SPHINX Application Programming Interface for Third Parties

User Manual

SPHINX

A Universal Cyber Security Toolkit for Health-Care Industry

Table of contents

1 Introduction 3
2 Installation/Deployment 3
2.1 Prerequisites and hardware 3
2.2 Configuration 4
3 Operation and features 4
3.1 Third-Party Management Functions 4
3.2 Third-Party Services Functions 15
3.3 Basic Case Examples 16

Table of figures

Figure 1: The S-API Concept 3
Figure 2: S-API User Login 5
Figure 3: S-API User Creation 6
Figure 4: S-API Dashboard Overview 7
Figure 5: S-API User Profile Menu 7
Figure 6: S-API Individual User Profile 8
Figure 7: S-API Company User Profile 8
Figure 8: S-API User Company Profile 9
Figure 9: S-API Dashboard 10
Figure 10: S-API Services 10
Figure 11: S-API SPHINX Service Information Details 11
Figure 12: S-API Usage Log 12
Figure 13: S-API Subscription Plans 13
Figure 14: S-API Support - Tickets 13
Figure 15: S-API Support - Add Ticket 14
Figure 16: S-API Support - Ticket Details 14
Figure 17: S-API Support - Contacts 15

List of tables

Table 1 SPHINX Services for Third Parties 15

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

1 Introduction

The SPHINX Application Programming Interface for Third Parties (S-API) enables third-party solution providers to access and interact with the SPHINX Platform tools and its services. Subject to authentication, authorisation and using end-to-end encryption, S-API exposes the advanced cybersecurity functionalities implemented by SPHINX tools anywhere anytime.

The S-API concept is presented in Figure 1.

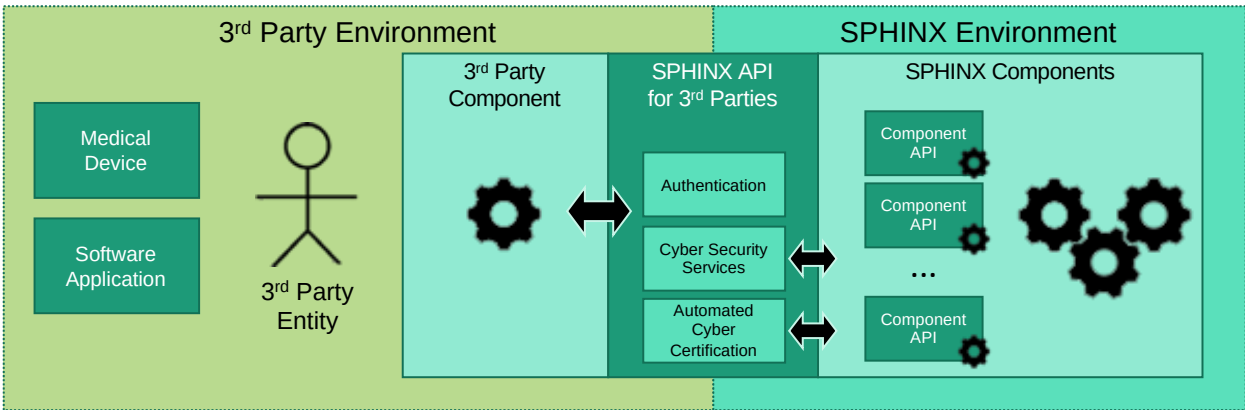

Figure 1: The S-API Concept

A particularly important feature in S-API consists in the delivery to third-parties of the SPHINX device certification service. Specifically, S-API may be used by medical device manufacturers (constrained hardware running specialised software or firmware) and software providers (specialised clinical software applications and solutions) to access the SPHINX Sandbox and receive assurance that the device, software and services are SPHINX-compliant and certified, therefore becoming trusted assets in a SPHINX-secured information technology (IT) ecosystem.

2 Installation/Deployment

Currently, S-API is provided as a service accessible in the cloud and it is operated by EDGENEERING. This deployment setup provides the most flexible and easy-to-maintain option for users, that only need to be concerned with the use of the service. EDGENEERING ensures the S-API tool's availability and maintenance.

2.1 Prerequisites and hardware

S-API runs on common hardware with the following specifications:

- CPU: 2GHz or higher;
- RAM: 4GB or higher;
- HDD: 50GB or higher.

S-API runs on Ubuntu 20.04 LTS and uses open source packages. The installation uses a Debian-generated (deb) package.

For the purposes of this document, the S-API instance to be used is available at:

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

- https://sphinx.edgeneering.eu/.

2.2 Configuration

S-API is provided as a cloud service to users using a fixed IP address. For S-API to work adequately, users need to setup network access from the S-API's IP address to the SPHINX services running in the user network. In this context, S-API provides a bridge between the SPHINX Environment and the user environment.

3 Operation and features

The S-API component provides the following primary functions, as defined in deliverable D3.6:

- **Third-Party Management Functions**, allowing third-party users to create and manage their account, providing information concerning their entity (personal, business or both) and select their appropriate **subscription plan**. Third-parties can also delete their account (and all associated data) at any time;
- **Third-Party Services Functions**, comprising:
  - **Third-Party Service Access Functions**, allowing third-parties to programmatically access functionalities provided by SPHINX services, including receiving notifications; and
  - **Third-Party SPHINX Certification Functions**, allowing access to the SPHINX Sandbox in order to validate and receive SPHINX compliance and certification reports concerning a third-party device, software or services.

3.1 Third-Party Management Functions

S-API provides a dedicated webpage for users with the following functions:

- S-API Login;
- Creation of User Account;
- Management of User Profile;
- S-API Dashboard;
  - Service Usage Overview, including charts;
- Services;
- Usage Log;
- Subscription Plans;
- Support.

An overview is provided concerning the above functions.

S-API Login

The **S-API Login** page can be accessed by opening the following URL using a web browser:

- https://sphinx.edgeneering.eu/.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

