09.02.09

Deliverable DS5.3.1: Report on Introduction of Monitoring System and Diagnostics Tools

Deliverable DS5.3.1

Contractual Date:	30/04/08
Actual Date:	09/02/09
Contract Number:	511082
Instrument type:	Integrated Infrastructure Initiative (I3)
Activity:	SA5
Work Item:	WI3
Nature of Deliverable:	R
Dissemination Level	PU
Lead Partner	CARNet
Document Code	GN2-09-008v2

Authors: Miroslav Milinovic, Dubravko Penezić (CARNet/Srce), Ian Thomson (DANTE) and SA5 group

Abstract

This Deliverable reports on the introduction of the initial monitoring system and diagnostics tools for the eduroam infrastructure.

Table of Contents

Table of Figures

0 Executive Summary

This deliverable describes the implementation and basic functionality of the eduroam monitoring service.

The eduroam monitoring service's main task is to assess the working status of the eduroam service. It does this by:

- Testing the functionality of the eduroam infrastructure, using methods that mimic the user experience as closely as possible.
- Storing test results in the eduroam database.
- Providing information about the status of the eduroam infrastructure via the eduroam monitoring website (which incorporates a specialised mapping tool, see http://monitor.eduroam.org) and dedicated mailing lists.

The monitoring service has proved to be successful, with two monitoring scenarios being in-production. A testing on demand monitoring scenario is planned to be released as an eduGAIN protected resource to a limited user population (namely institution-level and federation-level personnel).

Planned future developments include implementing more precise use-case monitoring, and providing more sophisticated diagnostic tools for analysing the monitoring data.

1 Introduction

The European eduroam service includes a technology infrastructure and supporting elements (see GN2-07-327v2:, DS5.1.1 “eduroam Service Definition and Implementation Plan”)

The eduroam technology infrastructure relies on a distributed set of AAA servers. The current configuration uses RADIUS as the AAA protocol, and is implemented as a hierarchy of RADIUS servers. The integration of the RadSec protocol into the eduroam infrastructure has been successfully tested and further deployment plans are discussed (see GN2-08-143, DS5.4.1 “Report on RadSec Integration”).

The eduroam supporting elements include:

- Monitoring and diagnostics facilities (monitoring service).
- eduroam Web site.
- eduroam database.
- Trouble ticketing system (TTS).
- Mailing lists.

The basic purpose of the eduroam monitoring is to test the functionality of the eduroam infrastructure. This document describes the architecture of the eduroam monitoring service and provides in-depth information about the implementation status. Plans for future development are also described.

2 Architecture of the eduroam Monitoring Service

The eduroam monitoring service fulfils the following tasks:

- Test functionality of the eduroam infrastructure.
- Store test results.
- Provide information about the status of the eduroam infrastructure.

The monitoring service is designed to test the functionality of the eduroam service in a manner as close to the user experience as possible. Therefore, both the status of the RADIUS servers and the status of the eduroam infrastructure are tested.

However, the eduroam monitoring service is not designed to test each and every element of the infrastructure. It monitors only the confederation infrastructure (i.e. European top level RADIUS servers (ETLRS) and Federation level RADIUS servers (FLRS)). It therefore relies on the National Roaming Operator's (NRO's) monitoring service in cases where a user reports a problem that is not related to the functionality of the confederation infrastructure.

In order to fulfil the tasks listed above, the eduroam monitoring service consists of three main components:

- Monitoring system.
- Monitoring database.
- Monitoring web site (http://monitor.eduroam.org).

2.1 Monitoring System

The Monitoring system consists of a monitoring probe (i.e. monitoring client) and related programs that enable probe management. The probe issues RADIUS requests, collects responses and saves them in the monitoring database. Note that the goal of the tests is to verify the functionality of the tested server or part of the infrastructure. Therefore it is not enough to know that a tested server is accessible. It should also reply appropriately in order to prove its functionality.

The basic monitoring process is shown in Figure 2.1.

