts to the federation operator that a given identity provider entity provides for user accountability.

This real world example illustrates the broad scope for which tags can be useful. On the other hand, such simple tags cannot provide any assurance on correctness by themselves. In particular the <UKFederationMember/> or <AccountableUsers/> tag make that obvious. The only assurance a metadata consumer gets is derived from the digital signature on the metadata entry or group of entries. Generally, the federation coordinator signs the federation metadata. Within a single federation, the assurance of the federation coordinator is probably good enough. The same body collects the metadata and defines the policies and processes. Therefore, trust in its proper action is a prerequisite.

For the inter-federation use case, better scalable assurance is required to allow tags to play a major role in end to end inter-federation IdP to SP interoperation. A metadata consumer has to be able to verify which body asserted a specific tag.

3.2.2.2 *Proposal for a metadata extension suitable for tags*

Based on discussions with the Swiss and the UK federation, in October 2008 the Shibboleth core team of Internet2 began work on a metadata extension suitable for tagging. In the meantime, a first draft was submitted to OASIS, the standards body for SAML [SAML2MetadataAttr].

The proposed name for this metadata extension is `<mdattr:EntityAttributes>`. It is a wrapper for one or more `<saml:Attribute>` or `<saml:Assertion>` elements [saml-core-2.0-os]. That way, tags are either SAML2 attributes or attributes within assertions. In case of an assertion, the issuer of the SAML2 attribute assertion certifies by its digital signature for eligibility and correctness of its contents. The signer of an assertion tag can be completely independent of the metadata signer.

Using a SAML attribute assertion adds complexity. However, it also allows reuse of already existing SAML2 library code for generating or evaluating such assertions. It is not necessary to invent and implement something new.

The standard for SAML attribute assertions defines the element `<Conditions>` with optional attributes `NotBefore` and `NotOnOrAfter` to specify time constraints for that assertion. The optional elements `<Audience>` and `<AudienceRestriction>` specify the tag's intended scope.

An additional advantage of encoding tags as attributes is their straightforward use inside an IdP or SP for internal policy decisions. So the presence of a certain tag could influence an IdP's decision on which user attributes to release to the relying party. It is expected that the Shibboleth open-source software will be implemented in this way.

To give an example:

Assume an institution offers audit services for IdPs and SPs to certify compliance with some standard or regulation. After a successful audit that institution will issue a tag within a SAML2 attribute assertion with a specific `NotOnOrAfter` value to indicate how long their assertion is valid. The IdP or SP admin will add that tag to the entity's metadata entry to make it known to the relying parties. The administrator is responsible for acquiring a new tag before the old one expires (as is the case for server certificates currently).

3.2.2.3 *Standardising metadata tags*

Once it is known how to encode tags in metadata, it remains to be defined what to express by tags, how long tags should be valid and who is eligible to assert which tag in order to be acceptable for relevant parties. In short, tags should be standardised.

This will be a significant task for inter-federation coordination bodies.

However, it is not limited to these bodies. Anybody could issue tags, it will be up to the parties involved to decide which tags they acquire themselves and which they will accept from others and find trustworthy.

Conformance with federation policies, assurance profiles, and data protection regulations are the most likely candidates for standardised tags.

A further category of tags could carry information on which type of parties to accept for interoperation. For example, an SP could use a tag to state that it only accepts IdPs from a certain set of countries or business categories. Such information could allow the SP to extract only the acceptable entities from a metadata file.

For GN3, eduGAIN has to define which metadata tags to adopt or require in the upcoming production service.

3.2.3 Distributed Metadata

One of the main tasks for cross-federation interoperability between SPs and IdPs is to distribute metadata about all necessary entities. The metadata documents include information about which entities to trust, which means metadata are tightly coupled with trust management.

3.2.3.1 *Metadata distribution today*

Distributing metadata between federations can be done in several ways. Most of them are based upon a centralised storage of metadata, where entities can pull down metadata for the whole set of entities. Trust is managed by transport over HTTPS and/or signature on the metadata document itself. The signature may be based on a full-complex PKI, where several partners may perform the signing, or it can be only one key-pair, where the metadata repository is the same entity signing the metadata.