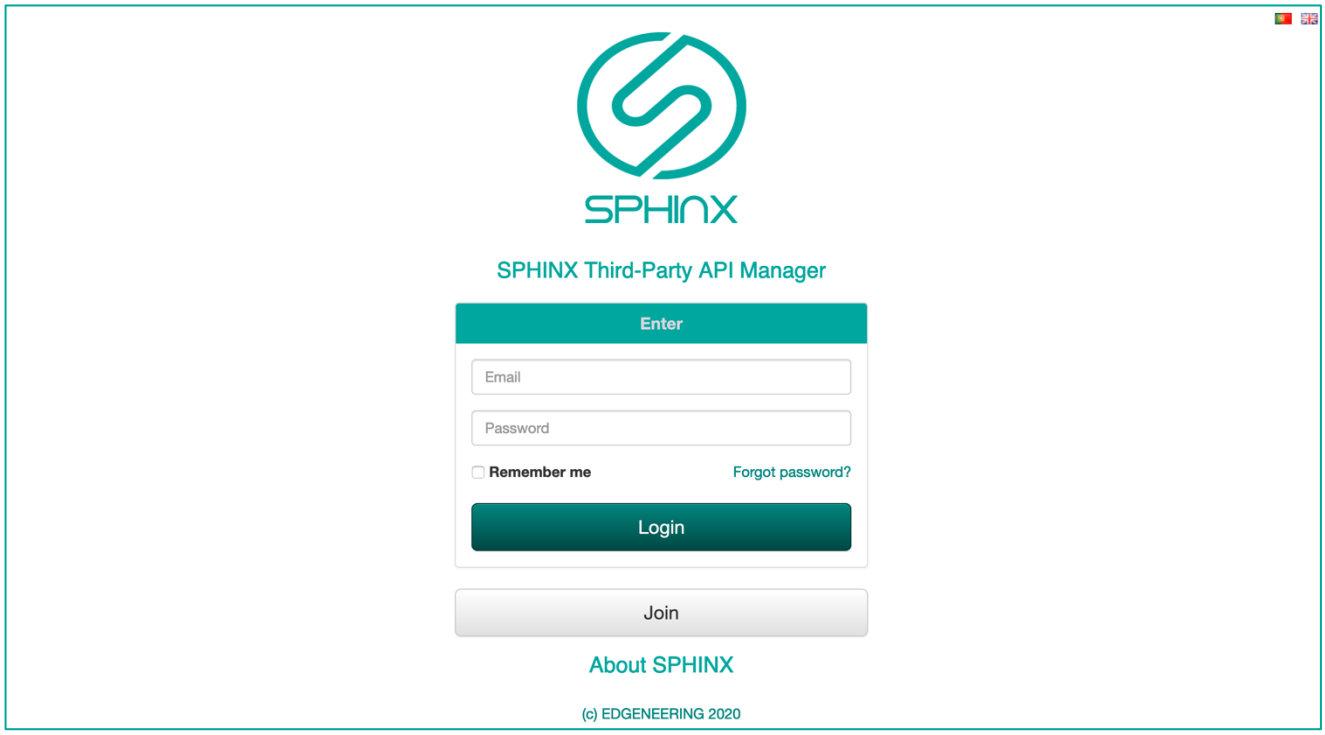

Figure 2: S-API User Login

If the user has an account, it should insert the credentials (username/email and password) and click on login.

The user may request a password reset by clicking on “Forgot Password”.

Creation of User Account

A user account may be created by accessing

- https://sphinx.edgeneering.eu/.

and clicking on “Join”. The following page opens.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

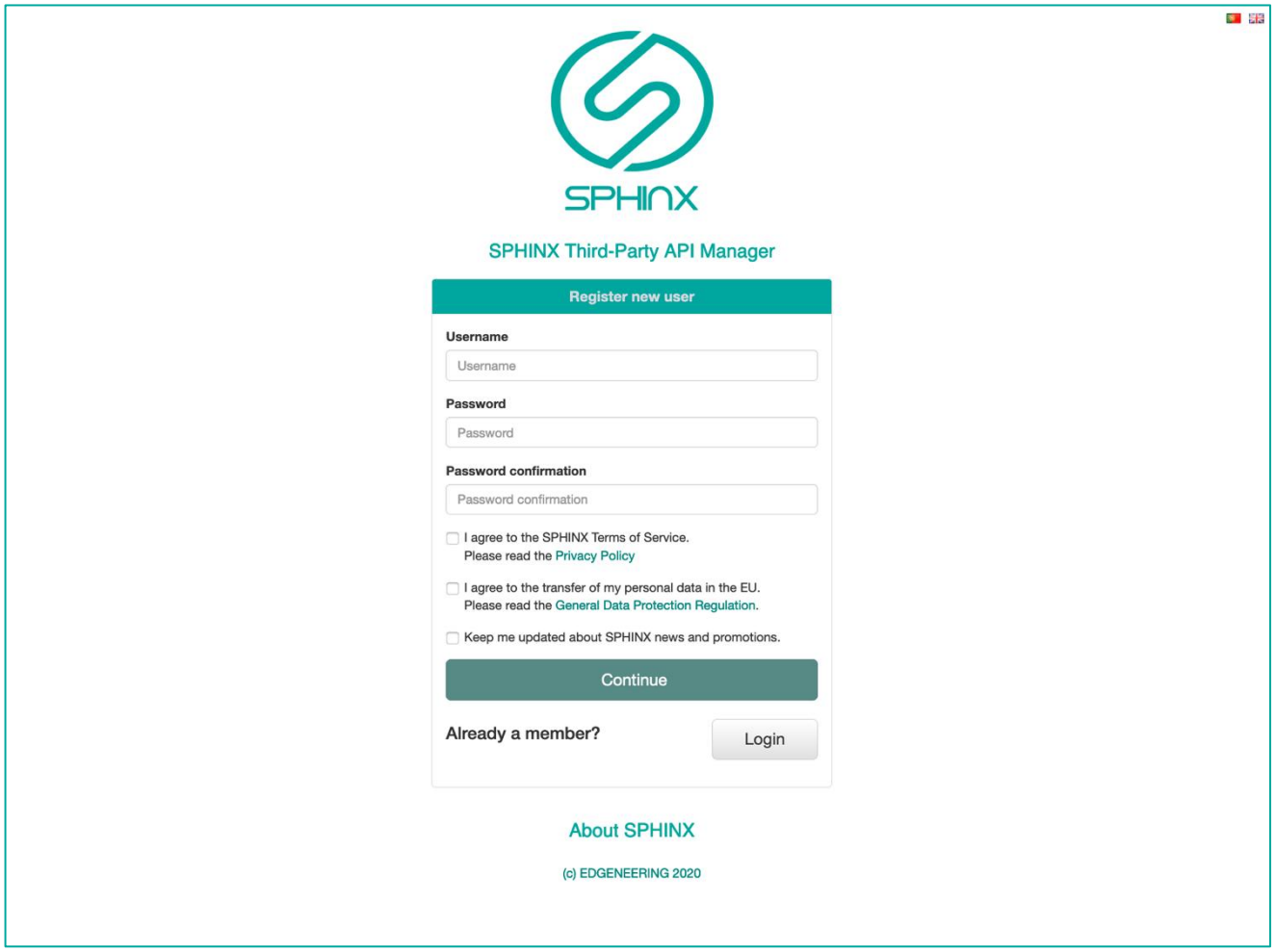

Figure 3: S-API User Creation

After inserting the username (valid email) and password, the user is required to agree with the "Privacy Policy" and "GDPR" terms, before proceeding. After these options are selected, the use should click on "Continue".

After clicking on "Continue", the user receives an email confirmation with a link to activate the account.

A valid email is required to create an account.

After the registration process, the user is required to confirm the registration by clicking on the link in the email sent by S-API to the user's email inbox.

Only after completing the email confirmation process, may the user login into S-API.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

S-API Display

After a successful login, S-API opens by default the Dashboard page that provides an overview of the user's options when using the S-API tool.

All the S-API pages present the following main spaces:

- the "Top Menu", displaying configuration menu options related with (1) language selection, (2) theme selection and (3) user profile menu;
- the "Left Menu", displaying the navigation menu for the tool, accessing the different S-API functions;
- the "Display Area", displaying information related with the selected function.

