



Tools for Legal, Regulatory, Privacy and Cybersecurity

Compliance

D6.6

Authors:

Eleni Panagou (UBI)

Kostas Mylonas (UBI)

Magda Foti (UBI)

Anastasis Tzoumpas (UBE)

Ferdinando Bosco (ENG)

Apostolos Kapetanios (ED)

Konstantinos Kotsalos (ED)

Responsible Partner	UBITECH ENERGY
Checked by WP leader	Vasilis Sakas (ED), 20/07/2023
Verified by the appointed Reviewers	Petr Barina (Schneider Electric), 27/07/2023
Approved by Project Coordinator	Padraic McKeever (Fraunhofer), 01/09/2023

Dissemination Level	Public
----------------------------	--------



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957739

Issue Record

Planned delivery date	31/07/2023
Actual date of delivery	01/09/2023
Status and version	V1.0

Version	Date	Author(s)	Notes
0.1	09/05/2023	UBI, UBE	Table of Contents
0.2	19/05/2023	UBI, UBE	Drafting of introduction
0.3	26/05/2023	UBI, UBE	Drafting of section 2
0.4	02/06/2023	UBI, UBE	Drafting of section 3
0.5	16/06/2023	UBI, UBE	Drafting of conclusions
0.6	30/06/2023	ED, EG	Contributions in section 2
0.7	6/07/2023	EG	Contribution in section 2
0.8	11/07/2023	ED	Contribution in section 2
0.9	21/07/2023	UBE	Final draft ready for review
1.0	31/08/2023	UBE	Final version ready for submission

Disclaimer:

All information provided reflects the status of the OneNet project at the time of writing and may be subject to change. All information reflects only the author's view and the European Climate, Infrastructure and Environment Executive Agency (CINEA) is not responsible for any use that may be made of the information contained in this deliverable.

About OneNet

The project OneNet (One Network for Europe) will provide a seamless integration of all the actors in the electricity network across Europe to create the conditions for a synergistic operation that optimizes the overall energy system while creating an open and fair market structure.

OneNet is funded through the EU's eighth Framework Programme Horizon 2020, "TSO – DSO Consumer: Large-scale demonstrations of innovative grid services through demand response, storage and small-scale (RES) generation" and responds to the call "Building a low-carbon, climate resilient future (LC)".

As the electrical grid moves from being a fully centralized to a highly decentralized system, grid operators have to adapt to this changing environment and adjust their current business model to accommodate faster reactions and adaptive flexibility. This is an unprecedented challenge requiring an unprecedented solution. The project brings together a consortium of over 70 partners, including key IT players, leading research institutions and the two most relevant associations for grid operators.

The key elements of the project are:

1. Definition of a common market design for Europe: this means standardized products and key parameters for grid services which aim at the coordination of all actors, from grid operators to customers;
2. Definition of a Common IT Architecture and Common IT Interfaces: this means not trying to create a single IT platform for all the products but enabling an open architecture of interactions among several platforms so that anybody can join any market across Europe; and
3. Large-scale demonstrators to implement and showcase the scalable solutions developed throughout the project. These demonstrators are organized in four clusters coming to include countries in every region of Europe and testing innovative use cases never validated before.

Table of Contents

1	Introduction	8
1.1	Task 6.6.....	8
1.2	Objectives of the Work Reported in this Deliverable	8
1.3	Report Outline	9
1.4	How to read this document.....	9
2	Requirements for Legal, Regulatory, Privacy and Cybersecurity Compliance	10
2.1	NISTIR 7628 Requirements.....	10
2.1.1	OneNet Monitoring and Analytics Dashboard	17
2.1.2	OneNet Decentralized Middleware and OneNet Connector	20
2.1.3	OneNet Orchestration Workbench	21
3	The OneNet Monitoring and Analytics Dashboard – Technical Characteristics.....	23
3.1	OneNet Monitoring and Analytics Dashboard Architecture	23
3.2	OneNet Monitoring and Analytics Dashboard – v1.0.....	23
3.2.1	OneNet Network Traffic & Endpoint Infrastructure Monitoring Tool	23
3.2.2	OneNet Data Analysis, Rating & Classification Tool	31
3.2.3	OneNet Authentication/Authorization and Administration	35
3.2.4	OneNet Monitoring and Analytics Dashboard GUI	40
4	Conclusions	46
5	References	47

List of Figures

Figure 1: WPs interdependencies	9
Figure 2: OneNet Monitoring and Analytics Dashboard service architecture	23
Figure 3: Log collection architecture	24
Figure 4: Logstash to Logstash direct HTTPS communication	24
Figure 5: Logstash to Logstash communication through a HTTP proxy	26
Figure 6: Example of sending requests through the Elastic Dev Console in Kibana	29
Figure 7: Visualization of number of records over one month in Kibana	29
Figure 8: Keycloak Admin Console: Main Authentication Realm	35
Figure 9: Keycloak Admin Console: OneNet Client	36
Figure 10: Keycloak Login Form	36
Figure 11: Keycloak Account Management: Personal Info	37
Figure 12: Keycloak Account Management: Applications page	37
Figure 13: Keycloak Account Management: Device Activity	38
Figure 14: Keycloak Account Management: Signing In page	38
Figure 15: Security Report	39
Figure 16: Dashboard login page	40
Figure 17: Dashboard register page	41
Figure 18: User menu	41
Figure 19: Account settings component	42
Figure 20: Access map chart	42
Figure 21: Last 24 hours (live)	43
Figure 22: Last 14 days	43
Figure 23: Data sent over time	44
Figure 24: Response codes over time	44
Figure 25: Anomaly detection	45
Figure 26: Connectors dropdown menu	45

List of Tables

Table 1: NISTIR 7628 requirements, related GDPR privacy principles and recommendations tailored to the OneNet Interoperable Network of Platforms	10
--	----

List of Abbreviations and Acronyms

Acronym	Meaning
API	Application Programming Interface
ASVS	Application Security Verification Standard
CIA	Confidentiality, Integrity and Availability
CORS	Cross-Origin Resource Sharing
DDoS	Distributed Denial of Service
DPIA	Data Protection Impact Assessment
DTO	Data Transfer Object
ENG	Engineering (Ingegneria Informatica)
ENISA	European Union Agency for Cyber Security
GDPR	General Data Protection Regulation
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IP	Internet Protocol
LDAP	Lightweight Directory Access Protocol
NIST	National Institute of Standards and Technology
PAM	Pluggable authentication module
RAM	Random-access memory
RFID	Radio Frequency Identification
SPA	Single Page Applications
SSH	Secure Socket Shell
UI	User interface
USB	Universal Serial Bus
VM	Virtual Machine
VPN	Virtual Private Network
WP	Work Package

Executive Summary

One crucial aspect of an online, decentralized platform such as OneNet is cybersecurity, which involves securing sensitive information, protecting against unauthorized access or data breaches, fortifying the developed components against vulnerabilities and ensuring the integrity and privacy of data generated by interactions and transactions throughout the OneNet Interoperable Network of platforms.

This report presents an assessment of the legal, regulatory and cybersecurity guidelines and principles followed by the OneNet project team members to safeguard the OneNet Interoperable Network of platforms. These measures include the following requirements: Physical access to machines hosting the OneNet Connector, Middleware, Orchestration Workbench and Monitoring and Analytics Dashboard shall be restricted to the OneNet project members, while remote access shall be allowed only with the use of SSH keys. Connector logs shall be collected, analyzed, and used for network monitoring and anomaly detection. Access, visitor management, passwords, document security, and personal computer usage must be governed by strict policies. Reporting of security incidents must be performed in a timely manner, while computing resources shall be securely destroyed at the end of their lifecycle. Finally, data protection should be ensured through DPIA assessments aligned with GDPR.

This report also presents the implementation of the OneNet Monitoring and Analytics Dashboard which aids in the preservation of cybersecurity within the OneNet solution. The current release of the OneNet Monitoring and Analytics Dashboard implements continuous monitoring of logs originating from the OneNet Connector and network traffic classification based on the Isolation Forest machine learning algorithm. The implementation includes a web UI providing monitoring capabilities and is supported by a Spring Boot back-end, the Keycloak Identity Manager and an anomaly detection model developed using Python and machine learning libraries. The dashboard provides access to monitoring and analytics functionality through a number of graphs displaying various statistics regarding Connector use and facilitates administration through the generation of a security report which enriches the results of the data analysis, rating and classification tool. It also offers account management functionality in cooperation with the capabilities offered by Keycloak.

The implementation of the aforementioned cybersecurity measures across all stages of development of the components constituting the OneNet Interoperable Network of Platforms has proven to be of vital importance. In particular, this work has yielded valuable insights into proactive threat and incident management strategies, technological advancements in the cybersecurity domain and potential cybersecurity challenges a smart grid information system may face. These insights have contributed to the safeguarding of the OneNet Interoperable Network of Platforms and its digital assets, the adoption of a security-focused perspective during the development and deployment of the platform's services and the design of the OneNet Monitoring and Analytics Dashboard, which enables the extraction of valuable observations from collected data.

1 Introduction

The European electricity system is today often managed in a fragmented way, on a country- or area-level basis. OneNet will develop an open and flexible architecture to transform the European electricity system to be pan-European, smarter and more efficient, while maximizing the consumer's capabilities to participate in an open market structure. WP6 is responsible for the development of the OneNet system, starting from the design of the OneNet Reference Architecture described in D5.2 [2] and all the necessary requirements and specification provided in the other WP5 tasks. In addition, WP6 also performs monitoring and evaluation activities during the integration and execution phase, specifically focused on the compliance with the OneNet Reference Architecture and the Cybersecurity aspects.

1.1 Task 6.6

This task ensures the integration of legal, regulatory and cybersecurity principles in the project's design and development. It involves monitoring of historical and real-time data of the network and endpoint infrastructure levels. Additionally, it aims to establish robust anomaly detection methods by leveraging network traffic analysis and automated rating and classification techniques based on machine learning.

1.2 Objectives of the Work Reported in this Deliverable

This report presents the work conducted throughout the development activities of WP6 to comply with legal, regulatory, privacy and cybersecurity guidelines established in the NISTIR 7628 report [1] and describes the implementation of the tools offered to the OneNet administrator and users for network monitoring and analytics.

The report is divided into 2 parts. The first part reports on the cybersecurity measures first analyzed in D5.8 [3] and in what manner and to what extent they were implemented by the various tools and components comprising the OneNet Interoperable Network of Platforms, such as the Middleware, the Connector, the Orchestration Workbench and the Monitoring and Analytics Dashboard.

The second part is dedicated to the implementation of the OneNet Monitoring and Analytics Dashboard which facilitates network traffic monitoring and abnormal client behavior for the OneNet administrator as well as OneNet users. It describes the design and implementation of v1.0 of the dashboard components. It first details the Connector log collection procedure, how the logs are filtered, stored and indexed and describes the data analysis, rating and classification tool used to determine whether a client displays normal or abnormal behavior. Next, it defines the authentication and authorization process and the identity manager selected to perform it and finally, it describes the dashboard GUI functionality.