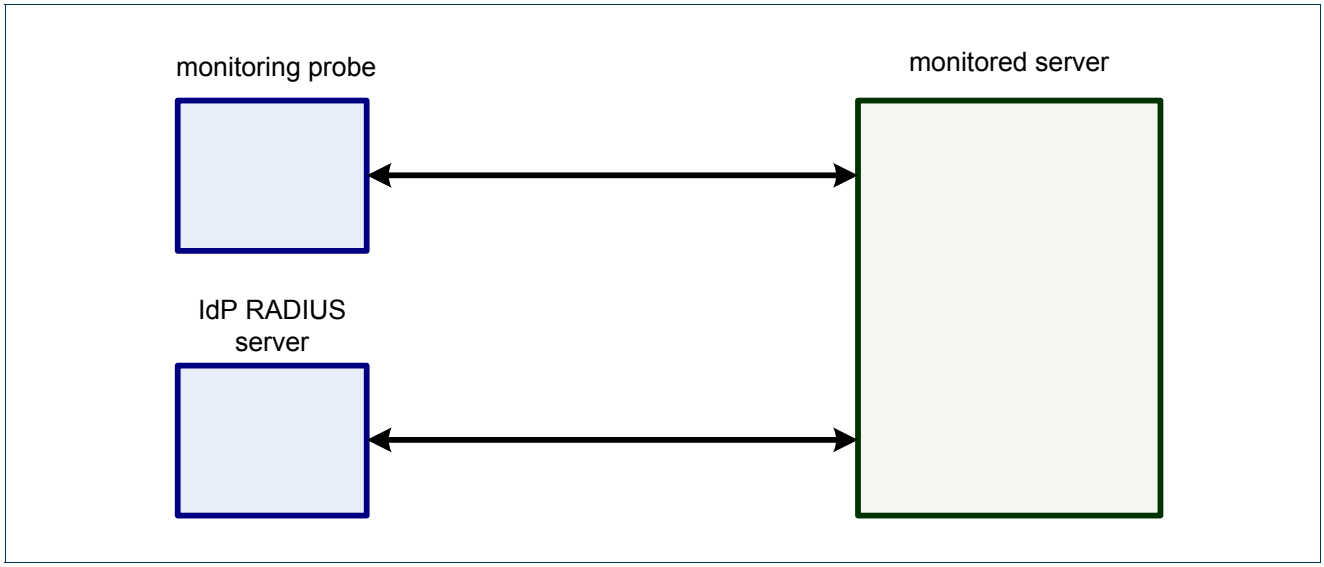

Figure 2.1: Basic eduroam monitoring process

In Figure 2.1:

- The monitoring probe is a RADIUS client capable of sending various types of RADIUS request (typically EAP/TTLS).
- The monitored server is a RADIUS server whose function is being tested. Typically this is a FLRS.
- The IdP RADIUS server is a server that issues the response, thus acting as a loop-back server. Its function is to close the (EAP) tunnel and send back standard responses to the monitoring probe. This function typically can be implemented inside the monitored server.

A special realm (*eduroam.<tld>* , e.g. eduroam.hr) was chosen for monitoring purposes. Therefore each member NRO must set-up all of its FLRS to handle requests for the respective testing realm appropriately. If the realm *eduroam.<tld>* is already used internally by the NRO, another testing realm must be defined.

An example of a FLRS configuration is given in Appendix A. It is important to ensure proper response for the monitoring requests coming either directly form the monitoring probe or from both ETLRS. For further information on FLRS setup, see GN2-07-200v5 (DJ5.1.5,2): “Inter-NREN Roaming Infrastructure and Service Support Cookbook - Second Edition”.

To perform a test, a server monitoring probe issues two test requests:

- Accept test.
- Reject test.

The tested server is marked as working properly if it replies with proper responses to both the Accept and Reject requests.

In both steps:

- The probe creates RADIUS attributes specifically for monitoring purpose.
- The probe creates a specific RADIUS request based on the selected authentication type (typically EAP/TTLS):

```
<engine.item.setup> Request decode:
NAS-IP-Address = 161.53.2.204
NAS-Port = 8484
Calling-Station-Id = "eduroamMON"
Called-Station-Id = "eduroamSCH"
NAS-Identifier = "SA-EAP-TTLS"
Connect-Info = "eduroam-monitoring"
User-Name = test@eduroam.<tld>
EAP-Message = <hidden>
Message-Authenticator = "1CC35899AFA184662A666A5CF86C5184"
```

- The probe sends the RADIUS request and starts measuring the response time.
- The monitored server handles the request and sends back the response.
- The probe evaluates the received response and updates the database. If the probe receives the wrong type of response (e.g. Reject instead of Accept), or a response with incorrect attributes, the response is marked as wrong,
- If the response message did not reach the probe in the expected time (i.e. request is time-outed), the probe resends the request (note that the number of retries is a configurable parameter).

Examples of response messages are:

- Accept:

```
MS-MPPE-Recv-Key =
D021BAE0D5F93B26DFB9E3DEB589B977F141172C3B483B17082CB8D920BDD973
MS-MPPE-Send-Key =
9DA77DB584ED36BD69B5773328821C0016B39A92743BB444D39CCA2CFE5DED7C
Message-Authenticator = "00000000000000000000000000000000"
EAP-Message = <hidden>
```

- Reject:

```
Message-Authenticator = "00000000000000000000000000000000"
EAP-Message = <hidden>
```

2.1.1 Security concerns

In order to strengthen security and avoid the abuse of the monitoring mechanism by the third parties, the following attributes with specific monitoring values are used in creating the RADIUS requests:

```
NAS-IP-Address = 161.53.2.204 (this is IP address of the probe)
NAS-Port = 8484
Calling-Station-Id = (probe specific data)
Called-Station-Id = (probe specific data)
NAS-Identifier = SA-EAP-TTLS (depends on the authentication type used)
Connect-Info = eduroam-monitoring
```

Furthermore, the testing user's password is known only by the monitoring probe. This password can be re-generated for each test.

2.2 Monitoring Database

The monitoring database is a specific part of the eduroam database (see GN2-08-051v2: DJ5.1.6 "Evaluation of New Roaming Technologies and Possible Integration into AAI" for more information on the eduroam database). It is used to store the data acquired by the monitoring system, thus providing vital information about the status of the eduroam infrastructure at any given time.

Figure 2.2 shows the data model of the monitoring database and its relation with the rest of the eduroam database. Detailed information on the monitoring database design is given in Appendix B.