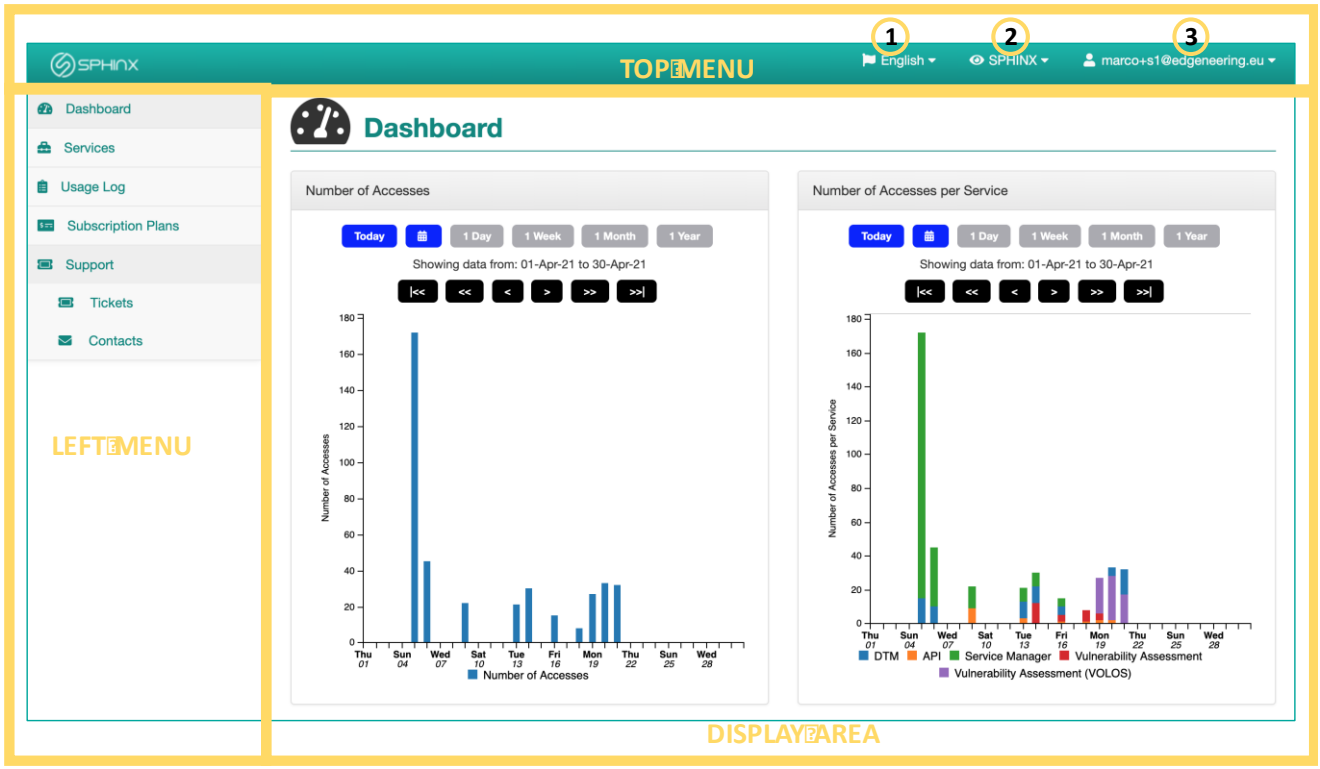

Figure 4: S-API Dashboard Overview

Management of User Profile

S-API considers two types of users: individual users and collective users (businesses).

The user profile can be managed and edited by using the options in the configuration menu in the "Top space" and clicking on (3) user profile menu and selecting "Profile".

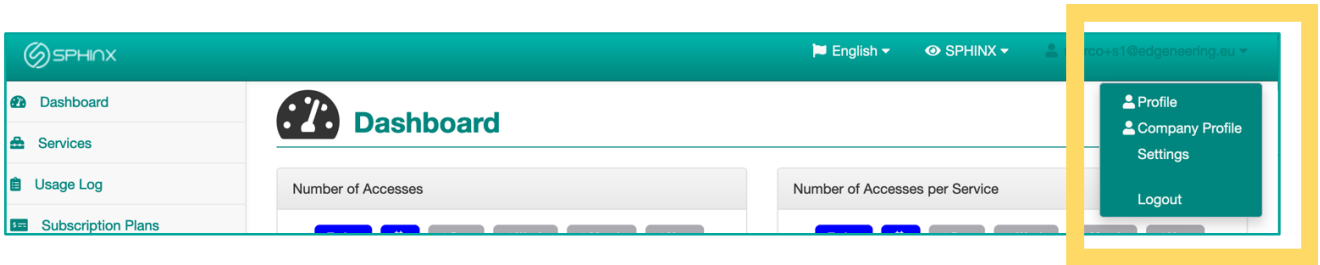

Figure 5: S-API User Profile Menu

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

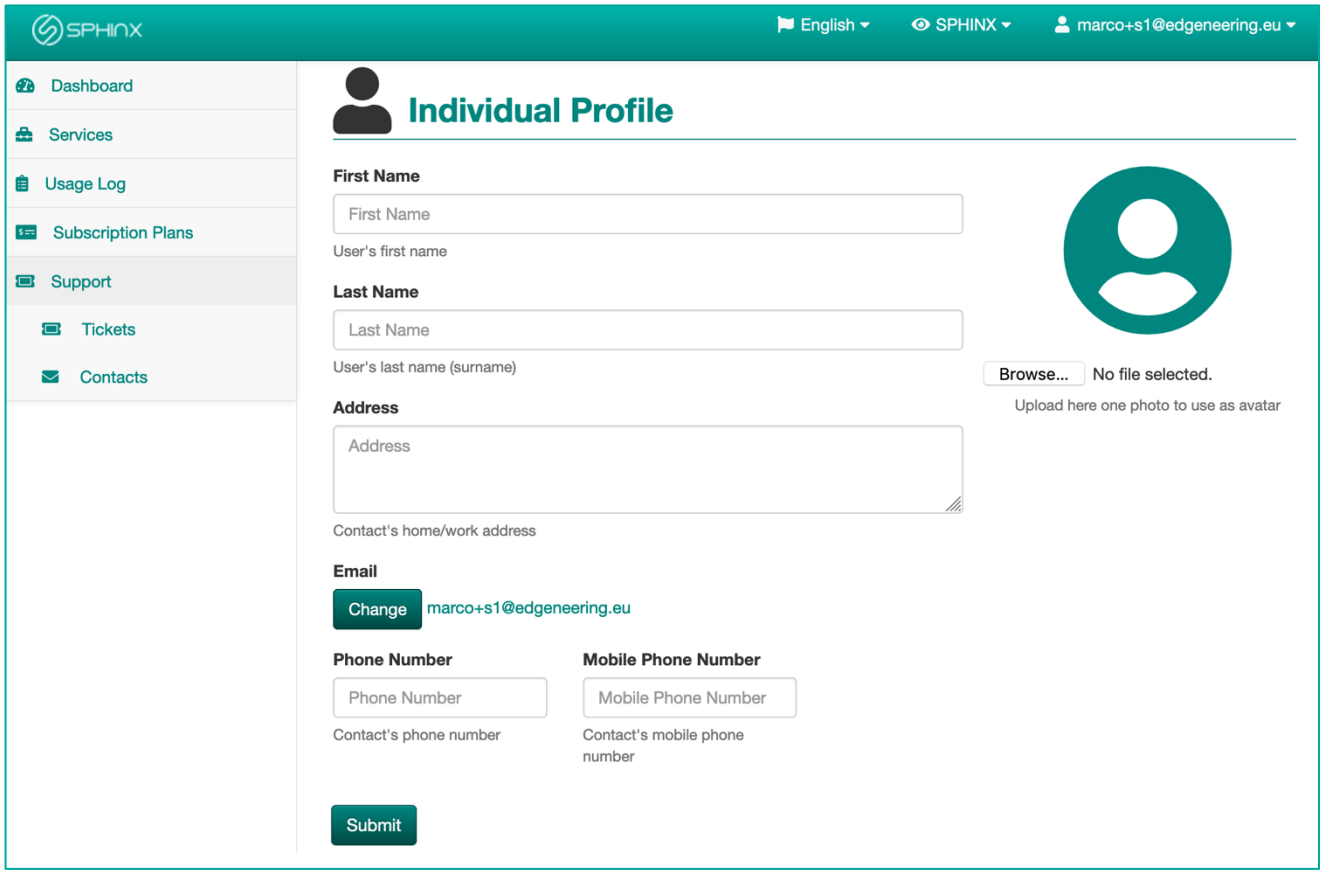

Figure 6: S-API Individual User Profile

The user may also change the user profile to a business user, by clicking on the (3) user profile menu and selecting “Company Profile”.