WPs Interactions

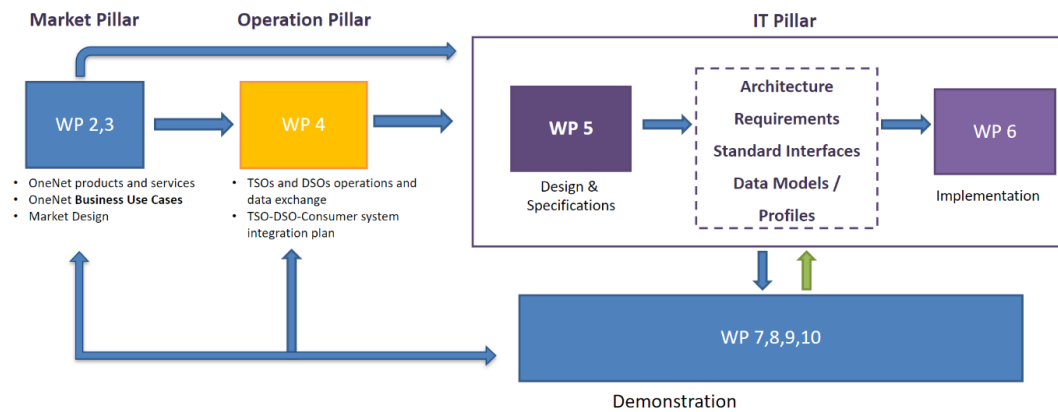


Figure 1: WPs interdependencies

1.3 Report Outline

This report has the following structure:

Chapter 2 describes the manner and the extent to which legal, regulatory, privacy and cybersecurity requirements have been fulfilled by the tools and components which constitute the OneNet Interoperable Network of Platforms, based on the guidelines provided in the NISTIR 7628 document [1].

Chapter 3 presents the technical characteristics of the OneNet Monitoring and Analytics Dashboard, analyzing the implementation of its first version (v1) and providing details on each of its components.

Chapter 4 concludes the report.

1.4 How to read this document

The reader is expected to have read the following deliverables as background to this document:

- D5.8 [3], particularly Chapter 7, which provides an overview of the OneNet Architecture and enumerates the identified cybersecurity requirements and constraints of the OneNet Interoperable Network of Platforms.
- D6.4 [4], particularly Chapter 7, which analyzes the architecture of the OneNet Monitoring and Analytics Dashboard, the services it is composed of and the process of packaging and deployment of this component.

2 Requirements for Legal, Regulatory, Privacy and Cybersecurity Compliance

2.1 NISTIR 7628 Requirements

As mentioned in D5.8 [3], in terms of cybersecurity, privacy and regulatory aspects of the OneNet architecture, the NISTIR 7628 Smart Grid Cyber Security standard [1] and the SGIS Report [5] are considered to be the most relevant. It is noted that both of the aforementioned reports are developed by the National Institute of Standards and Technology (NIST), a reputable authority in the field of cybersecurity. Due to the nature of the project, we have opted to focus on NISTIR 7628 which is specifically tailored to smart grid system security. Therefore, we have dedicated this section to the NISTIR 7628 standard and how we address its guidelines in each of the components which constitute the OneNet Interoperable Network of Platforms.

Table 1: NISTIR 7628 requirements, related GDPR privacy principles and recommendations tailored to the OneNet Interoperable Network of Platforms

NISTIR 7628 requirements	Description & Recommendations	CIA	GDPR privacy principles
SG.AC Access Control	<p>Ensure resources are only accessed by authorized personnel.</p> <p><u>Recommendations:</u></p> <ul style="list-style-type: none"> • Separation of duties should be enforced to eliminate conflicts of interests. (NISTIR 7628 SG.AC-6) • Principle of least privilege should be implemented. (NISTIR 7628 SG.AC-7) • For critical systems with higher security levels consider using of multi-factor authentication, cryptographic devices, or client-side certificates for higher impersonation resistance. (OWASP ASVS 2.2.4) <p><u>Notes:</u></p> <p>According to survey results, most reported security breaches were spam and phishing emails, further reinforcing the need for separating less critical office systems from mission critical ones.</p>	C, I, A	<ul style="list-style-type: none"> • Lawfulness, fairness and transparency • Purpose limitation • Data minimisation • Integrity and confidentiality (security) • Accountability

SG.AC Access Control	<p>Ensure resources are only accessed by authorized personnel.</p> <p><u>Recommendations:</u></p> <ul style="list-style-type: none"> • Separation of duties should be enforced to eliminate conflicts of interests. (NISTIR 7628 SG.AC-6) • Principle of least privilege should be implemented. (NISTIR 7628 SG.AC-7) • For critical systems with higher security levels consider using of multi-factor authentication, cryptographic devices, or client-side certificates for higher impersonation resistance. (OWASP ASVS 2.2.4) <p><u>Notes:</u></p> <p>According to survey results, most reported security breaches were spam and phishing emails, further reinforcing the need for separating less critical office systems from mission critical ones.</p>	C, I, A	<ul style="list-style-type: none"> • Lawfulness, fairness and transparency • Purpose limitation • Data minimisation • Integrity and confidentiality (security) • Accountability
SG.AU Audit and accountability	<p>Security of OneNet information system should be validated by conducting periodic audits and logging of critical activities.</p> <p><u>Recommendations:</u></p> <ul style="list-style-type: none"> • Detect and record security relevant events. (OWASP ASVS 7.1.3) • Non-repudiation measures should be implemented. (NISTIR 7628 SG.AU-16) <p><u>Notes:</u></p> <p>According to survey results insider attacks were considered one of the most critical threats for Energy industry.</p>	I	<ul style="list-style-type: none"> • Lawfulness, fairness and transparency • Integrity and confidentiality (security) • Accountability

SG.CA Security assessment and authorization	<p>Compliance of OneNet information system should be regularly assessed. In case of nonconformance appropriate corrective actions should be implemented.</p> <p><u>Recommendations:</u></p> <ul style="list-style-type: none"> Conduct routine self-assessments. (ENISA Smart Grid Threat Landscape and Good Practice Guide 9.1.9) 	C, I	Integrity and confidentiality (security)
SG.CM Configuration management	<p>Policies and procedures must be set in place to manage and document all configuration changes to information system. All updates and patches should be thoroughly tested on a non-production environment.</p> <p><u>Recommendations:</u></p> <ul style="list-style-type: none"> System components should be configured to provide only essential functionality with unnecessary functions, ports, protocols and services disabled. (NISTIR 7628 SG.CM-7) Baseline configuration for smart grid information system should be developed, documented and maintained as well as keeping previous baselines for possible rollback (NISTIR 7628 SG.CM-2) <p><u>Notes:</u></p> <p>Establishment of configuration management process is recommended. (ENISA Smart Grid Threat Landscape and Good Practice Guide 9.1.3)</p>	I, A	<ul style="list-style-type: none"> Integrity and confidentiality (security) Accountability
SG.CP Continuity of operations	<p>Capacity to continue or resume operations after disruptions should be documented. Security measures necessary for maintaining required continuity level must be guaranteed.</p> <p><u>Recommendations:</u></p> <ul style="list-style-type: none"> OneNet systems should integrate fail-safe response procedures upon the loss of communications with other systems. (NISTIR 7628 SG.CP-11) Use of backup telecommunication provider(s) (NISTIR 7628 SG.CP-8) and 	A	N/A

	<p>alternate control center(s) should be considered. (NISTIR 7628 SG.CP-9)</p> <p><u>Notes:</u></p> <ul style="list-style-type: none"> • Distributed/decentralized architecture would increase availability and reliability of OneNet information system (Based on the experience from UXP [33]). • Load balancing should be used for critical components to guarantee continuous functioning of the infrastructure (Based on experience from UXP). • It should be possible to increase the reliability and performance of all components by adding redundancy (Based on experience from UXP). 		
<p>SG.IA Identification and authentication</p>	<p>Identity of users must be verified before granting them access to OneNet information system.</p> <p><u>Recommendations:</u></p> <ul style="list-style-type: none"> • Authentication mechanism should obscure feedback during authentication process. (NISTIR 7628 SG.IA-6) <p>Anti-automation measures should be implemented to mitigate breached credential testing, brute force and account lockout attacks. (OWAPS ASVS 2.2.1)</p>	<p>C, I</p>	<p>Accountability</p>
<p>SG.ID Information and document management</p>	<p>Important and sensitive information and documentation must be protected and retained.</p> <p><u>Recommendations:</u></p> <p>Communications with devices outside OneNet system should be limited only to the devices that need to communicate. (NISTIR 7628 SG.ID-4).</p>	<p>C, I, A</p>	<ul style="list-style-type: none"> • Lawfulness, fairness, and transparency • Integrity and confidentiality (security) • Accountability

SG.IR Incident response	<p>Capability to maintain or resume operations of information system in the event of disruption must be maintained.</p> <p><u>Recommendations:</u></p> <ul style="list-style-type: none"> In case of wider adaptation of technologies developed during OneNet need for European Organization similar to US ICS-CERT has been identified. (SGIS Report) <p><u>Notes:</u></p> <p>Based on survey results over 50% of OneNet stakeholders require intrusion detection for power systems from second to minutes timescale.</p>	C, I, A	Accountability
SG.MA Smart grid information system development and maintenance	<p>Security measures should be sustained and improved through effective maintenance of OneNet information system.</p> <p><u>Recommendations:</u></p> <ul style="list-style-type: none"> Administration and management functions should be limited to authorized administrators. (OWASP ASVS 13.1.2). <p>Authorized administrators should be able to verify integrity of all security relevant configurations. (OWASP ASVS 14.1.5).</p>	C, I, A	Integrity and confidentiality (security)
SG.MP Media protection	<p>Access to physical media should be limited only to authorized users.</p> <p><u>Recommendations:</u></p> <ul style="list-style-type: none"> Passwords, integrations with databases and third-party systems, API keys should resist offline attacks. (OWASP ASVS 2.10.4) <p>Regulated private data should be stored encrypted. (OWASP ASVS 6.1.1)</p>	C, I	<ul style="list-style-type: none"> Lawfulness, fairness and transparency Purpose limitation Data minimisation Accuracy Integrity and confidentiality (security) Accountability
SG.PE Physical and environmental security	<p>Physical access control and surveillance mechanisms should be implemented to ensure only authorized access to system components.</p>	C, I, A	<ul style="list-style-type: none"> Integrity and confidentiality (security) Accountability