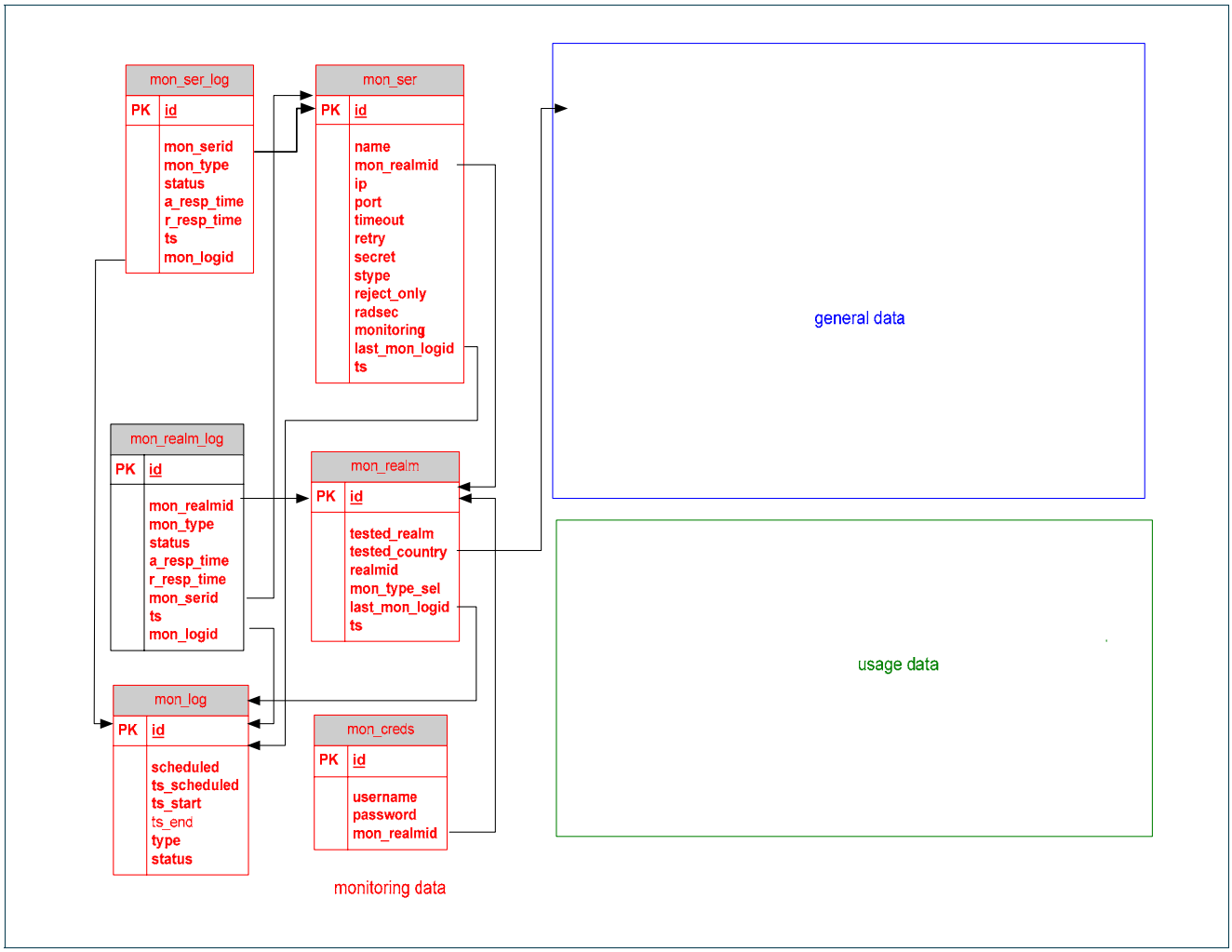

Figure 2.2: eduroam database

2.3 Monitoring Web Site

The monitoring web site is used to display the information stored in the database. The web site provides all user groups (see DS5.1.1.) with easy access to information on both the current status of the infrastructure ("eduroam weather map") and daily statistics on its "health".

The public part of the monitoring web site is available at http://monitor.eduroam.org. The internal web pages, which should only be used by restricted users (e.g. SA5 group members), are protected as an eduGAIN resource.

See section 3 "Implementation status" for details of the implementation of the monitoring web site.

2.4 Monitoring Work Flow

Three different monitoring scenarios are envisaged, resulting in three different monitoring workflows:

- Server monitoring.
- Infrastructure monitoring.
- Testing on demand.

2.4.1 Server monitoring

Server monitoring is the simplest scenario, where each and every ETLRS and FLRS is tested by the probe. The workflow is shown in Figure 2.3:

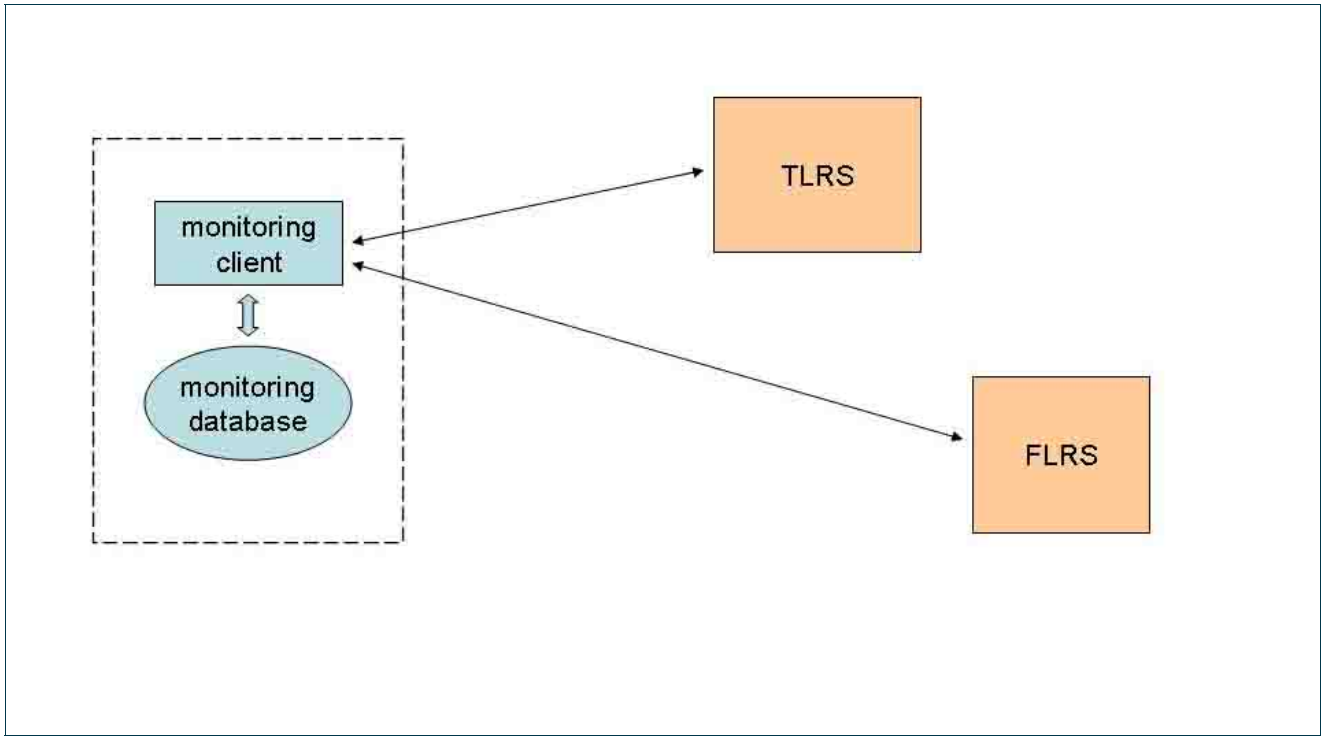

Figure 2.3: Server monitoring

2.4.2 Infrastructure monitoring

Infrastructure (realm) monitoring is slightly more complicated than server monitoring as it tests the functionality of the respective realms (not each individual FLRS). The probe issues a request for a realm A (e.g. *.hr*) and

sends it via each of the ETLRS servers. Thus we test the infrastructure path between federation A and European TLRSs. The workflow is shown in Figure 2.4.

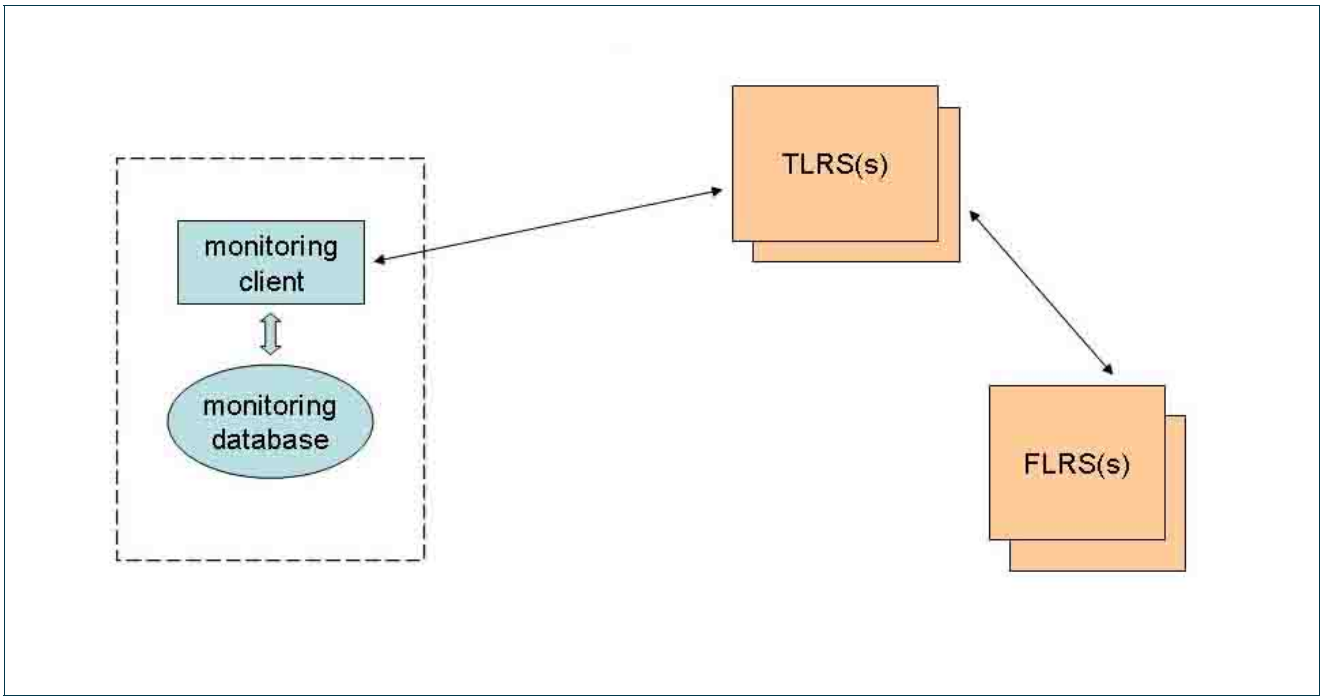

Figure 2.4: Infrastructure monitoring

2.4.3 Testing on demand

With the testing on demand scenario, the test is designed to mimic the user experience as much as possible. The probe issues a request for a realm A (e.g. *.hr*) and sends it to the FLRS(s) of realm B (e.g. *.de*). Thus we test the infrastructure path between federation A and federation B. The workflow is shown in Figure 2.5.