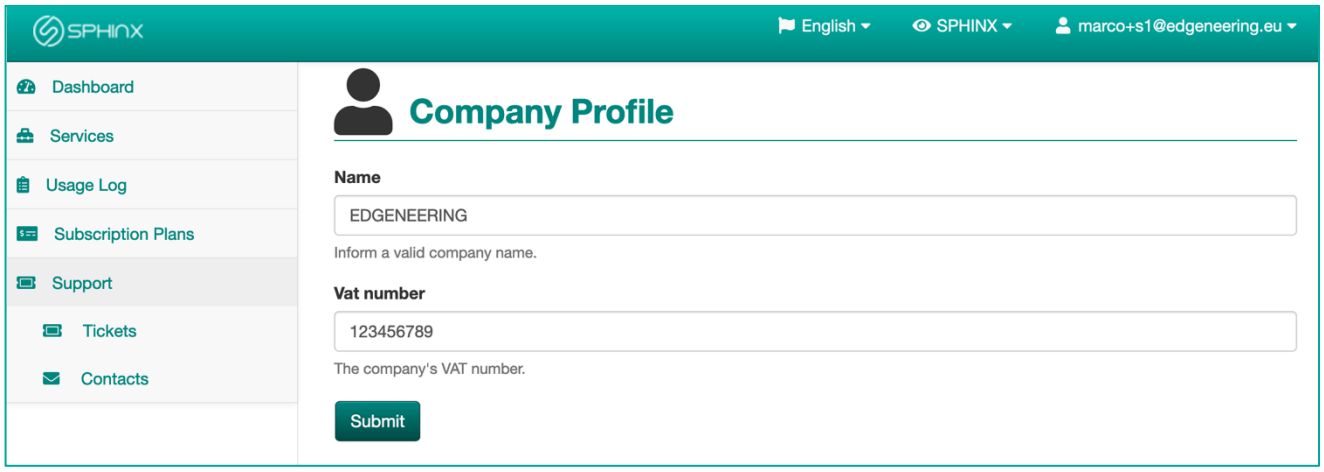

Figure 7: S-API Company User Profile

Finally, the user may change the settings related with email, password and profile (individual or company) by clicking on the (3) user profile menu and selecting “Settings”.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

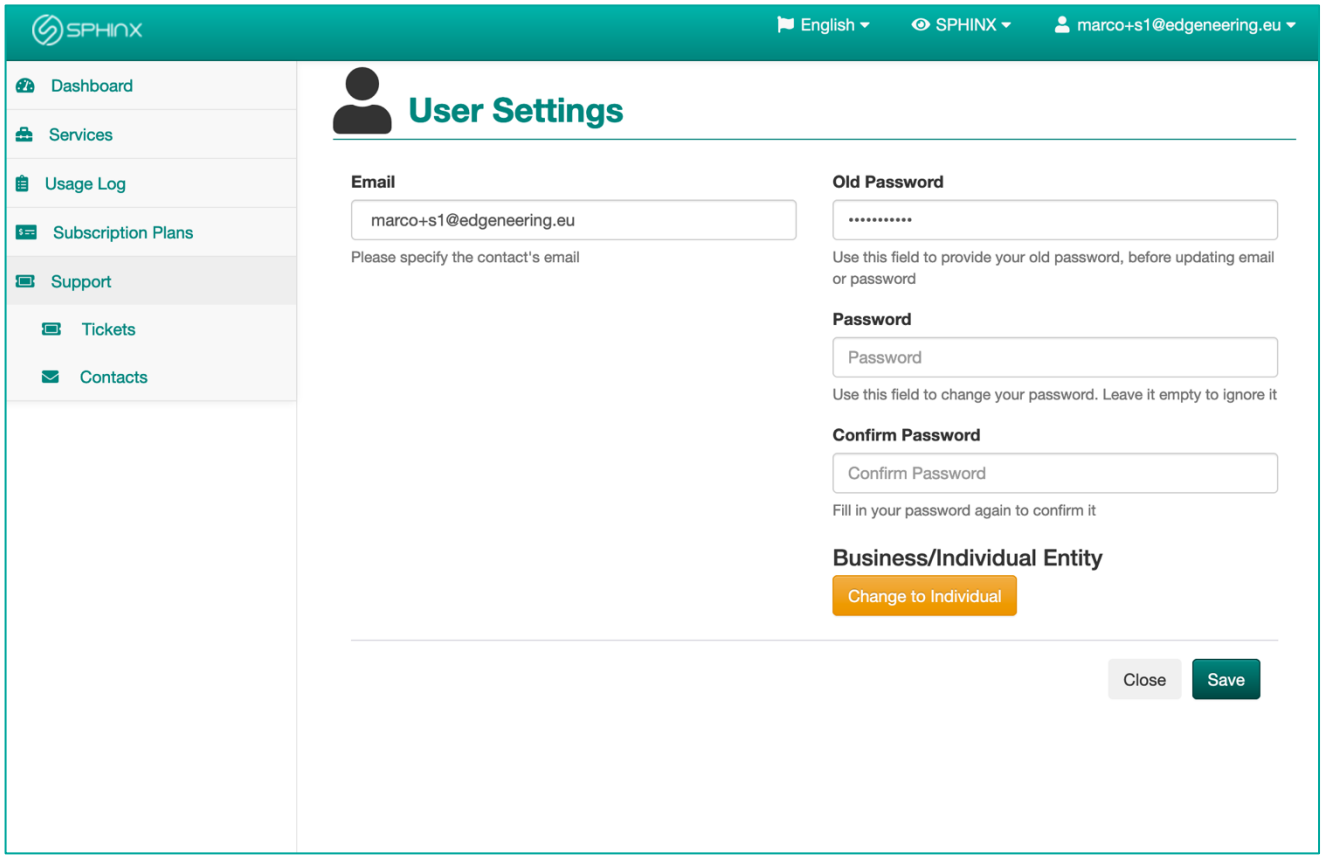

Figure 8: S-API User Company Profile

Change of email and password requires email confirmation.

Only after completing the email confirmation process, the user may login into S-API.

S-API Dashboard

The S-API **Dashboard** page provides an overview of the user's utilisation of the S-API tool.

It is the page that opens after a successful login and it is always accessible via the navigation menu in the "Left Menu".

The S-API **Dashboard** page displays:

- Charts with overall S-API usage (number of accesses) and SPHINX services' usage (number of accesses per service). These charts support the selection of specific timeframes, as well as the selection of the different SPHINX services to visualise;
- List of the available SPHINX Services accessible through S-API.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

Figure 9: S-API Dashboard

Services

The **Services** page displays the list of available SPHINX Services accessible through the S-API.

The **Services** information page is always accessible via the navigation menu in the “Left Menu”.

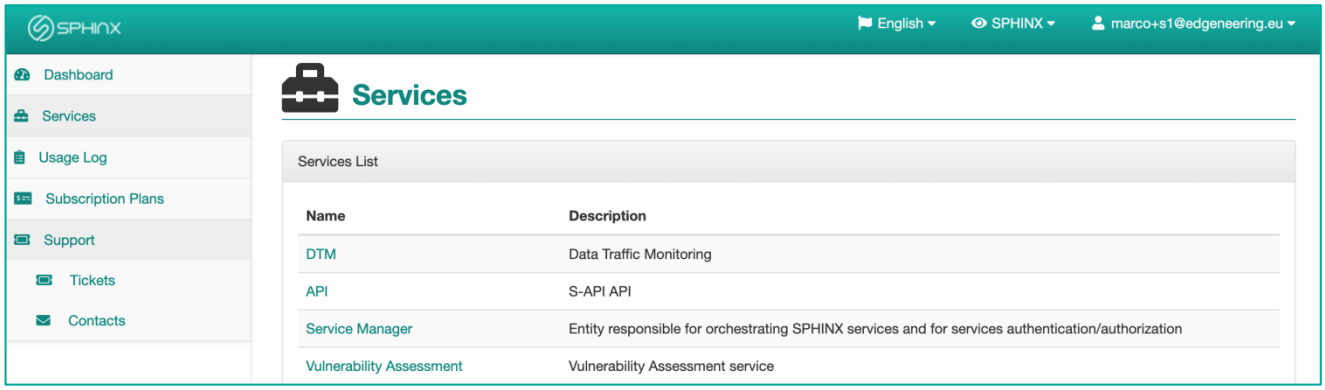

Figure 10: S-API Services

Details concerning a specific SPHINX Service may be accessed by clicking on the respective service.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

A new page opens, dedicated to the selected SPHINX Service and providing the following information:

- The name of the SPHINX Service;
- The short description of the SPHINX Service;
- Information required to programmatically access the SPHINX Service, namely “Client Id” and “Client Secret”.

