DjNRO: Django-based application for National Roaming Operators, or how to manage your eduroam database (and more)

Leonidas Poulopoulos
(leopoul@noc.grnet.gr)
Network Applications Developer

Zenon Mousmoulas
(zmousm@noc.grnet.gr)
Network Applications Developer

Greek Research & Technology Network (GRNET) NOC
56, Mesogeion Ave.
11527 Athens, Greece

Paper type: Technical paper

Keywords: eduroam, NRO, database, distributed, management, extensions

Abstract: Eduroam database management is an important responsibility for eduroam National Roaming Operators (NROs). In this paper we present a web-based application we developed, which tackles the management of the eduroam database, among other things. The application has been deployed in GRNET since 2012 and is vastly simplifying and improving operations for the NRO as well as drawing attention from end users and administrators of Greek eduroam federation member institutions. The project is released as open-source and has already seen interest from several parties for deploying it.

1. Introduction
Eduroam stands for EDUcation ROAMing. It offers users from participating academic institutions secure Internet access at any other eduroam-enabled institution. The eduroam architecture that makes this possible is based on a number of technologies and agreements, which together provide the eduroam user experience: "open your laptop and be online". The crucial agreement underpinning the foundation of eduroam involves the mechanism by which authentication and authorization works[1]. An eduroam federation comes with administrative as well as technical requirements. A National Roaming Operator (NRO) is the administrative entity responsible for running eduroam at a country level. The NRO must maintain a comprehensive overview of eduroam within its' service area and must regularly report about the state of the national eduroam federation. In the European region the vehicle for such reporting is provided through the eduroam database, where information about the NRO, the service providers (SPs) and identity providers (IdPs) is stored. In this paper we present a management application for NROs, which initially has a twofold purpose: expose information in the structured format harvested by eduroam.org and serve as a national eduroam web frontend. DjNRO was designed and developed by members of GRNET NOC using the Python Django framework.

GRNET (Greek Research and Technology Network) provides Internet connectivity and services to the Greek Universities and academic and research institutes. GRNET maintains points-of-presence in all major Greek cities (approximately 40) and leases dark fiber across the country for its backbone and access network.

2. Reasons to develop DjNRO
Maintaining and managing a local eduroam database is an important responsibility for an NRO. Eduroam.org periodically polls the web sites of the members of the European eduroam confederation for information about participating IdPs, SPs and service locations, as well as basic accounting data [2]. The source of information should be the NRO database, so that any changes there are reflected in the XML files harvested by eduroam.org. Therefore service locations, contacts etc. should be kept up-to-date in order to provide a consistent view of eduroam service availability and make it easier for eduroam users to seek support when necessary. This was the original design goal for DjNRO.

2.1 Before DjNRO
Before moving forward, it is worth mentioning how GRNET previously maintained the NRO database and exposed information to eduroam.org. A minimal schema was used for the database and records were maintained by federation administrators, collecting information from Greek IdPs and SPs through e-mail and tracking changes through the general purpose trouble ticket system supporting operations in GRNET. In practice,
However, such procedures would often be neglected, thus database records would become stale. On the other hand, the eduroam web site had been developed quite some time ago and employed technologies that had become obsolete, making it hard both to use and (mainly) to maintain. The web site (Figure 1) contained a static webpage that the NRO administrators would update manually, which was completely detached from the data provided to eduroam.org.

Figure 1. Old eduroam.gr website

Considering the above, it was clear that we needed to make it easier, faster and more attractive for institution administrators to manage their own data as well as to expose this information to end users on the eduroam web site, through a clean and functional interface. One of the major goals we aimed at was offloading the NRO administrators from having to maintain all records for all institutions in Greece. Where possible, we also opted for a domain-agnostic and multi-lingual design, anticipating that the code would eventually be released as open-source software and hoping that it may be used by other NROs. This is how DjNRO was born.

2.2 DjNRO database

Since DjNRO is meant to be an eduroam database management platform, it inherits the eduroam database schema with some modifications and a number of extensions, where necessary. The original eduroam database schema is depicted in Figure 2.
DjNRO core database schema is shown in Figure 3.