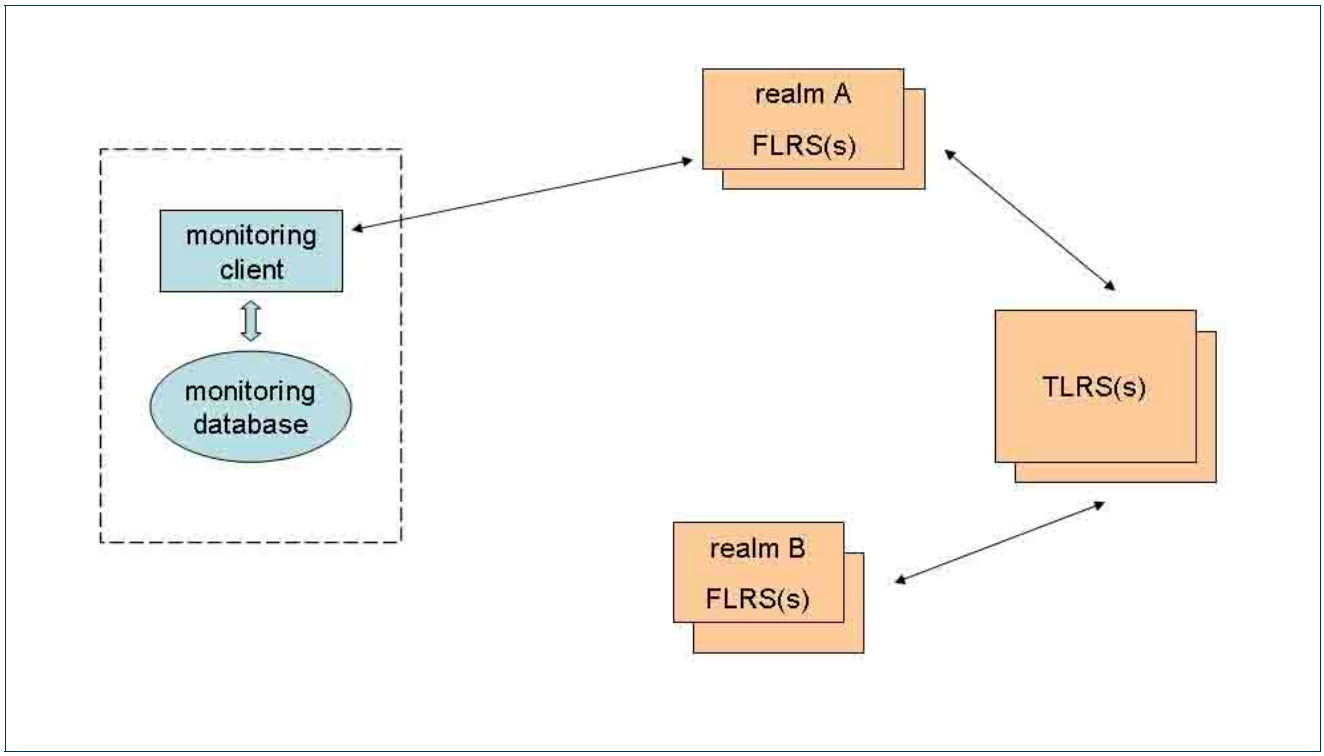

Figure 2.5: Testing on demand

3 Implementation status

The work on implementation of the monitoring system stared at the beginning of the SA5 (fourth quarter of 2007). The first prototype was available in January 2008. Work on the first version of the service was finished in May 2008. The monitoring service was implemented in August 2008 and has been running in full production since then. It is maintained by the SA5 Operations Team (OT) participants from Srce.

A dedicated host (hostname: monitor.eduroam.org, IP: 161.53.2.204) has been installed and put into production. A complete monitoring service (probe, database and web site) has been installed on that machine and has been running smoothly since the start.

Currently, two monitoring scenarios (server monitoring and realm monitoring) are run in-production, while testing on demand is planned to be released as an eduGAIN protected resource to a limited user population (namely institution-level and federation-level personnel).

In order to actively provide information about possible malfunction of the infrastructure or monitored servers monitoring service is capable of sending e-mail messages with error reports to the corresponding e-mail addresses registered by the NROs in the eduroam database.

By the end of 2008 29 out of 34 member NROs are actively monitored (see Appendix C for full list). The process of adding the remaining five realms to the system is ongoing, and requires action (i.e. configuration changes) by the respective NROs. Currently the monitoring system tests two ETLRS and 54 FLRS servers.

The monitoring service is IPv6 enabled. RadSec monitoring capability has been tested and will be available in full production by the end of February 2009.

As well as regular, automatic tests (performed once per hour) there is also the ability to issue an on-demand, immediate test.