One issue is that if all entities rely on a central metadata repository, this may become a performance bottleneck during busy periods, and also a likely target for denial of service attacks.

3.2.3.2 *Dynamic SAML*

Dynamic SAML is an alternative (in fact, the opposite) to a centralised metadata repository, and was described in a recent draft publication (http://rnd.feide.no/content/dynamic-saml-20-metadata-retrieval-simplesamlphp). This functionality enables entities to use an EntityID that matches a URL where metadata can be retrieved via a HTTP(S) GET request. This allows an entity to send an `|AuthNRequest|` to another entity where no metadata has been previously exchanged. At the time an entity receives such a request, metadata may be looked up instantly, and used.

The issue here is that metadata exchange in this fashion is difficult to combine with trust management.

3.2.3.3 *A distributed model by delegation*

An alternative approach to a more distributed metadata exchange could be achieved by letting a metadata repository delegate responsibility for a subset of its entities to an external repository (see Figure 3.3):

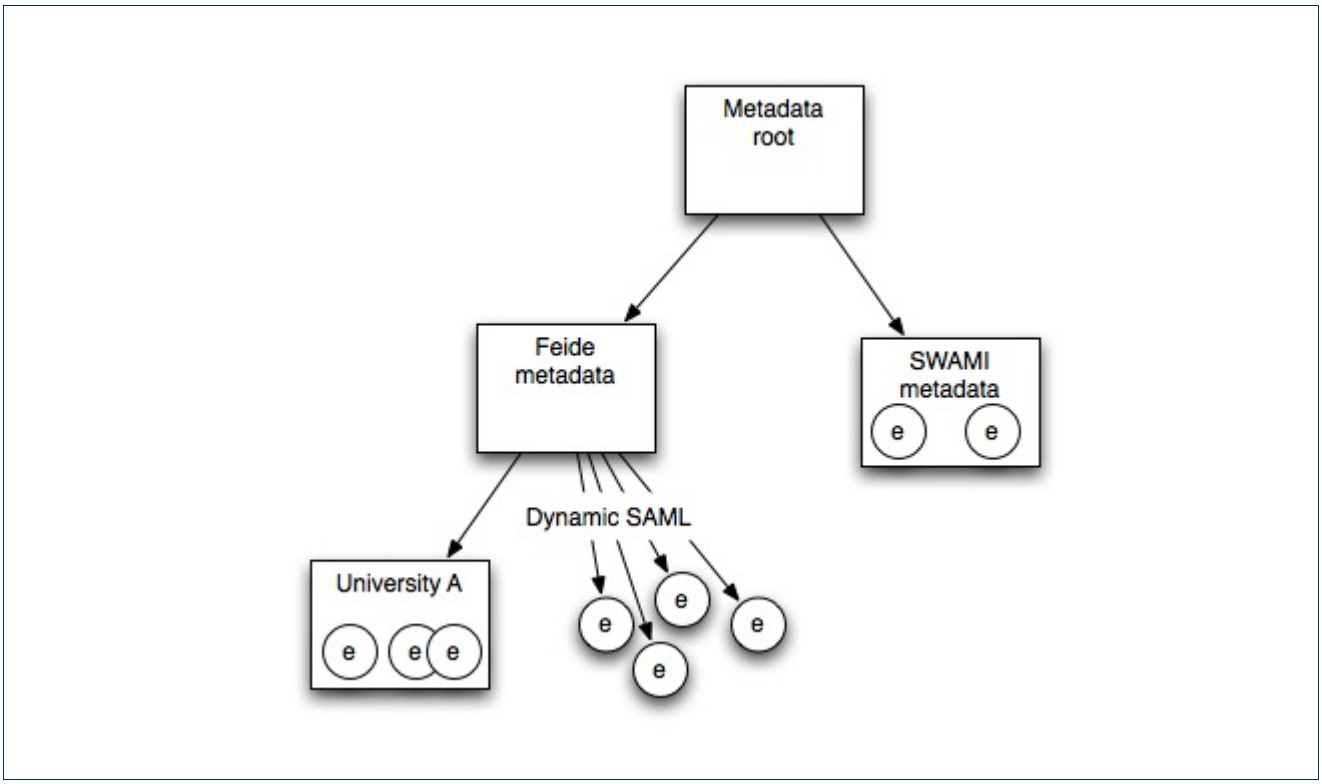

Figure 3.3: Delegation distributed model

In this way a root repository could contain pointers to (for example) national metadata services. Those services could either present an aggregate of all its entities, or point further on (to a university, for example). Pointers could also be to single entities that support Dynamic SAML.