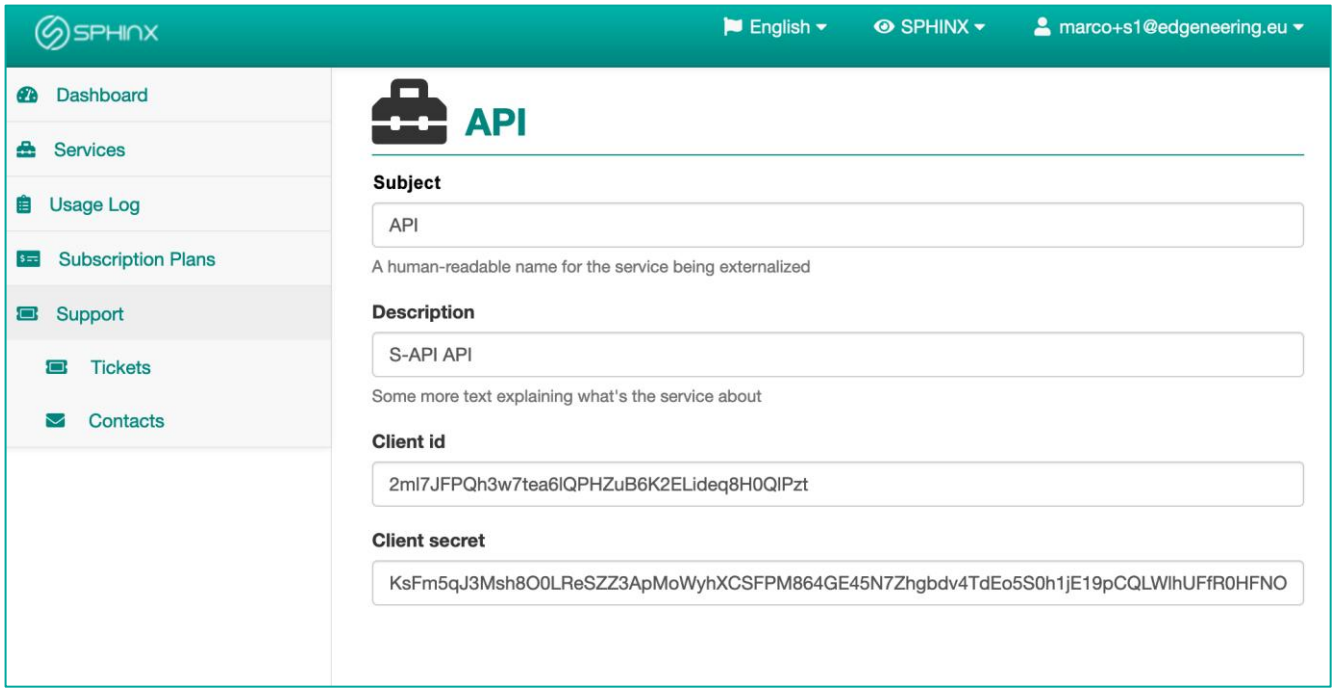

Figure 11: S-API SPHINX Service Information Details

The **Services** page displays information needed to programmatically access the specific SPHINX service.

The user will need the following information: “Client Id” and “Client Secret”.

Usage Log

The **Usage Log** page displays information concerning the utilisation of the available SPHINX services accessible through S-API. This information includes:

- The date and time of access;
- The service’s name;
- The service operation;
- The service’s endpoint;
- The status code of the request as per HTML guidelines[1] (e.g., 200 indicates success, 400 indicates error, 500 indicates server error).

[1] https://developer.mozilla.org/en-US/docs/Web/HTTP/Status.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

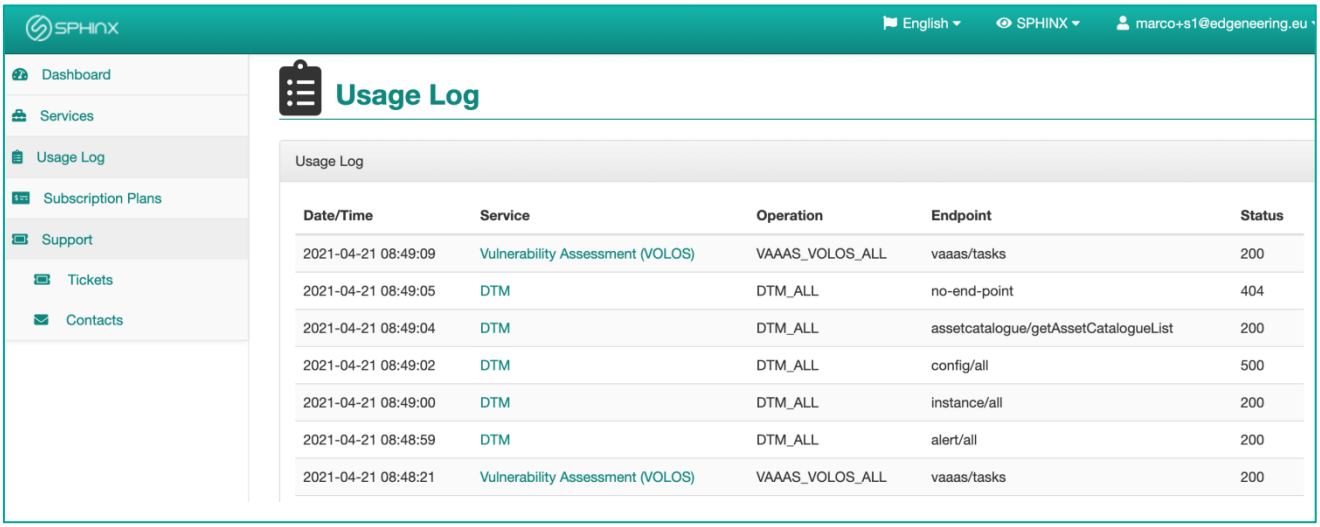

Figure 12: S-API Usage Log

The user may order the fields of information in an ascending or descending order by clicking on the respective header row field.

The **Usage Log** page is always accessible via the navigation menu in the "Left Menu".

Subscription Plans

S-API provides access to enabled SPHINX Services, and the usage of these service is limited, depending on the subscription plan selected by the user.

In the **Subscription Plans** page, the user may see the different subscription plans offered by S-API and the subscription plan already selected.

Also, at any time, the user may select a different subscription plan that best suits own needs.

Currently, S-API offers 3 subscription plans:

- Basic;
- Professional;
- Premium.

The **Subscription Plans** page is always accessible via the navigation menu in the "Left Menu".