As explained in section 2.1 “Monitoring System”, the monitoring probe retries a test request if it has been timed-out. Currently the number of retries is set to 1 and time-out is set to 5 seconds.

With infrastructure (realm) monitoring, probe sends the request via both ETLRS servers.

Acquired data is stored in the database (see section 2.2 “Monitoring Database“).

Currently five different statuses of a server or a realm are recognised and represented on the web site:

- **OK**: All of the requests sent by the probe resulted with proper responses.
- **Possible Problems**: At least one of the requests sent by the probe resulted with proper response, while others might have resulted in errors (wrong response or no response at all).
- **Wrong Response**: All of the requests sent by the probe resulted in the wrong response, but the probe registered at least one response that was not timed-out.
- **No Response**: All of the requests sent by the probe resulted in no response (i.e. timed-out)
- **Not Monitored**: The server/realm is not monitored.

A special map tool has been developed in order to present the current status of the eduroam infrastructure on the monitoring web site. The following pictures show the server status and realm status at a given time (see http://monitor.eduroam.org):

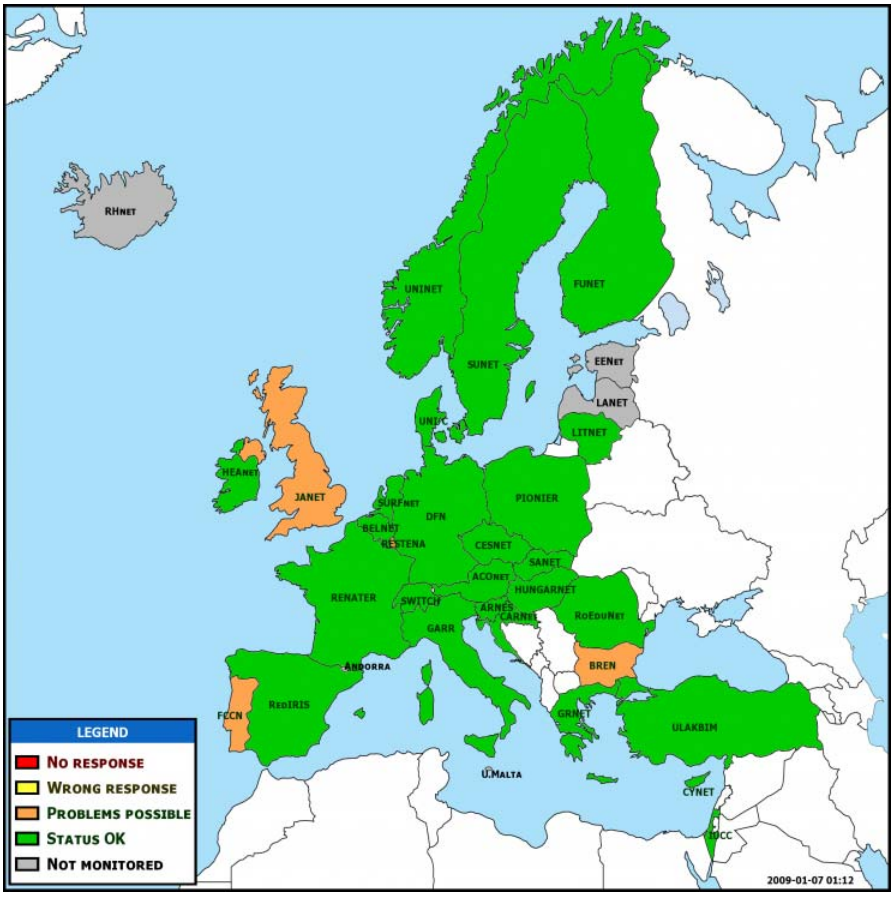

Figure 3.1: Server status

If a NRO has more then one FLRS, the status shown on the map is determined as follows:

- **OK**: All of the servers are marked as OK.
- **Possible Problems**: At least one of the servers is marked as OK, while others might have been marked in errors (wrong response or no response at all).
- **Wrong Response**: All of the servers are marked as wrong response, but at least one server was not marked as timed-out.
- **No Response**: All of the servers are marked as no response (i.e. timed-out)
- **Not Monitored**: Servers are not monitored.

More detailed information on ETLRS and FLRS status, including the monitoring history, is available at http://monitor.eduroam.org/eduroam/mon_server.php.