Figure 3. DjNRO database schema

More details for the core database schema along with sample data taken from GRNET’s deployment are provided in Appendix I.
3. Architecture

The “Dj” part in DjNRO’s name comes from the web framework that was used to develop it: Django. Django is a high-level Python Web framework (MVC) that encourages rapid development and clean design. Django models the DjNRO database into Python objects through its’ ORM and transparently manages an application’s interaction with the database, which boosts development productivity. DjNRO’s presentation is carried out through the template engine that is built-in to Django. Since web development steers towards mobility and responsiveness, we designed the frontend of DjNRO using the Twitter Bootstrap web-front framework, also integrating the Google Maps API front-end library. DjNRO runs as a WSGI application that is typically served by the Apache HTTP server. Apart from the core Google Maps library, some additional server-side libraries have been integrated, to serve the needs of geo-location, marker grouping, (front/reverse) geocoding and path finding. The application supports multiple languages through translation tags, which cover almost every text occurrence. The Greek eduroam web site [3] is powered by DjNRO (both the front-end and the management application at the back-end) and was rolled out in November 2012. DjNRO architecture, services and APIs are depicted in Figure 4.

Figure 4. DjNRO architecture and services

Migration from the previous status quo to the DjNRO database was handled by a bootstrapping XML script that transformed the XML files exported to the eduroam database into DjNRO objects. One of the major goals of development was compatibility with eduroam database and the proper generation of the XML files eduroam.org harvests. Having data in place the frontend was setup to depict eduroam service locations. Along came the development of the management API to ease the administration on behalf of our institutions administrators. Domain data, the CAT provisioning API and the development of external applications came later on when we had sufficient real data to work with. All the above are described in detail in the following paragraphs.

4. DjNRO services and APIs

4.1 Management interface
In essence, DjNRO offers 2 major web interaction modes: the management interface and the end-user interface. As one could easily guess, changes via the management interface are reflected on the users’ interface. Beyond keeping eduroam.org up-to-date, DjNRO is also a distributed management application, in the sense that information about institutions, locations and services is maintained by the respective eduroam administrators. In proportion to the federated nature of eduroam, our deployment uses SAML-based federated authentication and authorization through GRNET’s AAI federation. However authentication is also possible through social networks, namely Google and Twitter (practically any OAuth capable provider can be supported), as shown in Figure 5.

![Figure 5. Authentication options](image)

Institution administrators provision their own accounts in the management application and then NRO administrators simply verify their association with a particular institution. After that, they manage eduroam service locations, contact points and institution information, taking responsibility for their own data. The management interface is shown in Figure 6.

![Figure 6. Management interface](image)

Extensive geo-location and mapping functionality is included in the management application, using Google Maps, which makes it easier and faster for eduroam administrators to accurately provide coordinates of service locations. For example, to add a service location the institution administrator can drag and drop the eduroam location pin, or type the address in a corresponding field or even fill in the location coordinates.

### 4.2 Infrastructure monitoring extensions

Institution administrators also provide data for the RADIUS infrastructure and contacts per service location. The institution data, which is relevant for the federation RADIUS and monitoring infrastructure, is aggregated and serialized as YAML or JSON documents, which are exposed through protected HTTP resources. This data can then be consumed by a script that generates RADIUS server configuration (for freeradius and radsecproxy) as well as service monitoring configuration (for nagios/icinga). The data can be updated at regular intervals or semi-automatically and the whole process can be managed by configuration and automation software, such as Puppet and Ansible. The configuration generator is implemented as a Python script that translates data to configuration for the previously mentioned software by using purpose-built Mako templates, which are bundled as examples with DjNRO.
Figure 7. Infrastructure monitoring extensions

The YAML view (Table 1), which is used by default in the configuration generator, also provides a human readable synopsis of the infrastructure that comprises the federation; it is also natively supported by Puppet and Ansible as a data format, so there can also be other uses for (subsets of) this data.

Table 1. YAML export sample

```yaml
---
clients:
  client_6_28cbd99cc8c905492133a46723049ee7:
    host: <censored>
    label: <censored>
    secret: <censored>
  client_8_9eb3058409431a079f523ea64bd5ac6:
    host: <censored>
    label: <censored>
    secret: <censored>
institutions:
  - clients:
        - client_8_9eb3058409431a079f523ea64bd5ac6
        - client_10_dd6a92ed97d6697737d3dde8f39a3cc5
  id: auth.gr
realms:
  "*.auth.gr":
    proxy_to:
      - server_8_9eb3058409431a079f523ea64bd5ac6
      - server_10_dd6a92ed97d6697737d3dde8f39a3cc5
      auth.gr:
        proxy_to:
          - server_8_9eb3058409431a079f523ea64bd5ac6
          - server_10_dd6a92ed97d6697737d3dde8f39a3cc5
        type: 3
servers:
  server_10_dd6a92ed97d6697737d3dde8f39a3cc5:
    acct_port: 1813
    auth_port: 1812
    host: <censored>
    label: <censored>
    rad_pkt_type: auth+acct
    secret: <censored>
    status_server: true
  server_8_9eb3058409431a079f523ea64bd5ac6:
    acct_port: 1813
    auth_port: 1812
    host: <censored>
    label: <censored>
    rad_pkt_type: auth+acct
    secret: <censored>
    status_server: true
...
### Table 2. Generated freeradius configuration sample