SG.PL Planning	Security planning should be utilized to prevent undesirable interruptions to continuity of operations.	C, I, A	N/A
SG.PM Security program management	<p>Security program management should be utilized throughout life cycle of information system in order to guarantee adequate security policy.</p> <p><u>Recommendations:</u></p> <ul style="list-style-type: none"> • Senior management authority should be appointed to coordinate, develop, implement and maintain security program. (NISTIR 7628 SG.PM-3) <p>Framework of management accountability should be defined so that it establishes roles and responsibilities related to cybersecurity across the organization. (NISTIR 7628 SG.PM-8).</p>	C, I	<ul style="list-style-type: none"> • Integrity and confidentiality (security) • Accountability
SG.PS Personnel security	Procedures for background checks, employee and contractor onboarding and offboarding should be documented.	C, I, A	<ul style="list-style-type: none"> • Lawfulness, fairness and transparency • Integrity and confidentiality (security) • Accountability
SG.RA Risk management and assessment	Risk identification and classification process should be continually performed to ensure information system's compliance to necessary requirements.	C, I, A	Integrity and confidentiality (security)

SG.SA Smart grid information system and services acquisition	<p>Detailed procedures for reviewing acquisitions of new system components should be enforced in order to avoid introduction of additional vulnerabilities into the OneNet information system.</p> <p><u>Recommendations:</u></p> <ul style="list-style-type: none"> Security engineering principles should be applied in specification, design, development and implementation of all OneNet information systems. (NISTIR 7628 SG.SA-8) Information system documentation should include guides on how to install, configure and use security features built into the system. (NISTIR 7628 SG.SA-5) <p>System development lifecycle methodology should include security. (NISTIR 7628 SG.SA-3)</p>	C, I, A	Integrity and confidentiality (security)
SG.SC Smart grid information system and communication protection	<p>Measures should be considered to protect information system components and communication links against cyber intrusions.</p> <p><u>Recommendations:</u></p> <ul style="list-style-type: none"> Industry proven or government approved cryptographic algorithms and libraries should be used. (OWASP ASVS 6.2.2). <p>Recommendations on cryptographic algorithms and key sizes should be updated frequently. (OWASP ASVS 6.2.3).</p>	C, I	Integrity and confidentiality (security)
SG.SI Smart grid information system and information integrity	<p>Integrity of sensitive data should be maintained.</p> <p><u>Recommendations:</u></p> <ul style="list-style-type: none"> Security functions should be verified on system start-up, restart and at defined frequency when tasked by user with appropriate privileges. (NISTIR 7628 SG.SI-6) <p>Announced software and firmware flaws as well as flaws discovered during security assessments need to be addressed. (NISTIR 7628 SG.SI-2).</p>	I	Integrity and confidentiality (security)

Different partners have been working on different components of the OneNet Interoperable Network of Platforms. Therefore, each of the following subsections refers to the cybersecurity measures taken by the respective partner during the development of the respective component.

2.1.1 OneNet Monitoring and Analytics Dashboard

Regarding access control, access to virtual machines hosting the OneNet Monitoring and Analytics Dashboard services shall be solely provided to the members of the organization who work on the OneNet project. Furthermore, the virtual machines are to be remotely accessed through SSH using a specific SSH key given to the team members and they must be unable to be accessed outside of the organization's network. Password authentication shall be disabled and PAM authentication shall be enabled in the SSH server configuration. Accounts allowed to connect to the corporate VPN must be managed by the organization administrator. Regarding access to the information system itself, the OneNet Monitoring and Analytics Dashboard should employ an identity server such as Keycloak for the purpose of managing authentication and authorization to the smart grid information system. Through an Admin Console, the administrator should have the ability to manually manage accounts and roles, review activity and configure settings related to authentication and authorization. Two-factor authentication should also be supported as a method of authentication. Users should be able to enable it through their Account Management page and authenticate themselves utilize a smartphone application which generates verification codes.

Regarding security information and event management, all traffic originating from OneNet Connectors' user interface shall be logged and the resulting logs shall be centralized in a middleware Virtual Machine using the log collection process as described in D6.4 [4] and this report. Logs should include fields such as timestamp, country, IP address, bytes sent, response code and more. The logs shall be processed, analyzed and used as input to machine learning algorithms in order to detect anomalous behavior. The monitoring and analytics dashboard may arrange this information in a set of graphs displaying both real-time as well as historic data, which may be monitored by the dashboard users at any time. Through the dashboard it should also be possible to generate a security report which displays additional information about clients classified as abnormal. Accountability and non-repudiation shall be achieved through the use of the selected identity manager.

Furthermore, through Git, our version control system of choice, the codebase should be organized in develop and master git branches. The changes from the develop branch shall be merged into the master branch after thorough testing. Environment variables used for service configuration and stored in files must be excluded from version control to ensure that sensitive information such as API keys are not exposed. The deployment of the services shall be orchestrated using docker and docker-compose as described in D6.4 [4]. We should also maintain audited and trusted docker images for common tools and components which are available for download from the container registry.

Virtual machine resources shall be allocated upon request to the IT department through a ticketing system.

For the OneNet Monitoring and Analytics Dashboard we have adopted a container-based service deployment scheme using Docker. For a number of RAM-intensive services we shall perform manual configuration of the container maximum RAM usage. The configuration of other services should be performed using configuration files or environment variables passed to the respective container via docker-compose.

Regarding collaborative computing, the development team shall utilize the functionality offered by Google Workspace, which includes web applications for email, calendar, video conference, document editing and cloud storage. When used through a modern web browser, the user is asked for permission before their microphone and camera can be activated and used by the respective web application, which fulfills our requirements for explicit indication of use.

The use of personal cloud resources such as Google Drive and OneDrive for the storage of corporate information regardless of nature and classification must be strictly prohibited. In addition, the use of organization passwords such as the personal LDAP password and server passwords in personal services (personal email, social media and web banking) should be highly discouraged.

The corresponding personnel assigned by the administrator shall be responsible for managing the visitors they have personally invited. Visitors are divided into two categories, temporary (e.g., deliveries, couriers) and permanent (e.g., business meetings). In both cases, the visitors should be picked up at the entrance after informing the Secretariat. Unsupervised movement and activity of the visitors inside the premises shall be prohibited. Both temporary and permanent visitors ought to be accompanied when exiting the building. In the case of permanent visitors requiring the use of computing resources, a specific electronic invitation and registration procedure must be followed.

Developers shall be provided with a personal entry card (RFID), a username and password, and an alarm code. The entry card shall be used to control physical access to the site, the username and password to access all online services and the alarm code to activate and deactivate the alarm in the facilities. Upon logging in for the first time, the password must be changed.

The password policy dictates that passwords must necessarily consist of a minimum of 9 characters with at least one capital and one lowercase character and a special symbol. Passwords shall expire every 90 days. After their expiration, for reasons of business continuity, access to the services should continue. However, the Security Officer should be updated daily and the respective accounts can be deactivated at any time. One week before the password expiration date, an automated notification should be sent as a reminder.

This password must be strictly personal and must not be disclosed under any circumstances to anyone inside and outside the organization. The use of the password in third-party services shall be prohibited as it jeopardizes organization resources. The security department should conduct periodic audits in order to determine if password hashes exist in public lists of hashed passwords. In case of loss of the password, immediate notification of the Security Officer is necessary. The same procedure must be followed in the case of loss of the entry card.

Documents containing confidential information related to the project shall remain locked in the employee's personal locker in their absence. In the case of document loss or suspicion of loss, the Security Officer must be informed. Documents shall be destroyed following a shredding process.

Users must be responsible for the management of their personal computer and shall be required to fulfill a number of security obligations. They should be encouraged to enable screen locking when leaving their workstation and enable automatic screen locking after 10 minutes of inactivity. Furthermore, they should be required to install and periodically update antivirus software and enable encryption in their home folder. Finally, browsers should be configured not to store credentials. The security department may make periodic checks to determine users' compliance with the aforementioned guidelines.

Usage of USB sticks that have not been given by the company shall not be allowed. It should also be prohibited to run files received via email (such as files with the extensions *.exe, *.sh, *.iso). Finally, it shall be forbidden to run macros in Microsoft Office files that have been sent as attachments. In the event that an attached file is executed, the Security Officer should be contacted. Users shall be encouraged to be especially alert to suspicious emails that demonstrate social engineering.

Sending files outside the organization shall be performed using the service organization Drive, in conjunction with encryption and time-limited sharing only. In the case when the use of external services (e.g. GitHub) is necessary, the use of hardened passwords, according to the password policy mentioned above, should be highly encouraged.

All services that are accessible outside the organization such as email, drive and chat are TLS enabled. However, it is recommended to always connect to the corporate VPN for protection from a range of attacks.

In the case of a detected security incident, it is vital to ensure that the Security Officer is immediately informed and not proceed with any corrective actions that may destroy audit trails. Forensic analysis and the management of such an incident shall be initiated by the Security Officer according to a prescribed procedure.

In case of failure of a computing resource that has been made available to an employee, such as a laptop or USB Disk, the resource must be delivered to the Security Officer in order to execute the destruction protocol.

Finally, prior to integrating any data processing activities and software components, a comprehensive Data Protection Impact Assessment (DPIA) has been carried out, as described in D6.5 [6]. This assessment adheres to the existing legal framework in the European Union, specifically the General Data Protection Regulation (GDPR), as well as the relevant guidelines outlined for the OneNet project which align with the common requirements specified on page 10 of the "Horizon 2020 - Work Programme 2018-2020 Secure, clean and efficient energy" document [7].

2.1.2 OneNet Decentralized Middleware and OneNet Connector

The OneNet Decentralized Middleware is the key component of the OneNet Interoperable Network of Platforms since it enables the process of managing and sharing information in a controlled and administrative environment, dealing with the user management, central meta-data brokerage and the logging of all occurring transactions for auditing reasons. This framework does not have access to the data shared by the OneNet participants, nor does it forward or process such data. In this way, the owners of the data can have total control of the sovereignty of their data. The OneNet Decentralized Middleware orchestrates processes such as the identity management, data catalogue (including the administration of OneNet Cross-Platform Services list), data quality process, logging processes and data access policies (including the Connector's discoverability based on meta-data information).

The OneNet Connector is a deployment instance of the OneNet Decentralized Middleware and, once deployed and integrated within the platforms of each OneNet participant, it allows a trusted pan-European data space for the electricity sector focusing on flexibility procurement and exploitation. The development of the OneNet Connector considered all functional and non-functional requirements collected starting from the different use cases of OneNet.

As per the NISTIR 7628 requirements:

- **SG.AC Access Control:** The OneNet Connector adheres to a pre-specified access control schema where access is defined by user registration through Identity Management & specific access rules through the dedicated GUI. As such we ensure that resources are only accessed by authorized personnel.
- **SG.AU Audit and accountability:** For every data exchange which occurs, the appropriate audit and logging information is registered in the Clearing House according to IDS Reference Architecture.
- **SG.CA Security assessment and authorization:** Providing access to the OneNet ecosystem, one needs to deploy locally the OneNet Connector, assuming for the proper SSL certificates, which are recorded in the OneNet Middleware. Upon this, users are assigned for each Connector and institution. This process is continuous until the development finalization.
- **SG.CM Configuration management:** Already established as a process in the GitHub environment from the 1st to final middleware version. All configuration changes are visible in GitHub (versioning) and all updates and patches are thoroughly tested before becoming part of the production environment. The deployed Connectors at demo partners are informed for updated versions via their UI interfaces with the central Middleware.
- **SG.CP Continuity of operations:** The specific requirement has been addressed through the P2P decentralized approach of the whole data exchange process. The nature of decentralized execution of data exchanges along with the central transaction logging at OneNet Middleware addresses this requirement.
- **SG.IA Identification and authentication:** Every user accessing the system is uniquely identified and verified (Keycloak IDM).