Unfortunately, it is difficult to express such delegation with the current SAML 2.0 Metadata XML schema. One of the issues is that the schema requires at least one entity to be specified in each document.

However, a possible approach could be to define a new schema for delegated SAML 2.0 Metadata that re-uses (as much as possible) the SAML 2.0 Metadata schema; for our purposes, at least the `|EntityDescriptor|` with all it's children.

Although this architecture also is based upon a central root node, it does not suffer from the same reliability problem as today's model. The reason is that when the root node contains more or less static data it can be cached for a long time, and downtime on the root repository would not be as critical.

An important feature of delegation is its coupling with restrictions. For example, when the root node delegates to a metadata repository for Feide (the Norwegian educational federation), the root node may also add some restrictions for all entities referred within that scope. Example of such restrictions could be:

- Limit the referred metadata document to be signed with a specific key (this would work as an alternative to the X.509 PKI, by including explicit public keys in the metadata document instead of performing this out of band).
- Limit the EntityID to match a URN prefix.
- Limit the allowed "realms" to this scope to be *.no. For example, this would put restrictions on released eduPersonPrincipalName.
- Limit the scope to only allow delegation on depth “1”.

3.2.4 Conclusions

The Metadata signing concept described in this chapter is planned to get operational in GN2. From practical aspects measurements for increased resilience of the MDS must be implemented as well. The other outlined concepts (tagging and distributed metadata) are to be discussed in detail in GN3 JRA3.

4 SSO Related Technologies

4.1 Level of Assurance (LoA)

In a federated AAI, users are authenticated before they gain access to a service. They only need to authenticate once, as the AAI supports Single Sign On (SSO) and establishes an SSO context. In many cases a number of alternatives on how an authentication can be performed are available. It is the IdP that chooses one or multiple alternatives of authentication that it wants to offer it's users. Users select one of the available alternatives when they are prompted to authenticate before granting them access to a service at some SP.

The term *Level of Assurance* (LoA) [LoAs] describes the strength of an authentication, which can be derived primarily from the authentication method but also from additional information, such as the IdP who issued the credentials. An IdP that provides multiple ways of authentication can provide different LoAs for its users to choose from. A common example for a lower LoA is authentication by username and password, and an authentication method based on smartcards could enable a higher LoA. From the user's perspective, the criteria for choosing are ease-of-use and price (if not free) but also the services that are expected to become available. A SP can offer many different types of such services; for example they can range from digital, personalised newspapers to electronic administration or banking. Some types of services require higher levels of security and protection than others, because the potential damage that could be caused by a successful attack on those systems means a higher risk for the owner of these services. For example the financial damage in case of unauthorised access to a digital newspaper would far lower than the losses resulting from unauthorised bank transfers.

More specifically, a LoA is a variable that can have one of several defined discrete values (LoA profiles or classes). These can be simple integers, Uniform Resource Identifiers (URIs) or others values. Additionally these values can be ordered on a scale, from lowest to highest (this ordering is not strictly required, but apparently always present). The LoA value is derived from a number of factors:

- The process of registering a digital identity.
- The type of token used for authentication.
- The authentication protocol (at the IdP).
- The communication mechanism between SP and IdP.
- The management of credentials at the IdP.

- The management of attributes of users at the IdP.
- The type of credentials used for the authentication.

A more extensive list of exemplary factors can be found in [IIAP]. Some of the factors are specific to the current (ongoing or recently performed) authentication while others are more static and specific per IdP. More general and also non-technical factors (such as existence of security best practices or experience as an IdP operator) can also affect the LoA a specific IdP can provide. Existing proposals for LoA often use few LoA classes; for example, the InCommon federation [IIAAF] supports three (none, bronze and silver). A federation must define in its agreements and policies what LoAs are part of the common vocabulary of IdPs and SPs, in what (syntactical) form they are represented, what they mean (semantics) and what entity is supposed to perform the assessment. To support LoA in a confederation, either a confederation-wide agreement or a policy describing these is required. Alternatively there could be an agreement on how federation-wide LoAs must be converted so that the result assures the same LoA, or at least not more than the original data.