```plaintext
#}}

```{auth.gr}

type = auth+acct
ipaddr = <censored>
port = 1812
secret = <censored>
response_window = 20
zombie_period = 40
revive_interval = 120
status_check = status-server
check_interval = 30
num_answers_to_alive = 3

```}

```plaintext
home_server server_8_9eb3058409431a079f523ea64bcb5ac6 {

type = auth+acct
ipaddr = <censored>
port = 1812
secret = <censored>
response_window = 20
zombie_period = 40
revive_interval = 120
status_check = status-server
check_interval = 30
num_answers_to_alive = 3

```}

```plaintext
home_server_pool auth.gr {

type = fail-over
home_server = server_8_9eb3058409431a079f523ea64bcb5ac6
home_server = server_10_dd6a92ed97d6697737d3d8e8f39a3cc5

```}

```plaintext
realm auth.gr {

``` pool = auth.gr
nostrip

```}

```plaintext
home_server_pool _wildcard_.auth.gr {

``` type = fail-over
home_server = server_8_9eb3058409431a079f523ea64bcb5ac6
home_server = server_10_dd6a92ed97d6697737d3d8e8f39a3cc5

```}

```plaintext
realm "~.+\auth\\.gr$" {

``` pool = _wildcard_.auth.gr
nostrip

```}

```#}}


**4.3 Mailing lists extension**

Similarly, lists of institution administrators and/or service location contacts are also produced by DjNRO in a simple tabular format that can be used by mailing list software, such as Sympa, to automatically maintain mailing lists in a federation.

**4.4 eduroam.org xml provisioning**

As already noted, one of the basic tasks for DjNRO is to provide federation data in XML format, according to the eduroam database specification. DjNRO thus provides `/general/realm.xml`, `/general/institution.xml` and `/usage/realm_data.xml`.

**4.5 eduroam CAT integration**

Beyond that, DjNRO has been integrated with eduroam CAT (Configuration Assistant Tool), which aims to support eduroam IdP administrators by allowing them to generate eduroam installers for their users. CAT is a both a software toolkit, developed mainly in the context of the GÉANT project, and a service provided by GÉANT [4]. It provides a simple enrollment API that is supported by DjNRO, making it possible for institution administrators to automatically create new institutions and take control by enrolling themselves in CAT (Figure 8). This eliminates the need for manual enrollment to be carried out by federation-level administrators and
allows institution administrators to opt-in to CAT at their own pace, as well as automatically provision to CAT institution data they have already provided in DjNRO. It should be noted that any number of different CAT instances can be supported in a DjNRO instance, by configuring the respective API endpoints and keys.

Figure 8. eduroam CAT enrollment integration

4.6 eduroam.TLD front-end
The end-users interface includes what users expect to see when they visit a national eduroam web site: service locations, contact details per institution, information about eduroam etc. Location information is visualized on the front-end through dynamically generated maps that instantly reflect updates, again using Google Maps. Location data beyond the territory of the NRO are also fetched periodically from eduroam.org in KML format. This enables the provision of maps showing eduroam service locations around the world, rather than having to redirect users to eduroam.org for such functionality.

4.7 Geolocation service and API
Using the available geo data, we developed the closest point service API, which handles locating the closest eduroam service location. The API receives the current location coordinates in a GET request, calculates the closest point through server-side triangulation and returns a JSON reply with its’ coordinates and a detailed description. Based on this API, we developed the closest point web page that allows users to find the closest eduroam location (provided through browser geo-location APIs) or a location they search for; both are shown on the map, along with a walking trail indicating how to reach the automatically selected eduroam location (Figure 9). We believe this tool can be useful to anyone exploring the availability of eduroam around the world, including travelers. It does require the user to go online, unlike offline navigation applications (for example on mobile devices); however it is fast and lightweight, since the user only provides coordinates of his location and computation is done server-side, and can be handy, since no prior installation or preparation is necessary.
Based on the closest point service API, a plethora of external mobile and web applications can be developed. As a proof of concept we developed an application for a wearable device, Pebble Watch. The application, upon loading, gets the current location, sends a request to eduroam.gr API and receives a response with the closest eduroam location coordinates and details. Then, through a request to Google Directions API, a response is received with walking directions to the closest eduroam location. Directions can be scrolled through using Pebble’s buttons, while the user location is frequently updated as the user walks towards the desired location. The application was developed using the Simply.js SDK, using the Cloudpebble web interface [5] and it is shown in Figure 10. The Pebble application can be downloaded via the Pebble app store by searching for “pebduroam”.