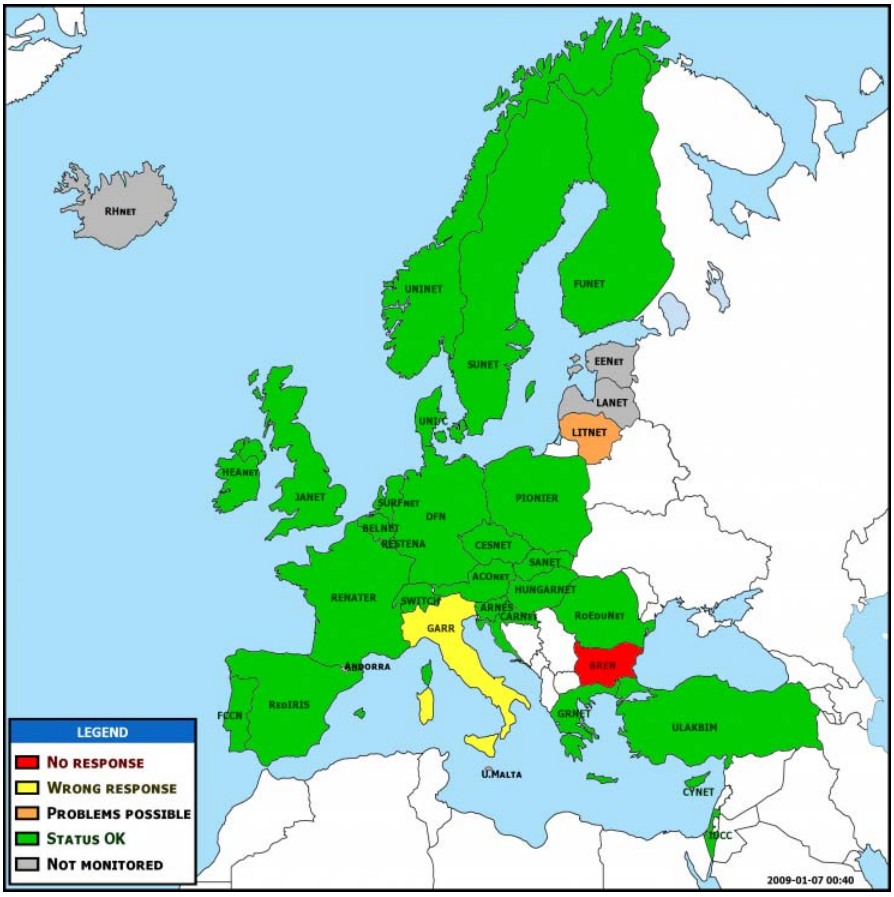

Figure 3.2: Realm status

The realm status shown in the display is determined based on the result of the infrastructure monitoring (see 2.4.2.). The probe sends the standard pair of requests via each of the ETLRS servers. The realm status is presented as one of:

- **OK**: All of the requests sent by the probe resulted in proper responses.
- **Possible Problems**: At least one of the requests sent by the probe resulted in a proper response, while others might have resulted in errors (wrong response or no response at all).
- **Wrong Response**: All of the requests sent by the probe resulted in the wrong response, but the probe registered at least one response that was not timed-out.
- **No Response**: All of the requests sent by the probe resulted in no response (i.e. timed-out)
- **Not Monitored**: The realm is not monitored.

More detailed information on realm status including the monitoring history is available at http://monitor.eduroam.org/eduroam/mon_realm.php

Finally, an easy to understand, end user oriented map is also generated (see Figure 3.3.).

Figure 3.3: eduroam infrastructure status

The status shown in the above picture is determined by the realm status monitoring and is presented as one of:

- **Service available**: The realm is marked as OK or with possible problems.
- **Service not available**: The realm is marked as wrong response or no response at all
- **Not Monitored**: The realm is not monitored.

4 Conclusions

eduroam members have been using the current monitoring service implementation with success. Feedback (particularly on the mapping functionality) has been favourable.

From this experience, it is believed that the current implementation will serve as an excellent basis for developing the service still further. Analysing the use of and feedback about the service, proposed future developments are to include:

- Implementing functionality to enable testing the user experience of eduroam in real use-case scenarios, which includes interaction with individual NRO's monitoring system.
- Providing more sophisticated diagnostic tools based on the data acquired with the current monitoring service.
- Improving the presentation of data acquired from the service, thereby making it easier for users to interpret and analyse the reports.
- Improving the usability of the public and internal web interface by analysing, developing and implementing user requirements.
- Improving real-time information and operational reliability of the monitoring service.
- Utilise eduroam TTS for reporting and resolution of problem by automatic ticket opening based on the monitoring status.
- Improving the coverage of the service so that it monitors all FLRSs and thus all federations (i.e. realms).

5 References

[DJ5.1.5,2]	GN2-07-200v5 DJ5.1.5,2 “Inter-NREN Roaming Infrastructure and Service Support Cookbook - Second Edition”
[DJ5.1.6]	GN2-08-051v2: DJ5.1.6 “Evaluation of New Roaming Technologies and Possible Integration into AAI”
[DS5.1.1]	GN2-07-327v2:, DS5.1.1 “eduroam Service Definition and Implementation Plan”
[DS5.4.1]	GN2-08-143, DS5.4.1 “Report on RadSec Integration”

6 Acronyms

AAA	Authentication, Authorisation, and Accounting
EAP	Extensible Authentication Protocol
eduGAIN	GÉANT Authorisation Infrastructure for the research and education community
ETLRS	European Top Level RADIUS Server
FLRS	Federation Level RADIUS Server
IdP	Identity Provider
IPv6	Internet Protocol Version 6
NRO	National Roaming Operator
OT	Operations Team
SA5	Service Activity 5
TLRS	Top Level RADIUS Server
TTLS	Tunnelled Transport Layer Security
TTS	Trouble Ticketing System

Appendix A FLRS configuration example

This section gives an example of a FLRS configuration for eduroam monitoring. It shows a configuration for Radiator RADIUS server and for testing .lu servers.

```
# --- ETLR monitoring ---
<Client 161.53.2.204>
        Secret blablabla
        Identifier Monitoring-ETLR-1
</Client>
<Handler User-Name=/^test\@eduroam\.lu$/>
        <AuthBy RADIUS>
                Host 161.53.2.204
                AuthPort 1812
                AcctPort 1813
                Secret blablabla
        </AuthBy>
</Handler>
```