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

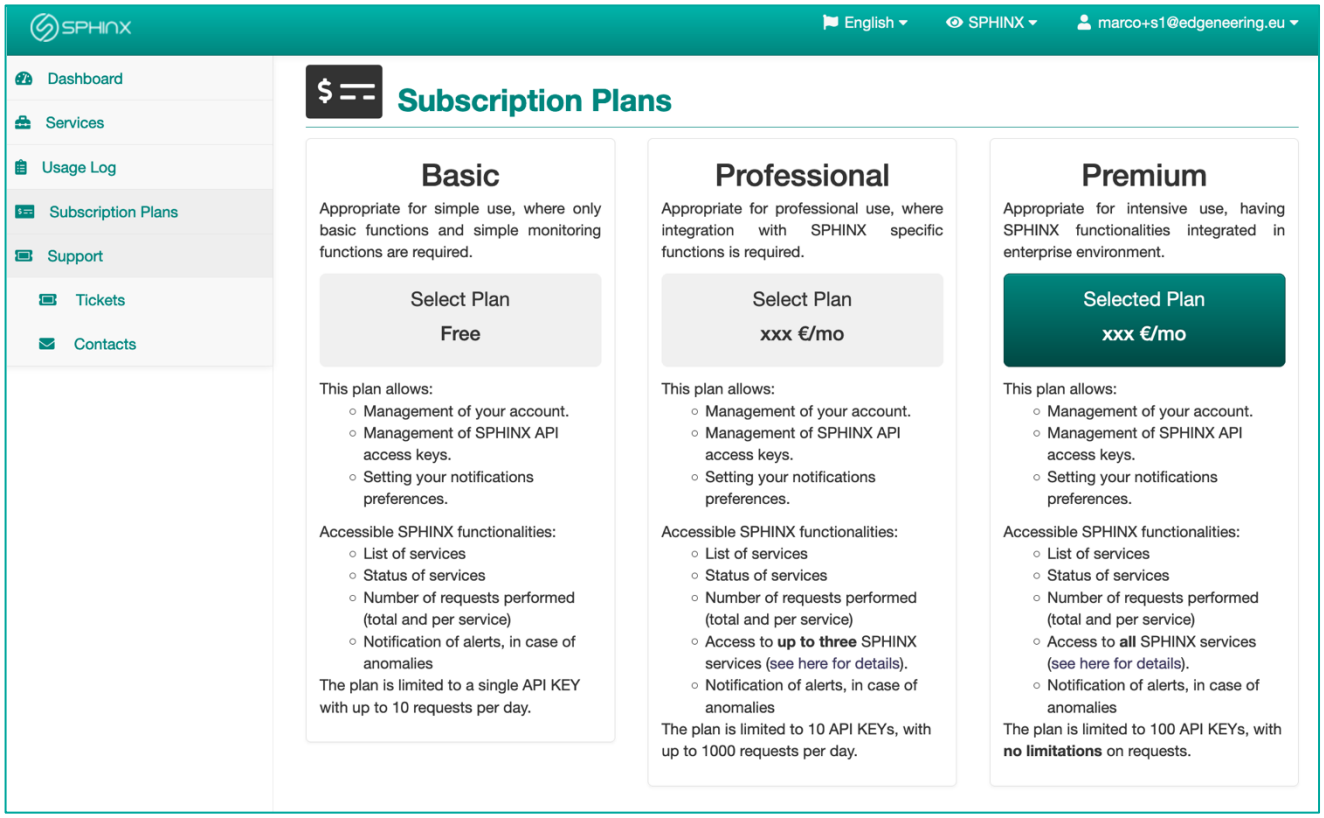

Figure 13: S-API Subscription Plans

Support

S-API provides technical support services to users in order to ensure a smooth and optimal use of the service. Users may report service problems or errors or technical difficulties experienced through tickets that are then managed the S-API technical support team.

The **Support** page is always accessible via the navigation menu in the “Left Menu”.

The **Support** page presents the **Tickets** page and the **Contacts** page.

The **Tickets** page presents the list of tickets issued by the user. This list displays the ticket’s subject and it includes the date the ticket was created, the ticket’s owner identification, the name of the technical team assigned to solve the reported issue. The user may also close the ticket by clicking on the “Close” button.

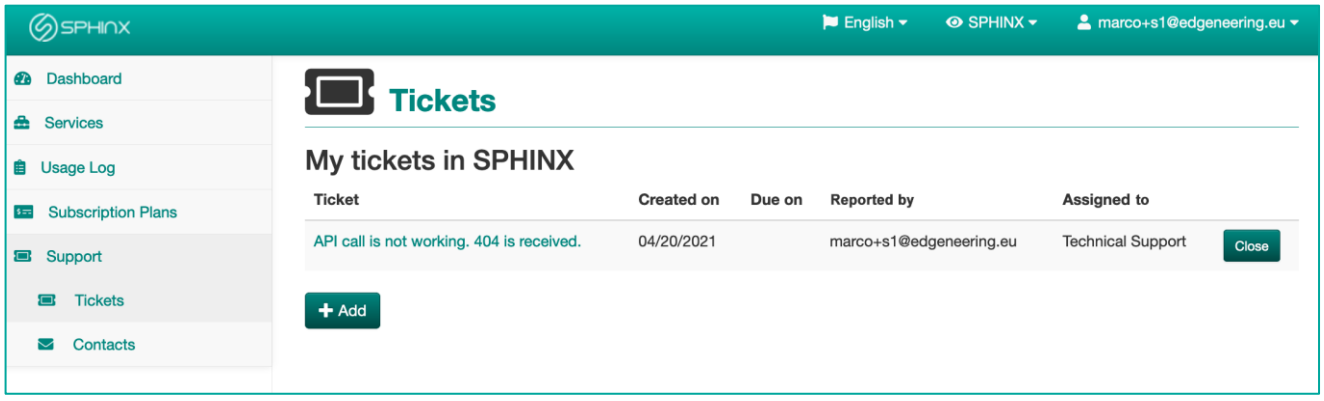

Figure 14: S-API Support - Tickets

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

The user may create a ticket by clicking on the “Add” button. Details of the ticket may also be inserted, such as subject and description.

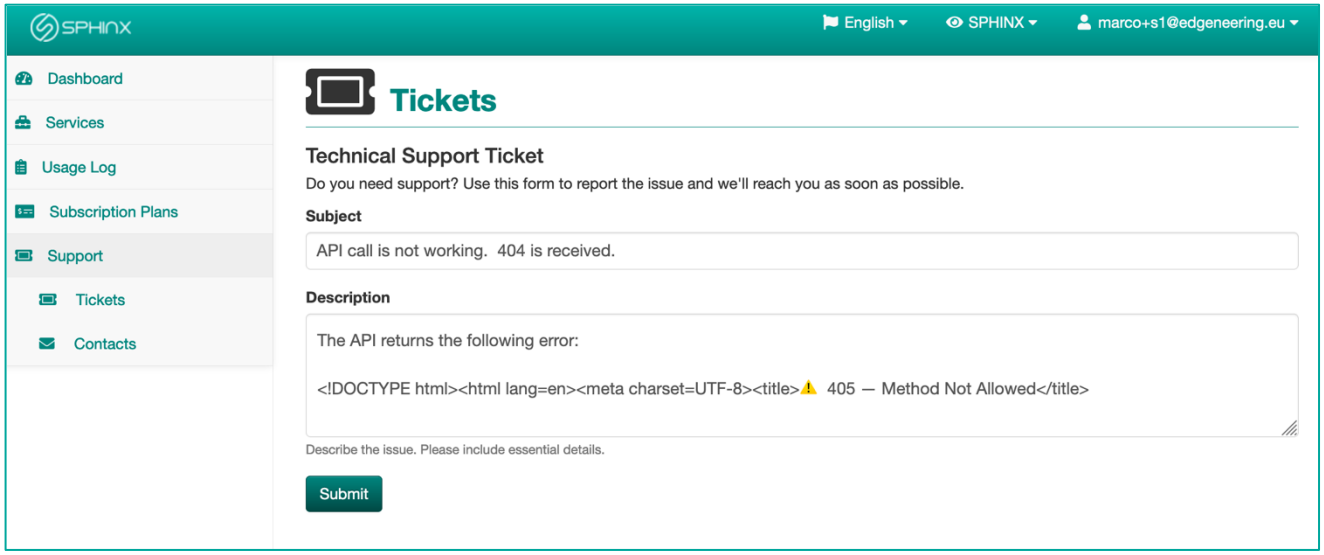

Figure 15: S-API Support - Add Ticket

Users are recommended to use a short but descriptive text on “Subject” and include as many details as possible in “Description”, in order to facilitate a proper analysis of the issue by the technical support team.