![Figure 9. Eduroam geolocation tool](image)

**4.8 Pebble-watch eduroam application**

Based on the closest point service API, a plethora of external mobile and web applications can be developed. As a proof of concept we developed an application for a wearable device, Pebble Watch. The application, upon loading, gets the current location, sends a request to eduroam.gr API and receives a response with the closest eduroam location coordinates and details. Then, through a request to Google Directions API, a response is received with walking directions to the closest eduroam location. Directions can be scrolled through using Pebble’s buttons, while the user location is frequently updated as the user walks towards the desired location. The application was developed using the Simply.js SDK, using the Cloudpebble web interface [5] and it is shown in Figure 10. The Pebble application can be downloaded via the Pebble app store by searching for “pebduroam”.

![Figure 10. Pebble-watch closest eduroam application](image)
5. DjNRO roadmap

Building on a solid foundation as well as a modular and extensible design, there are many interesting features already planned or being considered for extending the functionality of DjNRO. Connection to and/or interaction with general-purpose CRM systems that may be used by the NRO, as well as multi-domain operation support systems (OSS) for eduroam (involved for example in abuse handling or troubleshooting for end users), may make sense, in order to further streamline eduroam operations. In this context we plan to integrate helpdesk support in DjNRO. The development of a REST API is also in the roadmap of DjNRO. Using a REST API, institution administrators should be able to update service location data in DjNRO as well as get the service locations in JSON or XML/KML format, allowing for integration with other services and applications. Some large institutions in the Greek eduroam federation have expressed interest in being able to do bulk service location updates, either by pushing data to eduroam.gr or setting up eduroam.gr to pull such data from their systems. Such a feature is thus also considered for DjNRO; the data exchange could be based on a subset of the eduroam database XML schema. Since configuration for monitoring systems can be automatically generated, monitoring data can also be integrated (e.g. through icinga APIs) in a dashboard that would indicate problems in the eduroam infrastructure. Multi-tenancy support, enabling deployment of DjNRO for multiple NROs through a single instance, is one of the longer-term development goals of DjNRO. In order to be able to support multi-tenancy we are first going to introduce multi-level user roles and resource authorization. Moreover we will introduce an additional layer on top of the existing schema that will allow for administration at the confederation level. Finally we are always open to suggestions, which can be discussed on the DjNRO mailing list [6].

6. Support and adoption

DjNRO is available as open-source software [7] and is supported by GRNET NOC members. A mailing list has been setup for adopter support, questions and technical discussion [6]. DjNRO was presented at the 30th TF-Mobility and Network Middleware Meeting in Vienna [8]. Apart from Greece, DjNRO has been adopted by the NROs in Austria, New Zealand [9] (the latter having also provided reference to DjNRO in a presentation for eduroam [10]) and AARNet, Australia. There has also been interest in DjNRO from the NROs in Finland and South Africa.

7. Appendix I

In this section we provide a sample of data taken from GRNET’s deployment, which also shows the core database schema in more detail. For the sake of readability, some columns have been omitted. It is important to note that schema management and relationships at the database level are managed by the Django framework. Thus additional fields or tables that appear in the database are part of Django’s ORM.