- **SG.ID Information and document management:** Every outside participant is identified and can access the system by the same access control schema. No specific SG.ID Information and document management process is in place though.
- **SG.IR Incident response:** Non-applicable as it is covered by the OneNet Dashboard.
- **SG.MA Smart grid information system development and maintenance:** A complete maintenance plan will be developed as part of the final Interoperable Network of Platforms version.
- **SG.MP Media protection:** Data exchanges are explicitly among Data Producers and Providers. No other stakeholders can invoke or intervene these operations. Only specified administrators can have access to physical media.
- **SG.PE Physical and environmental security:** Non-applicable as and physical and environmental access are not related with the middleware functionality but are rather part of the deployment process (and the stakeholder security processes for local deployments)
- **SG.PL Planning:** See **SG.CP Continuity of operations.**
- **SG.PM Security program management:** Part of the project cybersecurity implementation and management aspect.
- **SG.PS Personnel security:** As defined in SG.PM Security program management methodology.
- **SG.RA Risk management and assessment:** As defined in PM Risk Management process.
- **SG.SA Smart grid information system and services acquisition:** See **SG.CA Security assessment and authorization;** all new users can openly access, deploy and test the OneNet connector. Entering the OneNet data ecosystem implies the registration to the OneNet Middleware.
- **SG.SC Smart grid information system and communication protection:** See **SG.PM Security program management.**
- **SG.SI Smart grid information system and information integrity:** The whole OneNet middleware functionality is based on the security, trust, integrity & sovereignty pillars. As such integrity of sensitive data is a basic consideration. Assurance that sensitive data have neither been modified nor deleted in an unauthorized or undetected manner is achieved with the process of the service provision and registration with the subsequent contract between producer and consumer.

2.1.3 OneNet Orchestration Workbench

The OneNet Orchestration Workbench offers the possibility to integrate data sources through the OneNet Decentralised Middleware and OneNet Connector and to orchestrate and evaluate the performance and scalability of the OneNet cross-platform services.

Below is a list of NIST requirements taken into account during the design, implementation and deployment of the Orchestration Workbench. Smart Grid related requirements are not considered since the Orchestration Workbench does not interact with the smart grid environment.

The OneNet Workbench is a web-based platform hosted in ENG cloud environment. Access to the VM where the Orchestration Workbench is, is limited to ENG accounts and under VPN connection.

The platform is accessible through browser only for OneNet Participants and the access control is completely integrated with the OneNet Identity Management, ensuring a centralized access management in the overall system.

All the internal activities of the OneNet Orchestration Workbench are logged within the overall system. The data exchange and the usage of the OneNet Connector (integrated with the Workbench) are logged by the log collection process implemented in OneNet.

In order to ensure only authorized access and performed security assessment, the overall system of the OneNet Orchestration Workbench is monitored by cybersecurity specialists who monitor the activity and possible issues related to the system, intervening in advance to solve possible safety problems.

It's possible to manage the different configuration of the OneNet Orchestration Workbench development environment through a versioning system and source code repository hosted in private ENG cloud. The updates on the system are managed using the development branch for testing and main branch for releasing. In addition, any software updates are tested in a development environment and then released in the production one.

The continuity of operations is ensured by providing a continuous monitoring of the infrastructure and a dedicated ENG team is in charge to ensure the continuity of the systems. A daily backup is implemented and a recovery plan is automatized.

Identification and authentication mechanisms are centralized in the OneNet Connector and Decentralized Middleware, ensuring a unique identification and authorization mechanism for the whole OneNet system. Username and password (with strong password) are used for the UI access and OAuth Token are used for REST APIs authentication.

No personal information is managed by the Orchestration Workbench. All the managed data is stored in specific database of the OneNet Orchestration Workbench (if requested by the user) and never shared outside the Workbench itself.

There is a specific team in charge of managing any incident. The ENG cloud environment is completely isolated and can be stopped in case of incident without compromising other environments. Remote monitoring and maintenance tools are provided to the security teams for monitoring incident status.

All the media devices are registered within the ENG cloud server and no removable media devices are admitted. A daily backup all the media storage is automatically enabled.

In addition, the ENG cloud infrastructure provides a physical and environmental security, ensuring that only internal authorized personnel can physically access the ENG cloud systems.

3 The OneNet Monitoring and Analytics Dashboard – Technical Characteristics

3.1 OneNet Monitoring and Analytics Dashboard Architecture

This chapter focuses on the tools used to implement the OneNet Monitoring and Analytics Dashboard and presents a comprehensive description of its implementation and the functionality it offers. The architecture of the Dashboard has been described in detail in D6.4 [4]. Figure 2 provides an overview of this architecture, the services comprising the dashboard and their interconnection.

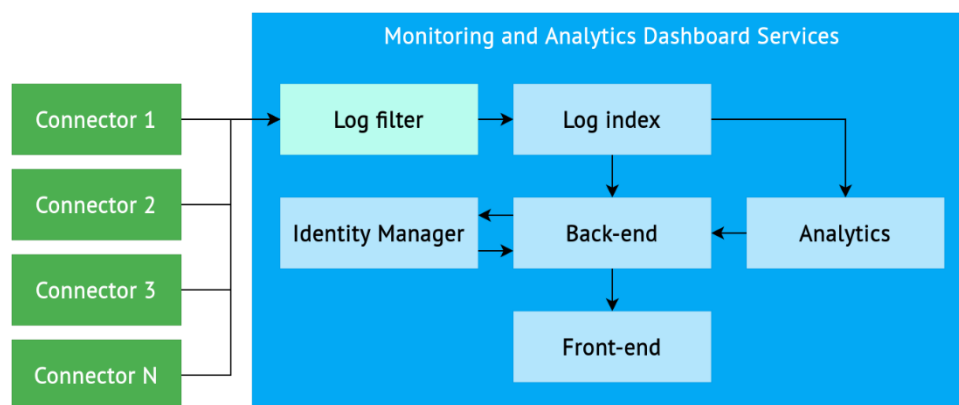


Figure 2: OneNet Monitoring and Analytics Dashboard service architecture

3.2 OneNet Monitoring and Analytics Dashboard – v1.0

The current document describes the first version of the OneNet Monitoring and Analytics Dashboard. The final version along with a detailed description of its new features and functionality will be reported in D6.8.

3.2.1 OneNet Network Traffic & Endpoint Infrastructure Monitoring Tool

The goal of the Network traffic monitoring tools is to achieve security through analytics. Our implementation was inspired by [Baskerville](https://github.com/deflect-ca/baskerville)¹, a security analytics engine for anomaly detection in web traffic, which collects access logs from clients and leverages machine learning to analyze them and classify them as normal or abnormal.

The primary objective is to collect logs from all the Connectors, which are set up in a decentralized manner and forward them to a centralized service which will proceed with the log analysis. This is achieved using the

¹ <https://github.com/deflect-ca/baskerville>

ELK stack along with Filebeat². ELK is a software stack composed of Elasticsearch³, Logstash⁴ and Kibana⁵. Within the context of the OneNet Monitoring and Analytics Dashboard, Elasticsearch is used to store and index Connector logs, Filebeat and Logstash are used to collect, transform and forward logs to Elasticsearch and Kibana is used for its visualization functionality.

In particular, we set up a Filebeat and a Logstash instance in each Connector. In the dashboard VM, we set up a Logstash instance and Elasticsearch. Whenever a client makes an action in the Connector web UI which triggers a network request, a log is collected through the Connector Logstash (sending Logstash) and forwarded to the Logstash instance running in the dashboard VM (receiving Logstash), where it is stored in Elasticsearch. From there, the backend and analytics services are able to read the collected logs and analyze them using statistical analysis and machine learning.

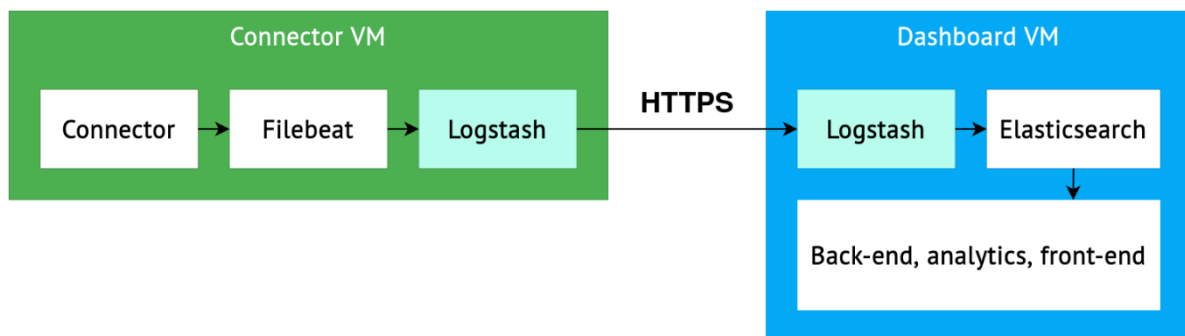


Figure 3: Log collection architecture

Communication between the Connector and the dashboard happens using HTTPS and Logstash-to-Logstash communication, as seen in the diagram below.

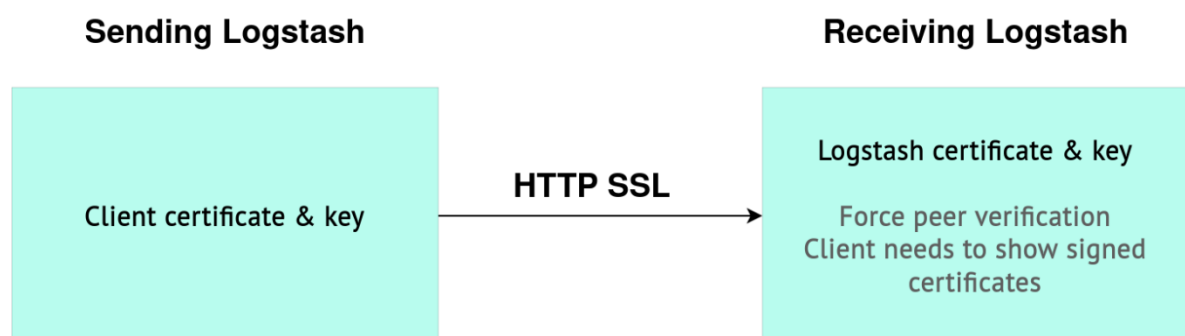


Figure 4: Logstash to Logstash direct HTTPS communication

Essentially, Filebeat gathers logs from the Connector's nginx server and passes them to Logstash. Using the Logstash HTTP output plugin, the logs are sent to the receiving Logstash instance running in the middleware,

² <https://www.elastic.co/beats/filebeat>

³ <https://www.elastic.co/>

⁴ <https://www.elastic.co/logstash>

⁵ <https://www.elastic.co/kibana>

which uses the HTTP input plugin to accept logs. The receiving Logstash is accessible by all Connectors through the following domain:

`https://onenet-logstash.euprojects.net`

Connector (sending) Logstash pipeline output configuration:

```
output {
  http {
    url => "https://onenet-logstash.euprojects.net"
    http_method => post
    cacert => "ca.crt"
    client_key => "client.key.pk8"
    client_cert => "client.crt"
  }
}
```

Middleware (receiving) Logstash pipeline input configuration:

```
input {
  http {
    port => 5044
    ssl => true
    ssl_key => "server.key.pk8"
    ssl_certificate => "server.crt"
    ssl_certificate_authorities => "ca.crt"
    ssl_verify_mode => force_peer
  }
}
```

In order to sustain larger traffic, it would be possible to utilize a message broker such as Kafka or RabbitMQ. This would enable the use of automatic horizontal scaling and load balancing between the sending and receiving Logstash instances.