A middleware for a federated AAI that wants to provide support for LoA must support a mechanism for SP and IdP to communicate the required LoA, the possible LoA and the actual LoA for an authentication. The IdP wants to express the degree of protection of the identities it manages, the SP requires a certain amount of trust in the identity data it bases its authorisation decisions on. If a user is not yet authenticated at all, the SP should state the required LoA so that the user can be authenticated with a corresponding method at an IdP supporting this LoA. If a user is already authenticated and an SSO context is established, an SP needs to decide if the LoA of this context is sufficient and if not, trigger a re-authentication.

4.1.1 LoA in eduGAIN and in DAMe

The federated AAI eduGAIN [DJ5.2.2,2] uses the Security Assertion Markup Language (SAML), therefore the LoA needs to be expressed by some construct inside a SAML Statement. A simple solution would be to express the LoA as a new attribute of the user. However, this is not particularly suitable as a LoA does not only describe a property of the user, but also a property of the authentication of the user. Therefore it may fit better into an AuthenticationStatement instead of an AttributeStatement. To cover user related aspects and aspects of the IdP related procedures, LoA can be handled both ways.

As it can be expected that IdPs offer more than one LoA for a user, the LoA cannot be a static value but needs to correspond to the actual authentication method. SAML 2.0 [SAML2.0] offers an AuthnContext [AuthnContext] element as part of an AuthenticationStatement, to express properties of the authentication (e.g. X.509 certificate, username + Password, etc.). There are several possibilities to express a LoA inside the AuthnContext. For example:

- The existing AuthnContext schemas could be used, or additional new ones defined if necessary. The LoA can be calculated by the IdP, that which would then only send something like “LoA = 1”; or it could be calculated by the SP using the information in AuthnContext as input.
- The new LoA extensions could be added to the deployed AAI middleware as Shibboleth (for example) or to the confederation middleware eduGAIN (which also can be used for single federations). The latter approach is also easier to integrate with the uSSO architecture of DAMe. The remote and the home

Bridging Element (BE) of eduGAIN would need to be extended to understand the LoA and to support a new profile for re-authentication. The metadata can be extended as well, to represent what LoAs a certain IdP supports. This is useful if, for example, a user has accounts at two different IdPs that provide different LoAs, so they are redirected to the one with the appropriate LoA. The SP (or rather its r-BE) will delegate authorisation to another entity, called Policy Decision Point (PDP), which will be able to check if the given LoA is sufficient to reply with an Access Accept, or to require a re-authentication for a specified higher LoA. These policies can be expressed in XACML [XACML1.0].

The unified SSO architecture of DAMe [DAMe] is also based on SAML, and therefore the same considerations on how to represent a LoA are valid as in eduGAIN. Any of the alternatives described above can be used with the SSO token "eduToken", but eduroam is used for the initial authentication, which gives access to another type of resource (the network) that can require a specific LoA. Using the fine-grained authorisation capabilities DAMe added to the eduroam infrastructure, it would also be possible to request a specific LoA for the network access. This would be more complex for end-users, as they currently don't need to reconfigure their supplicant when visiting different locations.

4.1.2 Conclusions

The LoA-enabled infrastructure can neither help in assigning minimum required LoAs to (types of) services within a federation, nor can it help in deciding what LoAs an IdP of a specific middleware in a certain configuration is able or not able to provide. Proposals (e.g. by the UK [UK e-Envoy], the US government [US E-Authentication] and the National Institute of Standards and Technology (NIST) [NIST]) exist that could be used as a basis for discussion, though the influences of confederated setups as well as (u-)SSO features would need to be considered carefully. A guideline concerning calculations of, and requirements for, specific LoAs would be helpful for the operators of the federation, but the infrastructure support should be as generic and flexible as possible to support multiple national "flavours" of LoA.