<table>
<thead>
<tr>
<th>Contact</th>
<th>contact_email</th>
<th>id</th>
<th>contact_phone</th>
<th>contact_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>zim....grnet.gr</td>
<td>2</td>
<td>+302....</td>
<td>Zenon Mousmoulas</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Realm Contact</th>
<th>id</th>
<th>realm_id</th>
<th>contact_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Institution</th>
<th>realm_id</th>
<th>id</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Institution Details</th>
<th>address_city</th>
<th>number_user</th>
<th>institution_id</th>
<th>ts</th>
<th>oper_name</th>
<th>number_id</th>
<th>address_street</th>
<th>id</th>
</tr>
</thead>
<tbody>
<tr>
<td>Athens</td>
<td>80</td>
<td>2</td>
<td>2014-05-16 15:15:19</td>
<td>admin.grnet.gr</td>
<td>80</td>
<td>Mesogion 56</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Institution Contact Pool</th>
<th>contact_id</th>
<th>id</th>
<th>institution_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Institution Details Contact</th>
<th>id</th>
<th>institutiondetails_id</th>
<th>contact_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>154</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>
### Inst Realm

<table>
<thead>
<tr>
<th>id</th>
<th>instid_id</th>
<th>realm</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>admin.grnet.gr</td>
</tr>
</tbody>
</table>

### Inst Realm Mon

<table>
<thead>
<tr>
<th>realm_id</th>
<th>id</th>
<th>mon_type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>8</td>
<td>localauthn</td>
</tr>
</tbody>
</table>

### Inst Realm ProxyTo

<table>
<thead>
<tr>
<th>id</th>
<th>instrealm_id</th>
<th>instserver_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>6</td>
</tr>
</tbody>
</table>

### Inst Server

<table>
<thead>
<tr>
<th>typ e</th>
<th>name</th>
<th>prot o</th>
<th>instid _id</th>
<th>secret</th>
<th>i d</th>
<th>host</th>
<th>status_serve r</th>
<th>acct_p  ort</th>
<th>auth_p ort</th>
<th>addr_ty pe</th>
<th>rad_PKT_T ype</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>eduroam.admin.gr net.gr</td>
<td>radi us</td>
<td>2</td>
<td>&lt;redacte d&gt;</td>
<td>6</td>
<td>&lt;redacte d&gt;</td>
<td>0</td>
<td>1813</td>
<td>1812</td>
<td>any</td>
<td>auth+acct</td>
</tr>
</tbody>
</table>

### Name i18n

<table>
<thead>
<tr>
<th>lang</th>
<th>id</th>
<th>name</th>
<th>object_id</th>
<th>content_type_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>en</td>
<td>1</td>
<td>GRNET</td>
<td>1</td>
<td>48</td>
</tr>
<tr>
<td>en</td>
<td>2</td>
<td>Greek Research and Technology Network - GRNET S.A.</td>
<td>2</td>
<td>44</td>
</tr>
<tr>
<td>el</td>
<td>4</td>
<td>Εθνικό Δίκτυο Έρευνας και Τεχνολογίας - ΕΔΕΤ A.E.</td>
<td>2</td>
<td>44</td>
</tr>
<tr>
<td>en</td>
<td>6</td>
<td>Greek Research and Technology Network - Headquarter s</td>
<td>2</td>
<td>42</td>
</tr>
<tr>
<td>el</td>
<td>8</td>
<td>Εθνικό Δίκτυο Έρευνας και Τεχνολογίας - Εδρα</td>
<td>2</td>
<td>42</td>
</tr>
<tr>
<td>en</td>
<td>10</td>
<td>Greek Research and Technology Network - Arkadias branch office</td>
<td>4</td>
<td>42</td>
</tr>
<tr>
<td>el</td>
<td>12</td>
<td>Εθνικό Δίκτυο Έρευνας και Τεχνολογίας - Παράρτημα Αρκαδίας</td>
<td>4</td>
<td>42</td>
</tr>
</tbody>
</table>

### Realm

<table>
<thead>
<tr>
<th>address_city</th>
<th>country</th>
<th>ts</th>
<th>address_street</th>
<th>id</th>
<th>stype</th>
</tr>
</thead>
<tbody>
<tr>
<td>Athens, Greece</td>
<td>gr</td>
<td>2012-11-19 15:51:25</td>
<td>Leoforos Mesogeion 56</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

### Service Loc

<table>
<thead>
<tr>
<th>address _city</th>
<th>transp_p roxy</th>
<th>SSD</th>
<th>institutio n_id</th>
<th>longitu de</th>
<th>wir ed</th>
<th>enc_level</th>
<th>N AT</th>
<th>IP v6</th>
<th>latitude</th>
<th>port_res trict</th>
<th>address_ street</th>
<th>id</th>
<th>AP_no</th>
</tr>
</thead>
<tbody>
<tr>
<td>Athens</td>
<td>0</td>
<td>eduro am</td>
<td>2</td>
<td>23.7657</td>
<td>7200</td>
<td>0</td>
<td>WPA2/ AES</td>
<td>0 1</td>
<td>37.9870</td>
<td>9500</td>
<td>1</td>
<td>Mesogion 56</td>
<td>2</td>
</tr>
<tr>
<td>Athens</td>
<td>0</td>
<td>eduro am</td>
<td>2</td>
<td>23.7664</td>
<td>7500</td>
<td>0</td>
<td>WPA2/ AES</td>
<td>0 1</td>
<td>37.9879</td>
<td>5000</td>
<td>1</td>
<td>Arkadias 31</td>
<td>4</td>
</tr>
</tbody>
</table>