Every log contains agent fields which contain the data about the software entity that collects events on a host. In our case, the software entities would be the Filebeat and Logstash instances running in the Connector. We may use the agent fields to determine which Connector the event originated from. For instance:

- `agent.id` is a unique identifier of the agent which collected the log.
- `agent.name` is a **custom** identifier which may be given to an agent upon Connector deployment using environment variables.

The Logstash-to-Logstash communication is secured using certificates. The certificates have been generated and both the sending and receiving Logstash instances have been configured to use them, as seen in the Logstash configuration.

It is noted that Logstash-to-Logstash HTTP communication allows using any HTTP proxy. We have also created and tested a Logstash configuration which would function correctly in the case of deployment in an infrastructure that involves a HTTP proxy, where all communication between the sending and receiving Logstash instances goes through the proxy. In this case, basic username/password authentication is used instead of certificates.

The alternative Logstash pipeline configurations are set as follows:

Sending Logstash

```
output {
  url => "https://onenet-logstash.euprojects.net"
  http_method => post
  headers => {
    "Authorization" => "Basic XXXXXXXXXXXXXXXXXXXX=="
  }
}
```

where XXXXXXXXXXXXXXXXXXXX would be the Base64-encoded version of the string my_username:my_password.

Receiving Logstash

```
input {
  http {
    port => 5044
    user => my_username
    password => my_password
  }
}
```

A secure connection is created between the sending Logstash and the proxy, where the data is then decrypted and routed to the receiving Logstash instance, as seen in the diagram below.



Figure 5: Logstash to Logstash communication through a HTTP proxy

The input of the Network Monitoring and Analytics Dashboard begins at the Logstash instance running in our virtual machine, which is responsible for filtering incoming Connector logs. The logs are generated each time a client makes an action in the Connector web UI which triggers a network request. It is vital to note that the logs

are not present in the file system of our containers but are continuously forwarded from the Connector to our Logstash instance. The Logstash pipeline has been configured to use HTTP input to accept logs from the Connector and write logs to our Elasticsearch instance using the index format `connectors-%{+YYYY.MM.dd}`.

This is an example of an original message received by Logstash:

```
192.168.3.1 - - [20/Apr/2023:10:07:28 +0000] "GET /api/settings/login-image
HTTP/1.1" 200 10434 "https://onenet-ubi-connector.eupprojects.net/login"
"Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101
Firefox/112.0" "213.249.38.66, 172.68.62.80"
```

Using Logstash filters, it is possible to extract additional information from this message and its headers:

```
filter {
  grok {
    match => { "message" => "%{COMBINEDAPACHELOG}"
  }
  mutate {
    convert => ["bytes", "integer"]
  }
  geoip {
    source => "client_ip"
    target => "client_geoip"
  }
  useragent {
    source => "message"
    target => "user_agent"
  }
}
```

Additionally, we use the filter to manually extract the IP of the client who has triggered the request into its own field (`client_ip`). Using this field and the `geoip` Logstash filter plugin we are able to add information about the geographical location of the client to the logs. The plugin is bundled with the free [GeoLite2](#) City database and uses it by default; however it is possible to configure it to use any database.

From there, in Elasticsearch this information is organized and displayed as a JSON object. The snippet below is an example of how this information is displayed inside the Elasticsearch logs (some of the fields have been hidden for brevity):

```
{
  "_index" : "connectors-2023.04.28",
  "_source" : {
    "request" : "/api/dashboard/by-id?id=a8b14b31-b2b3-4f1a-9060-
ea6056e808e3",
```

```

"client_geoip" : {
  "ip" : "yyy.yyy.yyy.yyy",
  "country_name" : "Greece",
  "location" : {
    "lat" : 37.9842,
    "lon" : 23.7353
  },
  "timezone" : "Europe/Athens",
  "region_code" : "I",
  "continent_code" : "EU",
  "region_name" : "Attica",
  "country_code2" : "GR",
  "city_name" : "Athens"
},
"agent" : {
  "version" : "8.3.2",
  "name" : "connector_name",
  "ephemeral_id" : "856c3f5b-02d1-42bc-bb08-e4723f6fec3e",
  "id" : "c950c2d7-624f-429b-873f-905a5271d045",
  "type" : "filebeat"
},
"client_ip" : "yyy.yyy.yyy.yyy",
"timestamp" : "28/Apr/2023:05:48:26 +0000",
"host" : "192.168.3.1",
"bytes" : 606,
"headers" : {
  "content_length" : "1332",
  "request_method" : "POST",
  "x_forwarded_for" : "192.168.3.135",
  "accept_encoding" : "gzip,deflate",
  "content_type" : "application/json",
  "http_host" : "onenet-logstash.euprojects.net",
},
"response" : "200",
"verb" : "GET"
}
}

```

Fields of interest include:

- `headers.x_forwarded_for`: The Connector IP address from which the event originated.
- `agent.name`: A Connector identifier.
- `bytes`: The bytes sent from the Connector to the client in the response.
- `response`: The response code.

Kibana offers a native dashboard connected with the Elasticsearch instance. The Kibana dashboard provides an intuitive interface for testing queries and aggregations, and serves as a testing ground for various features of

the Elasticsearch API. A Kibana container has been configured for testing and debugging purposes through the Dev Console, as well as visualizations of collected logs using the native Elasticsearch aggregation methods.

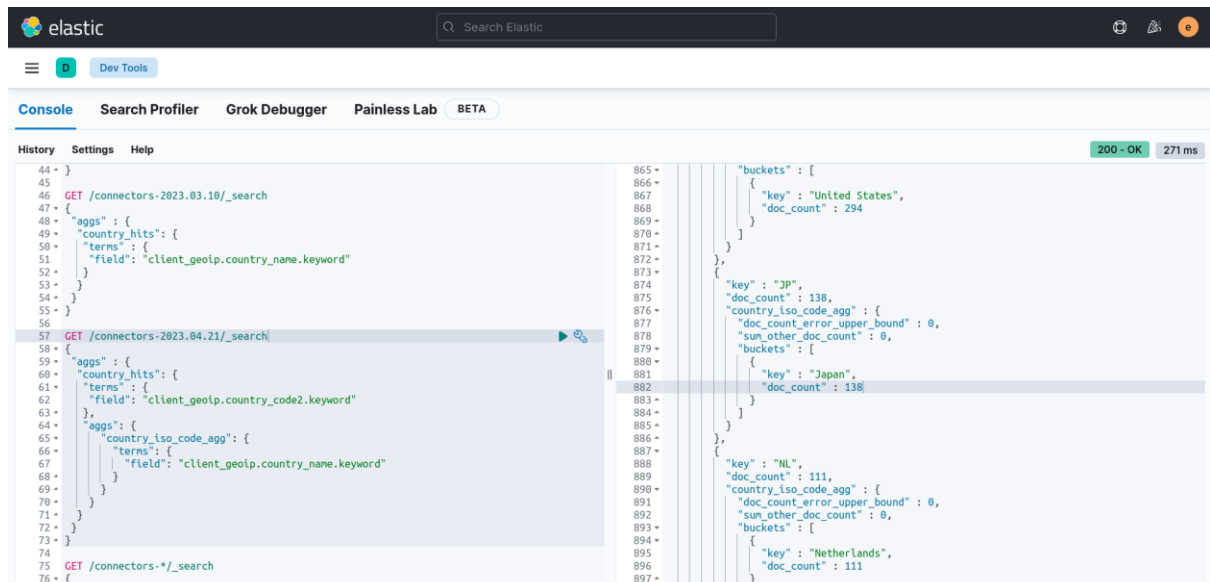


Figure 6: Example of sending requests through the Elastic Dev Console in Kibana

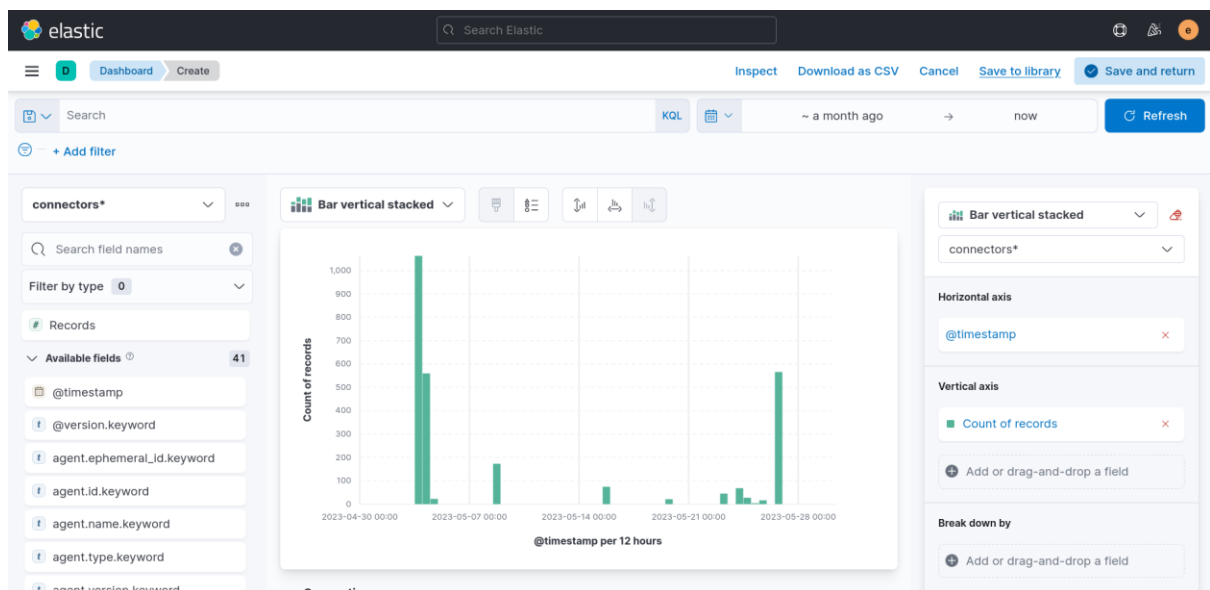


Figure 7: Visualization of number of records over one month in Kibana

Our back-end, developed with Spring Boot, is responsible for the integration with the Keycloak Identity Manager for authentication and authorization purposes. In addition, it queries Elasticsearch and the analytics service in order to create and secure API endpoints to be used by the dashboard front-end.