4.2 Single Log-Out

Logout is conceptually related to web-based login. The natural companion to single sign-on solutions is global logout. Support for global logout was neither included in SAML 1.1 nor in the Shibboleth SAML extension.

As the majority of educational national federations have deployed the Shibboleth architecture, there is no widely adopted standard for global logout in the educational environment. Some federations do not support logout at all, while others implement simple proprietary logout redirect mechanisms that implement a partial global logout.

Global logout was introduced in the Liberty Alliance standard [ID-FF] under the name Single Logout. Later, the SAML 2.0 standard included Single Log-Out.

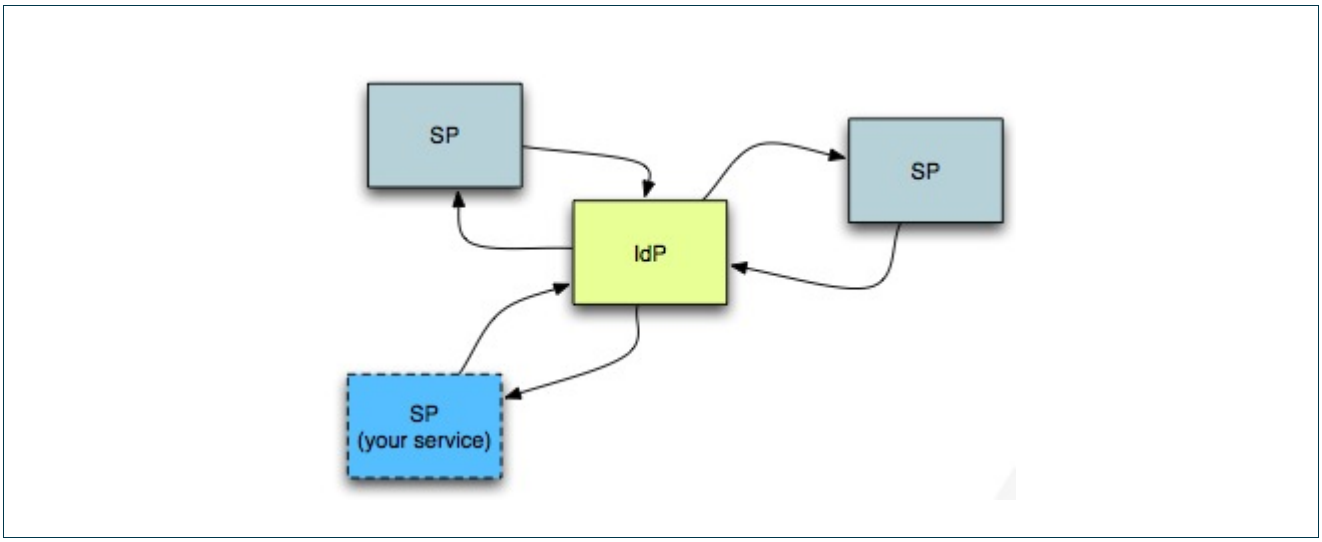

Figure 4.1: Single Log Out

4.2.1 Why not local logout?

When a SP is part of a federation that implements SSO, the user session is extended to include not only the SP, but also the IdP, as well as any other federated service where the user is logged in.

If local logout is implemented, and only the local session is terminated when logout is initiated, the user expects the service to be inaccessible until a new login is performed. With SSO, the user already has a session at the IdP, so if the service is accessed again after logging out, the user is immediately logged in through the SSO functionality. Consequently, local logout makes no sense where SSO is provided: Either, logout cannot be supported at all, or the logout must extend beyond the local session.

4.2.2 Single Log-Out issues

The solution seems obvious, and the standard for how to do it does exist: In the SAML 2.0 protocol suite, it is called the Single Log-Out (Single Log-Out) profile.

A major challenge with SLO concerns informing the user about what is going on. As SLO is not widely deployed, it may be hard for a user to understand that logging out from service A also implies log out from service B.

The SLO profile can be implemented using either front- or back-channel bindings, both having their pros and cons. Front-channel bindings are simple to implement because the SPs have easy access to the session ID in a cookie. Yet, front-channel bindings have a serious shortcoming; when a SP crashes during logout, or is unavailable, the user is already redirected to the SP, and there is no way for the IdP to revert control to the user. Consequently, one SP may interrupt the logout process and prevent the user from logging out from other services.