The status and details of a ticket may be accessed by clicking on the ticket’s subject in the tickets list.

The user may then view the ticket’s description, the identity of who was assigned to handle the request, the date when the request should be solved and the indication of the date when the ticket has been completed (if not completed, it would display “No”). The ticket may also include the comments provided by the technical team handling the ticket.

The user may “Close” the ticket, once the issue has been solved. The user may “Delete” the ticket at any time.

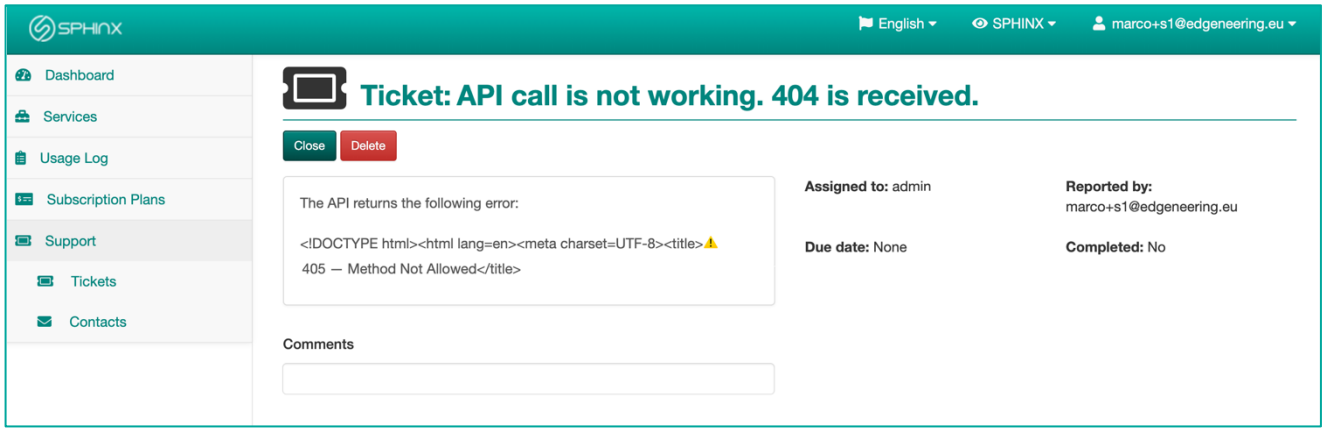

Figure 16: S-API Support - Ticket Details

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

S-API's technical support service is also available through email. The **Contacts** page displays the list of email contacts associated with specific technical support areas that may be used to address a service issue, technical difficulties or commercial queries.

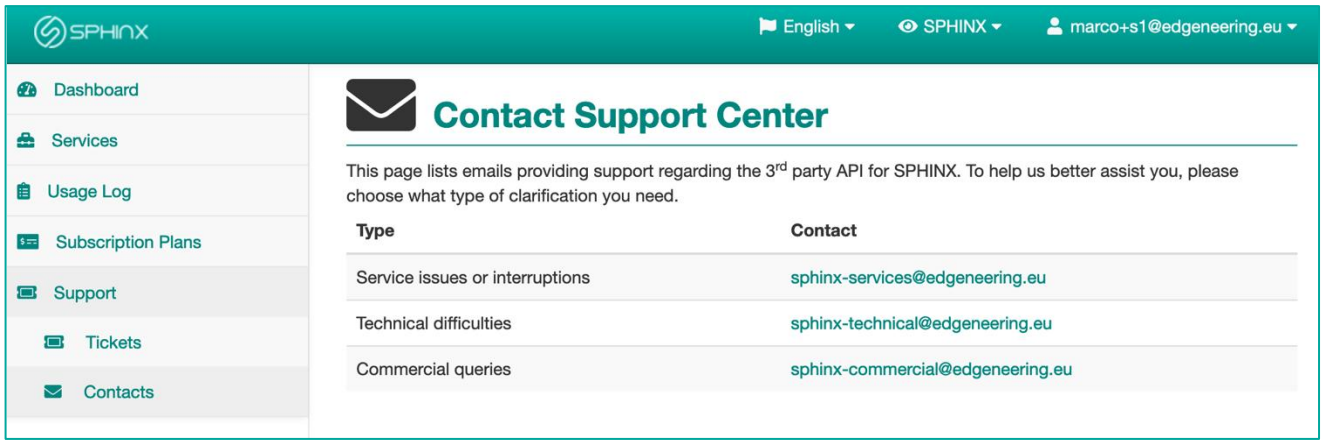

Figure 17: S-API Support - Contacts

3.2 Third-Party Services Functions

The S-API Third-Party Services Functions enable the programmatic access to SPHINX services anywhere anytime, automating the use of the available SPHINX services and enabling the development of service extensions (possibly new services).

The Third-Party Services Functions programmatic functions are accessed using two main modules:

- The S-API Client Authentication, based on OAuth2.0, providing a secure and controlled environment for users to access SPHINX services;
- The S-API SPHINX Services, allowing users to access available SPHINX Services exposed by S-API. The Services' access is categorised as (1) Access to SPHINX Services and (2) Certification Functions. This access follows a RESTful API approach, offering maximum flexibility for users to interact with S-API.

The following SPHINX services support interfaces for the S-API tool, rendering their functionality available to third-parties:

SPHINX Service	SPHINX Service Functions	Certification Functions
AD	•	
AP	•	
BBTR	•	
DTM	•	
FDCE	•	
HE	•	
SB	•	
SIEM	•	•
VAaaS	•	•

Table 1 SPHINX Services for Third Parties

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

S-API currently provides a client demonstrator developed in Python language. Future versions are planned for Android platform (in Java language) and web-based platforms.

The S-API client source-code is available at: https://sphinx-repo.intracom-telecom.com/sphinx-project/sphinx-api-for-third-parties/sphinx-sapi-client-python.

3.3 Basic Case Examples

This subsection describes a set of basic examples detailing how to access S-API using the programmatic mode, using the S-API client demonstrator presented in section 3.2.

To run the S-API client examples, appropriate credentials should be used.

Moreover, accessing a specific service requires retrieving "Client Id" and "Client Secret" information from the S-API Service information details page (**Figure 11**).

The available source-code includes several examples to call the SPHINX Services accessible through S-API. However, it is recommended to update it with the appropriate user's credentials.

Pre-conditions

- Bash command shell with Internet access;
- Python3.6 with "oauthlib" library installed;
- SPHINX user account created.

Case 1: Retrieve list of S-API registered services

Objective: The user wishes to retrieve the list of available SPHINX services registered in S-API, including the service credentials to use when calling each service.

Steps: The user should issue a GET call to **https://api.sphinx.edgeneering.eu/api/API/services**

For this, the settings file "settings_SAPI.json" can be used as follows:

```
python sapi_client.py settings_SAPI.json
```

An excerpt of the resulting output, regarding the DTM service, is provided below:


```
{
  "service_id": 3,
  "client_id": "aMrgc5BfM37qmNjgNMVXPtekmo0KB16puIxEjiyn",
  "client_type": "confidential",
  "authorization_grant_type": "password",
  "client_secret":
"YxVS0VVVg0UH7TOJe0Kt6X2cpEBQi5DYQagVW296Tj6lU3EVnRJAdiEQqEGSd7NrU4PR4Nr1TRXg56BvY
sok8gSKMjz3N2Jx1OZX3mqohmWmrL99bXsI8koNn93jYDgY",
  "name": "DTM",
  "skip_authorization": false,
  "created": "2021-01-30T15:31:05.729673Z",
  "updated": "2021-01-30T15:59:07.524929Z",
  "created_at": "2021-01-30T15:29:40Z",
  "modified_at": "2021-01-30T15:29:43Z",
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