Endpoints

Authentication:

The Spring Boot back-end integrates with Keycloak for authentication purposes. Using the administrator credentials and the `admin-cli` Keycloak client, an instance of the Keycloak class is created. This instance is used to perform all the authentication operations.

- **POST /login**
This endpoint accepts a username and password which are forwarded to Keycloak. Keycloak returns a JWT which is then returned in the response body.
- **POST /logout**
This endpoint accepts a refresh token and uses the logout URI of Keycloak to logout the user. It does not return any data in its response.
- **POST /refresh**
This endpoint accepts a refresh token which is sent to Keycloak. Keycloak in turn returns a new token which is then returned in the response body.
- **POST /register**
This endpoint accepts a username, password, password confirmation and email which are forwarded to Keycloak. The username and email are forwarded to Keycloak which creates the user in the `main-authentication` Keycloak realm. Next, user roles are assigned and the user password is set. It does not return any data in its response.

Monitoring:

The Elasticsearch Java API is used to fetch the filtered logs from our Elasticsearch instance. As the majority of the diagrams used in the dashboard display historic and real-time data, the date histogram aggregation method offered by Elasticsearch is the main bucket aggregation used in our queries.

After the data is retrieved from Elasticsearch, it is inserted into a Data Transfer Object (DTO) which is returned as a response and received by the front-end where it is displayed in a number of diverse diagrams.

The endpoints that begin with the `http-` prefix optionally accept a Connector ID which is used to return data derived from logs originating from a specific Connector instead of aggregate data of all Connector logs.

- **GET /monitoring/network/http-monthly**
This endpoint uses the date histogram functionality offered by Elasticsearch in order to generate 14 data points, one for each of the last 14 days, which consist of a date and the total number of requests that took place that specific day.
- **GET /monitoring/network/http-transactions-sse-periodic**
This endpoint uses the date histogram functionality offered by Elasticsearch in order to generate 24 data points, one for each of the last 24 hours, which consist of a date and the total number of requests that took place that specific hour. Unlike previously, this endpoint implements Server-Sent Events, which allows the client to automatically receive updates from Spring Boot without constantly requesting updates from the server. This is achieved using the `SseEmitter`, a specialization of `ResponseBodyEmitter` for sending Server-Sent Events offered by Spring Boot. The endpoint is configured to publish updates every 5 seconds.

- [GET /monitoring/network/http-monthly-per-country](#)

This endpoint exploits the GeoIP metadata added to each log through our Logstash filter. It retrieves logs of requests that took place in the last 30 days and it utilizes bucket aggregation based on country code, essentially creating a bucket for each unique country which appears in the retrieved logs. Finally, it returns a DTO which includes an array of countries and the total number of requests originating from it.

- [GET /monitoring/network/http-bytes-sent](#)

This endpoint performs a combined aggregation which consists of a date histogram aggregation, limiting the retrieved results to the last 3 days, and a sub aggregation based on the bytes field. It returns a DTO consisting of an array of 3 elements, one for each day, and the total number of bytes sent that specific day.

- [GET /monitoring/network/http-response-codes](#)

This endpoint performs a combined aggregation which consists of a date histogram aggregation, limiting the retrieved results to the last 3 days, and a sub aggregation based on the response field. It returns a DTO consisting of an array of 3 elements, one for each day, each element including an array of unique response codes and the total number of requests that resulted in that specific response code.

- [GET /monitoring/network/connectors](#)

This endpoint returns a list of unique Connectors that have ever appeared in the Elasticsearch logs.

Analytics:

- [GET /analytics/anomaly_detection](#)

This endpoint retrieves predictions using the API provided by the analytics service and returns the prediction result as a response.

- [GET /analytics/security_report](#)

This endpoint first retrieves the list of IPs classified as abnormal by the anomaly detection machine learning algorithm through the API provided by the analytics service. Then for each IP that belongs to this list, it performs a combined aggregation, limiting the results to the last 24 hours. The first part of the aggregation queries the number of hits of each IP. Next, two sub-aggregations are performed: an aggregation that retrieves the country code of each IP, similarly to the http-monthly-per-country endpoint, and an aggregation that retrieves the total number of requests of each IP that resulted in an error. Using the query results, it constructs a DTO consisting of an array with one element per abnormal IP address, each element containing the IP address, the country code, the total number of requests and the total number of requests that led to errors in the last 24 hours.

All monitoring and analytics endpoints are secured using the HttpSecurity class provided by Spring Boot.

3.2.2 OneNet Data Analysis, Rating & Classification Tool

As technology advances and digitization expands, the number of security threats and attacks increase at an alarming rate. One such threat is the Distributed Denial of Service (DDoS) attack, which can cripple a web server

by flooding it with more traffic than it can handle. These attacks, commonly launched from multiple different IP addresses, are particularly challenging to detect and counter. With the widespread adoption of distributed computing, the need for an effective method to identify such malicious activities has never been more pressing.

The 'Data Analysis, Rating, and Classification Tool' is developed with this need in mind. This tool uses Machine Learning to detect anomalous behavior in IP traffic that may indicate a DDoS attack. By continuously learning from new data, it stays up-to-date with the evolving attack patterns. Its primary objective is to classify IPs as 'normal' or 'malicious' based on their behavior, which is determined by their request patterns. By identifying and categorizing malicious IPs in near real-time within a time scale of a few minutes, the tool aids in mitigating potential threats promptly and efficiently.

The underlying Machine Learning algorithm used in this tool is the Isolation Forest. This algorithm is well-suited for anomaly detection and is especially effective on large, high-dimensional datasets. It isolates anomalies instead of profiling normal data points, offering a unique approach to anomaly detection.

The Isolation Forest works by 'isolating' data points. It randomly selects a feature and then randomly selects a split value between the maximum and minimum values of that feature. This process creates a partition of the data. The logic behind the algorithm is that anomalies are easier to isolate compared to normal points— thus, anomalies will take fewer steps to 'isolate' than normal points, meaning they will have shorter paths in the forest. Therefore, the anomalies are the points that have a smaller path length on average.

The Isolation Forest is more efficient, and it has a lower computational cost compared to traditional clustering-based or distance-based anomaly detection algorithms, and, importantly for this tool, it works well with very unbalanced datasets, as is common in anomaly detection where the number of normal points often far outweighs the number of anomalies.

The tool's implementation heavily relies on Python, utilizing libraries such as Scikit-learn for the Isolation Forest algorithm, Pandas for data manipulation, and Numpy for numerical computation. It uses the Nginx access logs to train the model, focusing on features such as the number of requests from each IP, the number of unique user-agents, average response length, and the count of 4xx responses.

The implementation comprises several key functions, with each handling a specific aspect of the model lifecycle:

- `chunk(df, delta_in_min)`: This function creates batches of access logs based on the specified time window. The idea is to analyze the behavior within that window for anomaly detection.
- `extract_features(df, batch_minutes)`: Here, the features are extracted from the batches generated by the chunk function. These features will be used to train the Isolation Forest model.
- `anomaly_detection_job()`: This job runs every couple of days to retrain the model based on the latest Nginx logs. It retrieves the data, trains the model, and stores the model in Elasticsearch.

- `prediction_job()`: Running every few minutes, this job uses the stored model to predict the behavior of IPs appearing in the most recent batch of logs. It retrieves the model, predicts the states of the IPs, and stores the results in a PostgreSQL database.
- `trigger_training_cronjob(request)` and `trigger_prediction_cronjob(request)`: These API endpoints trigger the training and prediction jobs manually.
- `get_predictions(request)`: This API endpoint retrieves the most recent prediction results.

By using Elasticsearch to store models and logs, the tool can efficiently handle data volumes of up to several terabytes. Simultaneously, it also provides a mechanism to keep track of the model's evolution over time. This way, you can always roll back to a previous model if needed, ensuring full traceability. With Postgres, the tool efficiently manages the results of the anomaly detection, providing a clean and structured way to query prediction results.

The Data Analysis, Rating, and Classification Tool makes use of two cron jobs, which are tasks scheduled to run automatically at fixed times, dates, or intervals. This enables the system to continuously train and update its models and make predictions without manual intervention.

- **Training Job:** The first cron job is responsible for the training of the Isolation Forest model. As defined in the cron job schedule `'0 0 */{} * *'.format(number_of_days)`, this job runs once every few days to ensure that the model is trained on a comprehensive set of data, capturing a broader range of patterns and behaviors, while also balancing computational efficiency - the exact number is determined by the `number_of_days` variable. This job triggers the `anomaly_detection_job` function, which retrieves historic data, trains the Isolation Forest model, and stores the trained model in Elasticsearch. This automation ensures that the model is regularly updated and adapted to the latest patterns in the data.
- **Prediction Job:** The second cron job is set to run the `prediction_job` function every few minutes - the exact frequency is determined by the `batch_minutes` variable, as set in the cron job schedule `'*/{} * * * *'.format(batch_minutes)`. This job retrieves recent data, loads the trained model, and predicts the status (normal or malicious) of the different IPs that visit a Connector. The results are then stored in PostgreSQL, allowing the system to maintain a real-time record of potential anomalies.

By using these two cron jobs, the system is able to function continuously and autonomously. It maintains a cycle of learning from the past data and applying this knowledge to make near real-time predictions. This not only increases the efficiency of the tool, but also ensures its capability to promptly detect anomalies, thereby providing a robust solution for network security.

The PostgreSQL database is used to store the output of the predictive model. This output consists of timestamped predictions of whether an IP is 'normal' or 'malicious'. The Predictions model in PostgreSQL is designed to store this information efficiently and allows for easy querying. Here's a quick breakdown of the model's components:

- **timestamp_from** and **timestamp_to**: These fields store the start and end of the time window for which the batch of IPs were classified. It provides context to the prediction and allows for precise tracking of the network activity.
- **ip**: This field is an array containing all IPs that were active within the specified time window. The **ArrayField** is a Django-specific field for PostgreSQL, allowing multiple values to be stored in a single database field. In this case, it's used to store the list of IPs in the batch.
- **ip_status**: This field is another **ArrayField**, parallel to the **ip** field. It contains the status of each IP in the **ip** field, as predicted by the Isolation Forest algorithm. Each status is a float, either -1 for malicious or 1 for normal.

Storing the prediction results in PostgreSQL serves several purposes. It creates a permanent, queryable record of all IP classifications. This allows for future analysis, audits, or investigations. Additionally, it provides a structured and efficient way to access this data, which can be critical for near real-time network monitoring or incident response.