Front-channel logout is not compatible with profiles other than Web SSO.

Full SLO means that all participating SPs implement it, and do so correctly. SLO, where only a subset of SPs implement it, is even harder to understand for users. In educational federations based on SAML 1.1 or Shibboleth, where basically no service supports SLO, migrating to SAML 2.0 and forcing all SPs to correctly implement SLO may be a challenge.

4.2.3 Solving Single Log-Out issues

Work has been done to handle SLO issues to make it simpler for users and SPs. One of the innovations in the area is the AJAX+iFrame SLO implemented by SimpleSAMLphp.

The AJAX+iFrame approach is based on the HTTP-REDIRECT front-channel binding for SLO protocol messages. The IdP, when receiving a `LogoutRequest` from a SP, presents a web page to the user that includes a hidden iFrame for each of the SPs the user is currently logged into. Each iFrame sends a HTTP-REDIRECT logout request to the service. When the response is sent back to the IdP, the logout state is updated. An AJAX call ensures that the user interface on the web page tells the user the current state of the logout process for each SP. If one or more SP fails to logout, the user is informed about steps necessary to ensure that no sessions remain open.

Figure 4.2: Screenshot from simpleSAMLphp logout

5 Conclusions

Detailed conclusions and recommendations are given within each of the research areas described in this deliverable, but in summary:

- **Persistent Privacy-Preserving User Identification**: The Chargeable-User-Identity attribute appears to be the best solution. The basic functionality is ready for implementation in the eduroam community and is now present in the production eduroam service of the Nicolaus Copernicus University in Poland. See “Conclusion” on page 13.
- **Internationalisation of user names**: There are no obvious answers to this problem. The use of international user names should be discouraged until the corresponding specifications and implementations have been updated. See “Future actions” on page 17 and “Conclusions” on page 17.
- **Dynamic server discovery with RadSec**: This analysis highlighted several issues that need to be addressed in the future, including:
  - Proxies may be needed to implement federation policy.
  - Proxies may be needed to sanitise authentication requests
  - Do direct connections expose locality information about the user?
- **Composable service**: Several network services have been released to users through Web Services interfaces. The logical progression of this work implies the development of a service framework able to consistently allocate individual services, offering them to authorised network users. The availability of such a framework would allow network administrators and user communities to define complex services by from already existing services. Some initiatives are already developing along this line, while others are being developed. See “Composable services” on page 22.
- **Metadata service enhancements**: Two approaches are to be used for eduGAIN metadata structure:
  - For each participating federation, one or several Federation Peering Points (FPP) are allowed to upload metadata to the MDS, according to the federation's and eduGAIN policies. According to the eduGAIN trust structure, each one of these FPPs has a valid eduGAIN certificate issued by a CA accredited to the eduGAIN Truststore. Each `EntityDescriptor` managed by an FPP is signed with the FPP’s eduGAIN private key. When a metadata element is read, its integrity can be established by verifying the digital signature of the elements. This is currently implemented in eduGAIN 1.0. See “Basic Metadata Signature (eduGAIN 1.0)” on page 25.
  - Independent metadata revocation: This requires the infrastructure being able to individually invalidate a certain signature without requesting the incumbent FPP to contact the MDS or killing its own trust by revoking its signing key. See “Independently Revocable Metadata” on page 26.

- **LoA**: Further discussions are needed to create guidelines concerning calculations of, and requirements for, specific LoAs, but the infrastructure support should be as generic and flexible as possible to support varied types of LoA. See “Conclusion” on page35.
- **Single Log-Out**: Work has been done to handle SLO issues to make it simpler for users and SPs. One of the innovations in the area is the AJAX+iFrame SLO implemented by SimpleSAMLphp. See “Solving Single Log-Out issues” on page 37.

Continuing analysis of these advances will take place during the GÉANT3 project. This analysis will also look into non-technical implications (such as policies, operational questions, and so on) in order to ensure that any implementation would not only be technically feasible, but also good operational practice.