```
  "context": {
    "app_id": "DTM"
  },
  "external_id": "DTM",
  "service_name": "DTM",
  "service_description": "Data Traffic Monitoring",
  "base_url": "DTM",
  "external_url": "https://sphinx-kubernetes.intracom-telecom.com/sphinx/dtm/",
  "active": true,
  "user": 1,
  "application_ptr": 4
},
```


Case 2: Retrieve list of all DTM generated alerts

Objective: The user wishes to retrieve the list of alerts generated by the SPHINX DTM service.

Steps: The user should issue a GET call to **https://api.sphinx.edgeneering.eu/api/DTM/alert/all**

For this, the settings file "settings_DTM.json" can be used as follows:

```
python sapi_client.py settings_DTM.json
```

The script provides an output of the list of alerts generated by DTM. If no alerts exist, it outputs an empty JSON array "[]".

Case 3: Retrieve list of all VAaaS generated reports

Objective: The user wishes to retrieve the list of reports generated by the SPHINX VAaaS service.

Steps: The user should issue a GET call to

https://api.sphinx.edgeneering.eu/api/VAAAS_VOLOS/vaaas/reports

For this, the settings file "settings_VAaaS_VOLOS.json" can be used as follows:

```
python sapi_client.py settings_VAAAS_VOLOS.json
```

An excerpt of the resulting output is provided below:


```
{
  "status_code": "0",
  "result": "GET_ALL_REPORTS_SUCCESS",
  "more": "",
  "items": {
    "reports": [
      {
        "id": "bundle--f35476ca-83aa-47a2-a83b-961813738939",
        "assessment_date": "Tue Apr 20 06:13:46 2021",
        "start": "",
        "stop": "",
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

```
    "task_name": "10.0.100.101",
    "objects": [
      {
        "value": "10.0.100.101",
        "type": "ipv4-addr",
        "spec_version": "2.1",
        "id": "ipv4-addr--560fbd49-89fe-5fcb-ae5f-52a6ec87dd07"
      }
    ],
    "cvss_score": 4.5,
    "total_services": 0,
    "type": "bundle"
  },
  {
    "id": "bundle--42f03f53-46d1-4906-ad17-a231f7cd647f",
    "assessment_date": "Tue Apr 20 06:35:03 2021",
    "start": "1618900606",
    "stop": "1618900504",
    "task_name": "10.10.2.103",
    "objects": [
      {
        "value": "10.10.2.103",
        "type": "ipv4-addr",
        "spec_version": "2.1",
        "id": "ipv4-addr--16c2fdd0-e05b-5c12-9f6f-8fb7d30619fb"
      },
      {
        "value": "00:50:56:8C:F5:86",
        "type": "mac-addr",
        "spec_version": "2.1",
        "id": "mac-addr--d0a2f5a7-a8ca-52ec-b409-b7a8d267d589"
      },
      {
        "type": "relationship",
        "spec_version": "2.1",
        "id": "relationship--2aa3fc27-0a15-44e8-aa19-8c81026e02ac",
        "created": "2021-04-20T13:39:41.209434Z",
        "modified": "2021-04-20T13:39:41.209434Z",
        "source_ref": "ipv4-addr--16c2fdd0-e05b-5c12-9f6f-8fb7d30619fb",
        "relationship_type": "has",
        "target_ref": "mac-addr--d0a2f5a7-a8ca-52ec-b409-b7a8d267d589"
      },
      {
        "port": "135",
        "protocol": "tcp",
        "state": "open",
        "service_name": "msrpc",
        "service_product": "Microsoft Windows RPC",
        "service_cpe_list": [
          "cpe:\/o:microsoft:windows"
        ],
        "type": "x-discovered-service",
        "spec_version": "2.1",
        "id": "x-discovered-service--c985e4d7-4e7d-4bd8-81ac-b85b083d6b6c",
        "created": "2021-04-20T13:39:41.209658Z",
        "modified": "2021-04-20T13:39:41.209658Z"
      },
      {
        "port": "139",
        "protocol": "tcp",
        "state": "open",
        "service_name": "netbios-ssn",
        "service_product": "Microsoft Windows netbios-ssn",
        "service_cpe_list": [
          "cpe:\/o:microsoft:windows"
        ],
        "type": "x-discovered-service",
        "spec_version": "2.1",
        "id": "x-discovered-service--53910115-9291-4385-9f4d-7194f0daed63",
        "created": "2021-04-20T13:39:41.20984Z",
        "modified": "2021-04-20T13:39:41.20984Z"
      },
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

Case 4: Retrieve list of all VAaaS tasks

Objective: The user wishes to retrieve the list of tasks performed by the SPHINX VAaaS service.

Steps: The user should issue a GET call to

https://api.sphinx.edgeneering.eu/api/VAAAS_VOLOS/vaaas/tasks

For this, the settings file "settings_VAaaS_VOLOS.json" can be used as follows:

```
python sapi_client.py settings_VAAAS_VOLOS.json
```

An excerpt of the resulting output is provided below:


```
{
  "status_code": "0",
  "result": "GET_ALL_TASKS_SUCCESS",
  "more": "",
  "items": {
    "tasks": [
      {
        "desktop_pc": {
          "name": "desktop_pc",
          "target": "10.0.100.101",
          "processes": {
            "NSE": {
              "status": "ended",
              "etc": 0,
              "progress": 100,
              "remaining": 0
            },
            "Ping Scan": {
              "status": "ended",
              "etc": 0,
              "progress": 0,
              "remaining": 0
            }
          },
          "reports": {
            "1618899226": {
              "__NmapReport__": {
                "_nmaprun": {
                  "scanner": "nmap",
                  "args": "\/usr\/bin\/nmap -oX - -vvv --stats-every 1s -sV -PR -T4 --script
vulners 10.0.100.101",
                  "start": "1618899226",
                  "startstr": "Tue Apr 20 06:13:46 2021",
                  "version": "7.70",
                  "xmloutputversion": "1.04"
                },
```


Case 5: Remotely initiate a vulnerability assessment

Objective: The user wishes VAaaS to initiate a vulnerability assessment on IP "10.0.100.101".

Steps: The user should issue a POST call to

https://api.sphinx.edgeneering.eu/api/VAAAS_VOLOS/vaaas/tasks/start

The following payload must be provided.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

```
{
  "name": "S-API VAaaS on-demand request",
  "target": "10.0.100.101",
  "speed": 1
}
```


For this, the settings file “settings_VAaaS_VOLOS.json” can be used as follows:

```
python sapi_client.py settings_VAAAS_VOLOS.json
```

An excerpt of the resulting output is provided below:


```
{
  "status_code": "0",
  "result": "SCAN_NETWORK_STARTED",
  "more": "Started assessment for 10.0.100.101",
  "items": []
}
```


Case 5.1: Check the status of the initiated vulnerability assessment

Objective: The user wishes to check the status of the initiated vulnerability assessment (of Case 5).

Steps: The user should issue a POST call to

https://api.sphinx.edgeneering.eu/api/VAAAS_VOLOS/vaaas/tasks

The following payload must be provided.


```
{
  "name": "S-API VAaaS on-demand request"
}
```


For this, the settings file “settings_VAaaS_VOLOS.json” can be used as follows:

```
python sapi_client.py settings_VAAAS_VOLOS.json
```

An excerpt of the resulting output is provided below:


```
{
  "status_code": "0",
  "result": "GET_TASK_SUCCESS",
  "more": "",
  "items": {
    "task": {
      "name": "S_API_request",
      "target": "10.0.100.101",
      "processes": {
        "NSE": {
          "status": "ended",
          "etc": 0,
          "progress": 100,
          "remaining": 0
        },
        "Ping Scan": {
          "status": "ended",
          "etc": "1618990974",
          "progress": "100.00",
          "remaining": "0"
        }
      },
      "reports": {
```


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

...

}

Case 6: Receive notification when anomalies are detected

Objective: The user is travelling, but wishes to receive notifications when anomalies are detected by the SPHINX services.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.