The schema allows for easy analysis of the data. For example, you could quickly pull all malicious IPs within a specific time frame or track the classification changes of a particular IP over time.

We conducted a comprehensive testing phase for the Data Analysis, Rating, and Classification Tool by deploying it within our connector. Initially, the tool was set to collect data under simulated normal traffic conditions, ensuring a rich dataset of typical network behaviors. After training the model on this 'normal' traffic, we introduced real-time simulated DDoS attack scenarios. The tool distinguished the sudden influx of abnormal traffic from the baseline, demonstrating its capability to promptly detect and flag potential threats within regular network activities. This hands-on approach validated the tool's practical effectiveness in a real-world environment, reinforcing its value as a reliable asset in our cybersecurity toolkit.

Also, if the anomaly detection system is part of a larger network security infrastructure, this data can be easily integrated and used by other components. For example, if a firewall or intrusion prevention system can access this data, it could use it to block or flag traffic from IPs that have been classified as malicious.

In summary, this tool offers an efficient and robust method to detect potential DDoS threats, enabling the proactive mitigation of such attacks. It demonstrates how machine learning can be leveraged to enhance network security, providing a valuable addition for the Connectors' cybersecurity toolkit.

3.2.3 OneNet Authentication/Authorization and Administration

Keycloak, an open source identity and access management tool, is our Identity Manager of choice. We have selected Keycloak mainly due to its ease of integration and its multitude of features which facilitate the storage and management of the OneNet user base and nullify the need for an in-house developed solution. Keycloak natively provides user-friendly web user interfaces through which a number of settings may be configured, namely the Account Console for users and the Admin Console for the administrator. We also determined that the Single-Sign On functionality offered by Keycloak is beneficial as it may be used to unify authentication and authorization across the entirety of the web applications which constitute the OneNet Interoperable Network of Platforms. A [main-authentication](#) realm has been created and configured to manage users and the [login-app](#) client is used to authenticate dashboard users. We employ a PostgreSQL database to be used for storage by the Keycloak instance.

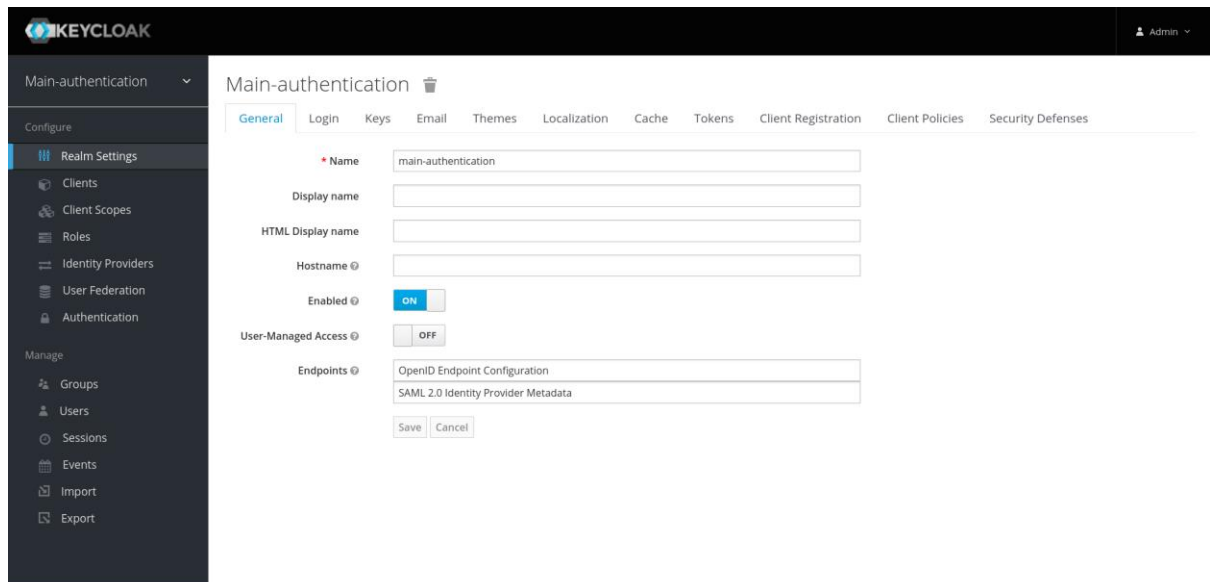


Figure 8: Keycloak Admin Console: Main Authentication Realm

Additionally, the [OneNet](#) client has been created in order to provide users with an alternative method of logging into the Connector user interface using the same credentials.

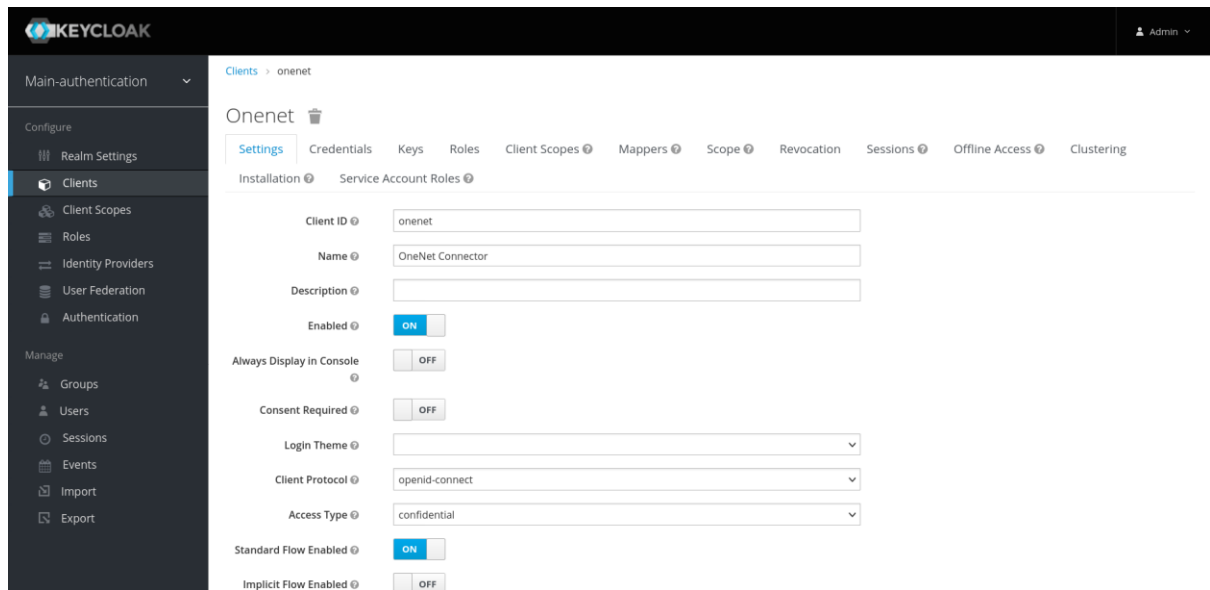


Figure 9: Keycloak Admin Console: OneNet Client

Keycloak provides an easily customizable login form which is common for all realm clients. Keycloak also provides a REST API which may be used not for authentication, among other purposes, which allows applications to implement their own login page.

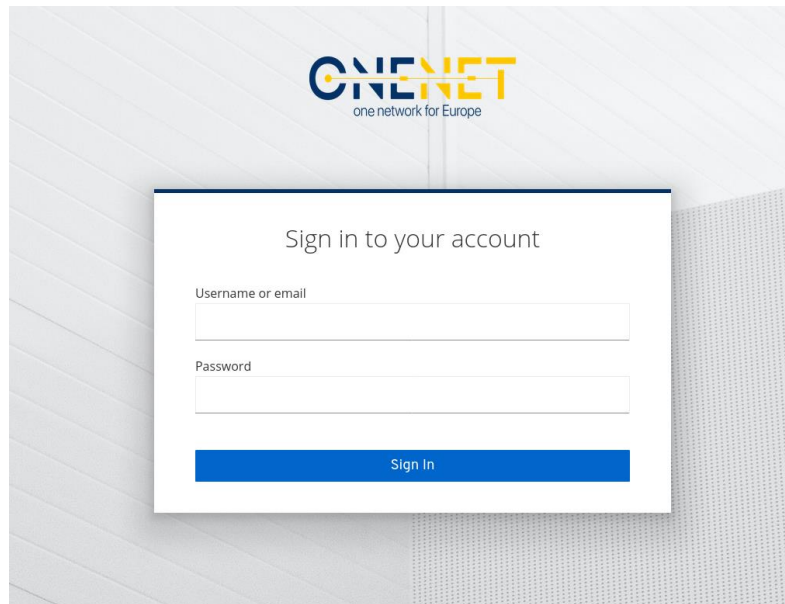


Figure 10: Keycloak Login Form

It is noted that Keycloak natively offers an account management page which allows users to manage account information, configure security settings and track account activity.

The Keycloak Account Management page may be accessed at the following URL:

<https://onenet-auth.europrojects.net/auth/realms/main-authentication/account/>

The Personal Info page allows users to manage their basic information such as their email address, first and last name.