6 References

6XACML1.0	Anderson, A., et al.: EXtensible Access Control Markup Language (XACML) V 1.0, OASIS Standard (February 2003)
AuthnContext	Kemp, J., Cantor, S., Mishra, P., Philpott, R., Maller, E.: Authentication Context for the OASIS Security Assertion Markup Language (SAML) v2.0, OASIS Standard (March 2005)
DAMe	DAMe Project web site, http://dame.inf.um.es
DJ5.2.2.,2	López, D.R., et al.: Deliverable DJ5.2.2,2: GÉANT2 Authorisation and Authentication Infrastructure (AAI) Architecture - second edition, GN2 JRA5. GÉANT2 (April 2007)
eduPerson2008	Internet2, **"**eduPerson Object Class Specification (200806)", June 2008, http://www.nmi-edit.org/eduPerson/internet2-mace-dir-eduperson-200806.html
IIAAF	InCommon Identity Assurance Assessment Framework (April 2008)
IIAP	InCommon Identity Assurance ProfilesBronze and Silver, Version 1.0 (April 2008)
LoAs	Zhang, N.: E-Infrastructure Security: An Investigation of Authentication Levels of Assurance (LoAs), Open Grid Forum (2007)
NIST	Polk, W.T., Burr, W.E., Dodson, D.F.: Electronic Authentication Guideline. Recommendations of the National Institute of Standards and Technology (April 2006)
RFC2759	G. Zorn, "Microsoft PPP CHAP Extensions, Version 2", January 2000, http://tools.ietf.org/html/rfc2759
RFC3748	H. Levkowetz (ed.), B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson: "Extensible Authentication Protocol (EAP)", June 2004, http://tools.ietf.org/html/rfc3748
RFC4282	B. Aboba, M. Beadles, J. Arkko, P. Eronen: "The Network Access Identifier",December 2005, http://tools.ietf.org/html/rfc4282
RFC4372	Adrangi, A. Lior, J. Korhonen, J. Loughney: "Chargeable User Identity", January 2006, http://tools.ietf.org/html/rfc4372
RFC5280	D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk: "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", May 2008, http://tools.ietf.org/html/rfc5280
SAML2.0	Cantor, S., Kemp, J., Philpott, R., Eve, M.: Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) v2.0, OASIS Standard (March 2005)
SAML2MetadataAttr	SAML V2.0 Metadata Extension for Entity Attributes, November 2008, http://wiki.oasis-open.org/security/SAML2MetadataAttr
saml-core-2.0-os	Assertions and Protocols for the OASIS Security Assertion(SAML) V2.0, March 2005, http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf
saml-metadata-2.0-os	Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0, March 2005, http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf

UK e-Envoy Office of the e-Envoy, UK online. e-Government Strategy Framework Policy and Guidelines. Version 2.0 (September 2002)

ukfed-tech-spec Federation Technical Specifications, Version 1.1, June 2007, http://www.ukfederation.org.uk/library/uploads/Documents/federation-technical-specifications.pdf

US E-Authentication Bolten, J.B.: E-Authentication Guidance for Federal Agencies (December 2003)

7 Acronyms

A	DNS address record (IPv4)
AAAA	DNS address record (IPv6)
AA	Authentication and Authorisation
AAA	Authentication, Authorisation and Accounting
AAI	Authentication and Authorisation Infrastructure
ASCII	American Standard Code for Information Interchange
CUI	Chargeable User Identity (defined in [RFC4372])
DS	Discovery Service
EAPoL	EAP over LAN
EAPoW	EAP over Wireless LAN
FLRS	Federation Level RADIUS Server
FPP	Federation Peering Points
IETF	Internet Engineering Task Force
IdP	Identity Provider
Internet2	U.S. advanced networking consortium led by the research and education community
NAI	Network Access Identifier
NAS	Network Access Server
OASIS	Open standards body for many standards like SAML
SAML	Security Assertion Markup Language
Shibboleth	The Shibboleth System is a standards based (one of them SAML), open source software package for web single sign-on across or within organizational boundaries
SP	Service Provider
TLRS	Top Level RADIUS Server
URI	Uniform Resource Identifier
UTF-8	8-bit UCS/Unicode Transformation Format
WAYF	Where Are You From (now named Discovery Service)