### Uri i18n

<table>
<thead>
<tr>
<th>URL</th>
<th>lang</th>
<th>type</th>
<th>id</th>
<th>object_id</th>
<th>content_type_id</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.eduroam.gr">http://www.eduroam.gr</a></td>
<td>el</td>
<td>info</td>
<td>1</td>
<td>1</td>
<td>48</td>
</tr>
<tr>
<td><a href="http://www.eduroam.gr">http://www.eduroam.gr</a></td>
<td>en</td>
<td>info</td>
<td>2</td>
<td>1</td>
<td>48</td>
</tr>
<tr>
<td><a href="http://www.eduroam.gr">http://www.eduroam.gr</a></td>
<td>el</td>
<td>policy</td>
<td>3</td>
<td>1</td>
<td>48</td>
</tr>
<tr>
<td><a href="http://www.eduroam.gr">http://www.eduroam.gr</a></td>
<td>en</td>
<td>policy</td>
<td>4</td>
<td>1</td>
<td>48</td>
</tr>
</tbody>
</table>

### Cat Enrollment

<table>
<thead>
<tr>
<th>id</th>
<th>cat_inst_id</th>
<th>inst_id</th>
<th>url</th>
<th>cat_instance</th>
<th>ts</th>
<th>applier_id</th>
</tr>
</thead>
</table>

### References

[1] eduroam in a nutshell: [https://confluence.terena.org/display/H2eduroam/eduroam+in+a+nutshell](https://confluence.terena.org/display/H2eduroam/eduroam+in+a+nutshell)

[2] However the provision of accounting data has been superseded by the F-Ticks mechanism in recent years: [http://monitor.eduroam.org/f-ticks/about.php](http://monitor.eduroam.org/f-ticks/about.php)

[3] [https://www.eduroam.gr/](https://www.eduroam.gr/)

[4] [https://cat.eduroam.org/](https://cat.eduroam.org/)

[5] [https://cloudpebble.net/](https://cloudpebble.net/)

[6] [https://lists.grnet.gr/wws/info/djnro](https://lists.grnet.gr/wws/info/djnro)


[9] [https://member.eduroam.net.nz/](https://member.eduroam.net.nz/)

[10] [http://www.tertiaryictconference.co.nz/page/198/programme](http://www.tertiaryictconference.co.nz/page/198/programme)
**Vitae**

**Leonidas Poulopoulos** received his Diploma in Electrical and Computer Engineering from the University of Patras in 2005 and his M.Sc degree on Computer Science from the Department of Computer Engineering and Informatics (University of Patras) in 2010. Currently, he is (with) the development team of GRNET NOC. He designs and develops network management applications and web platforms and quite often, a mix of both. His job/interest/research profile can be found at: [http://www.linkedin.com/in/leopoul](http://www.linkedin.com/in/leopoul)

email: leopoul@noc.grnet.gr

phone: +30 210 7471096, +30 697 3845436

**Zenon Mousmoulas** studied law at the Democritus University of Thrace and the University of Athens, but shifted focus to ICT. He has been with GRNET since 2003, where he has participated in a wide range of national and European projects, undertaking both technical and management responsibilities. He holds the position of Network Applications Developer and, as of 2009, he is also a member of the in-house operations team (GRNET NOC). He has in-depth experience in a wide range of technologies, from networking to systems, identity and multimedia. Among other tasks, he is developing and piloting live video applications and services; he is technically responsible for eduroam in Greece and has been involved in all aspects of identity federation (GRNET AAI and eduGAIN). He has been a member of the TERENA Networking Conference program committee and has chaired TERENA's task force on applied media. In the past he has also worked on legal and promotion issues of free and open source software, having essentially helped in the establishment of the Greek Free Software Society and the Creative Commons affiliate organization in Greece.

email: zmousm@noc.grnet.gr

phone: +30 210 7474244, +30 697 7780854