Appendix B Monitoring database structure

B.1 table: mon_realm

Contains information about monitored federations (realms).

field name	field description
id	automatically generated identifier
tested_realm	realm used for testing (usually eduroam)
tested_country	country code used for testing (usually respective realm's country code)
realmid	id of the monitored realm (i.e. federation)
mon_type_sel	coded type of tests to be performed (PAP, EAP-TTLS, ...)
monitoring	1=default, 0 if this realm should not be tested
last_mon_logid	id of the last successful monitoring job for this realm
ts	date: last changed

B.2 table: mon_ser

Contains information about monitored RADIUS servers.

field name	field description
id	automatically generated identifier
name	server's (host) name
mon_realmid	id of respective realm used for testing (mon_realm table)
ip	server's IP address
port	RADIUS server: port number
timeout	RADIUS server: timeout
retry	RADIUS server: number of retries
secret	RADIUS server: secret

stype	coded type of server (TLRS, FLRS)
reject_only	0=default, 1 if only reject logic tests are performed
radsec	0=default, 1 if it is RadSec server
monitoring	1=default, 0 if this server should not be tested
last_mon_logid	id of the last successful monitoring job for this server
ts	date: last changed

B.3 table: mon_ser_log

Contains results of RADIUS server monitoring.

field name	field description
id	automatically generated identifier
mon_serid	id of respective server
mon_type	coded type of performed tests (PAP, EAP-TTLS, ...)
status	server status code (see B.7)
a_resp_time	response time for accept test
r_resp_time	response time for reject test
ts	date: created
mon_logid	id of the respective monitoring job

B.4 table: mon_realm_log

Contains results of infrastructure (realm) monitoring.

field name	field description
id	automatically generated identifier
mon_realmid	id of respective realm (mon_realm table)
mon_type	coded type of performed tests (PAP, EAP-TTLS, …)
status	realm status code (see B.7)
a_resp_time	response time for accept test
r_resp_time	response time for reject test
mon_serid	id of TLRS used for test
ts	date: created
mon_logid	id of the respective monitoring job

B.5 table: mon_log

Contains internal monitoring information (e.g. info on scheduled tasks)

field name	field description
id	automatically generated identifier
scheduled	coded value (AUTOMATIC, MANUAL)
ts_scheduled	scheduled time
ts_start	start time
ts_end	stop time
type	job type (10=all servers; 11=single server; 20=all realms; 21=single realm)
status	job status (END, RUNING, START, ERROR)
ts	date: last update

B.6 table: mon_creds

Contains credentials used for monitoring

field name	field description
id	automatically generated identifier
username	test username
password	test password / automatically generated
mon_realmid	id of respective realm used for testing (mon_realm table)

B.7 server/realm status codes

code	description
0	Both Accept and Reject response are correct.
-1	Reject response was wrong (type). Accept response was correct
-10	Reject request timed-out (no response). Accept response was correct
-11	Reject response was wrong (syntax). Accept response was correct
-2	Accept response was wrong (type). Reject response was correct
-20	Accept request timed-out (no response). Reject response was correct
-21	Accept response was wrong (syntax). Reject response was correct
-3	Both Accept and Reject responses were wrong (type).
-31	Reject request timed-out (no response). Accept response was wrong (type)

-32	Accept request timed-out (no response). Reject response was wrong (type)
-33	Both Accept and Reject request are timed-out (no response)
-35	Reject response was wrong (syntax) and Accept response was wrong (type)
-36	Accept response was wrong (syntax) and Reject response was wrong (type)
-37	Both Accept and Reject responses were wrong (syntax)
-9	Wrong response caused by protocol/unexpected error

Appendix C Monitoring status per NREN/NRO

Domain	NREN/NRO	Country	Status
ad	U.Andorra	Andorra	Not monitored
at	ACOnet	Austria	Monitored
be	BELNET	Belgium	Monitored
bg	BREN	Bulgaria	Monitored
ch	SWITCH	Switzerland	Monitored
cy	CYNET	Cyprus	Monitored
cz	CESNET	Czech Republic	Monitored
de	DFN	Germany	Monitored
dk	Nordunet/UNI-C	Denmark	Monitored
ee	EENet	Estonia	Not monitored
es	RedIRIS	Spain	Monitored
fi	Nordunet/FUNET	Finland	Monitored
fr	RENATER/CRU	France	Monitored
gr	GRNET	Greece	Monitored
hr	CARNet/Srce	Croatia	Monitored
hu	HUNGARNET	Hungary	Monitored
ie	HEAnet	Ireland	Monitored
il	IUCC	Israel	Monitored
is	Nordunet/RHnet	Iceland	Not monitored
it	GARR	Italy	Monitored
lt	LITNET/KTU	Lithuania	Monitored
lu	RESTENA	Luxembourg	Monitored
lv	Sigmanet/LANET	Latvia	Not monitored
mt	U.Malta	Malta	Not monitored
nl	SURFnet	Netherlands	Monitored
no	Nordunet/UNINET	Norway	Monitored
pl	PIONIER/U.Torun	Poland	Monitored
pt	FCCN	Portugal	Monitored
ro	RoEduNet	Romania	Monitored
se	Nordunet/SUNET	Sweden	Monitored
si	ARNES	Slovenia	Monitored
sk	SANET	Slovakia	Monitored
tr	ULAKBIM	Turkey	Monitored
uk	JANET	United Kingdom	Monitored