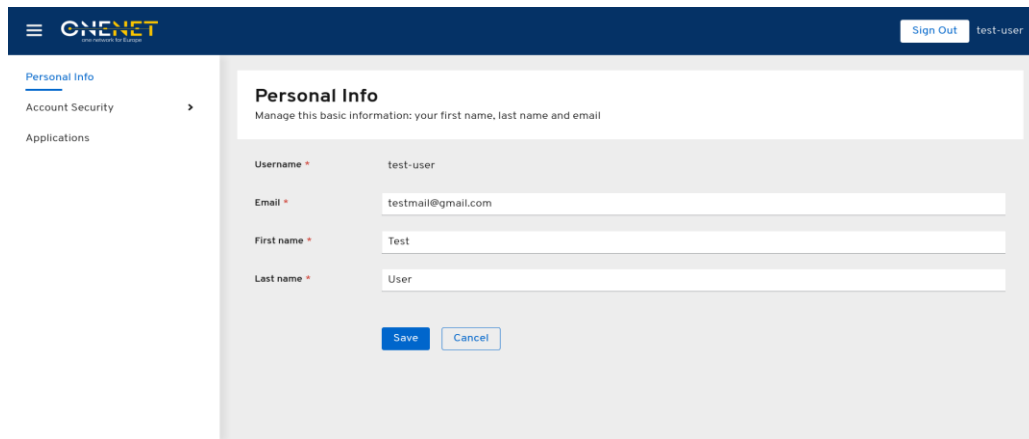
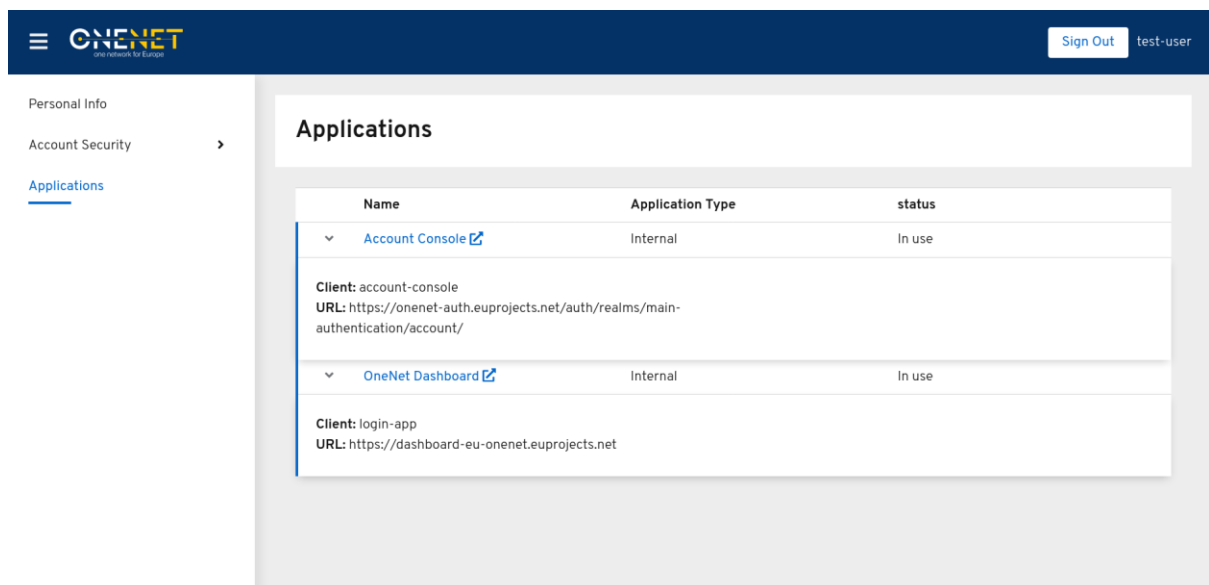


Figure 11: Keycloak Account Management: Personal Info

The Applications page renders a list of applications which the user is currently logged into. Internally, each application corresponds to a Keycloak client. For instance, in the following screenshot, the `account-console` client corresponds to the Account Management page while the `login-app` client, as mentioned above, corresponds to the Monitoring and Analytics dashboard. Client details such as the title, description and URL may be managed by the administrator through the Keycloak Admin Console.



Name	Application Type	status
Account Console	Internal	In use
Client: account-console URL: https://onenet-auth.euprojects.net/auth/realms/main-authentication/account/		
OneNet Dashboard	Internal	In use
Client: login-app URL: https://dashboard-eu-onenet.euprojects.net		

Figure 12: Keycloak Account Management: Applications page

The Device Activity page provides the capability to track device activity. It presents a list of devices currently logged into OneNet web applications, including information such as browser, browser version and operating system. Once again, each application corresponds to a Keycloak client. The user is able to track when each device

last accessed an application, when their access token was created and when it expires. It is also possible to sign out unfamiliar devices by clicking the Sign Out button.

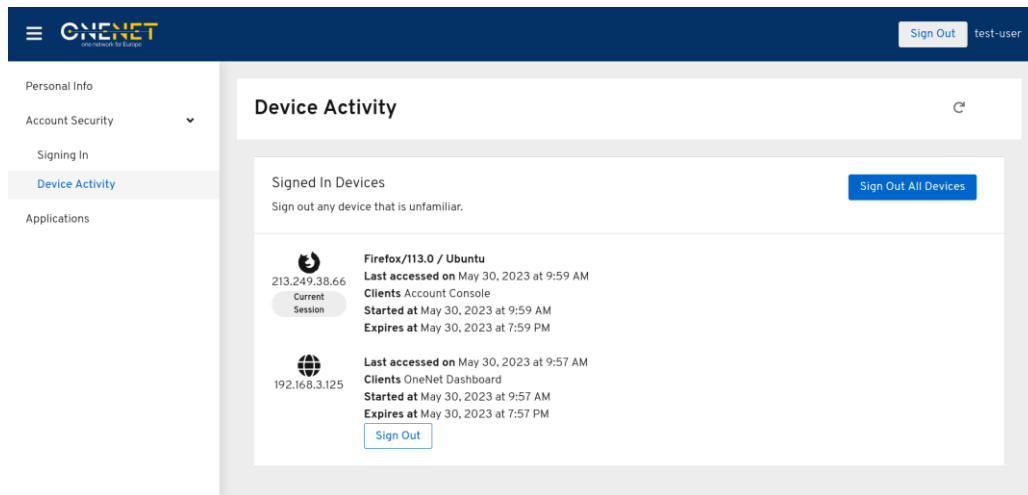


Figure 13: Keycloak Account Management: Device Activity

In the Signing In page, the user is able to track and configure authentication settings such as updating the password used to log into applications. The user may also optionally enable Two-Factor Authentication by setting up an Authentication Application such as FreeOTP or Google Authenticator on their smartphone.

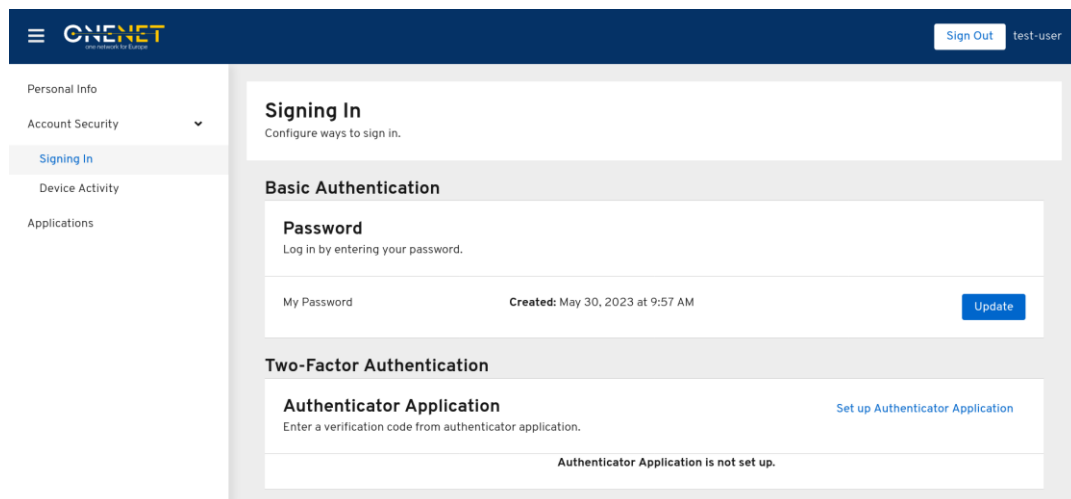


Figure 14: Keycloak Account Management: Signing In page

A custom Keycloak theme based on the OneNet color scheme and logo has been applied for the Login form and Account Management page.

The OneNet Monitoring and Analytics Dashboard provides statistics and tools which facilitate reviewing and evaluating the behavior of suspicious clients. Apart from the aggregated information displayed in the home page

of the dashboard, which will be presented in the next section, a security report may be generated and displayed by clicking the shield icon on the sidebar. The security report uses as input the list of IP addresses which were classified as abnormal by the analytics service. It displays the country, the number of requests made and the percentage of requests resulting in an error in the last 24 hours. The OneNet administrator may use the information provided by the security report to take corrective action against potentially malicious clients.

Security report	
Abnormal activity for the last 24 hours	
IP Address	212.101.173.114
Country:	 Greece
Total requests (24h):	538
Error frequency (24h):	33%
IP Address	206.71.50.230
Country:	 United States
Total requests (24h):	16
Error frequency (24h):	50%

Figure 15: Security Report

3.2.4 OneNet Monitoring and Analytics Dashboard GUI

The Network monitoring and analytics dashboard front-end is accessible at the following URL:

<https://dashboard-eu-onenet.euprojects.net/>

It is a Single Page Application (SPA) developed using Angular, a framework for developing dynamic web applications in the Typescript language. Regarding components, the Angular Material library was installed in the project and used throughout the project. ApexCharts.js and amCharts 5 were used for data visualization purposes. Components were created for all the required chart formats to facilitate chart creation. An nginx server is utilized for the hosting of the Angular front-end image. It has also been configured as a reverse proxy towards our Spring Boot back-end in order to provide a simple solution for errors related to Cross-origin resource sharing (CORS).

A simple login guard has been implemented to prevent the dashboard from being accessed without a valid access token. A non-authenticated user is greeted with a login page which communicates with the Spring Boot back-end authentication endpoints. Using a custom login page instead of the Keycloak native login page is convenient for Single Page Applications (SPA) such as the Monitoring and Analytics Dashboard.

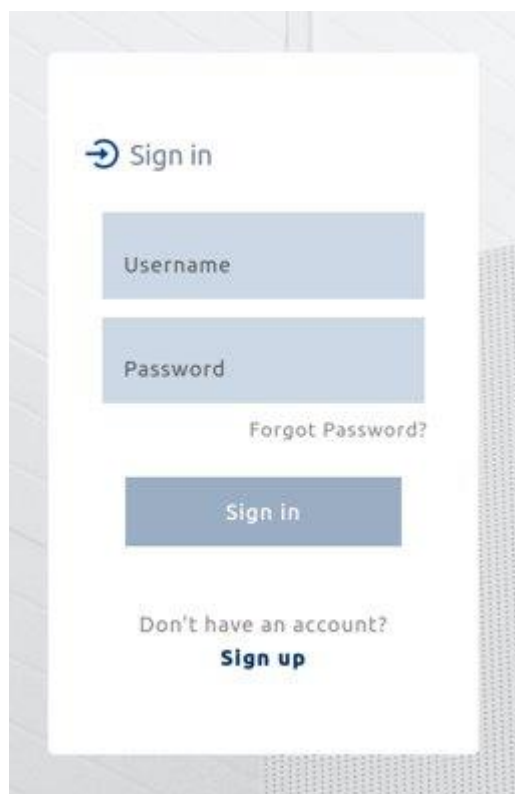



Figure 16: Dashboard login page

Users are able to create a new account by clicking on “Sign up”. For the user registration process, the required user information consists of a username, an email address and the account password, as seen in Figure 17.



→ Sign up

Username *

Email *

Password * 

Confirm Password * 

What kind of user are you?

☒ consumer

Register

Already have an account??

[Sign in](#)

Figure 17: Dashboard register page

Post-login, the user is redirected to the dashboard home page. The arrow button next to the user icon on the top right corner of the page allows the user to navigate to the account settings page or log out.

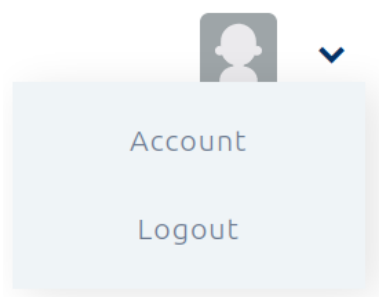


Figure 18: User menu

The account settings page includes the basic account information such as username and email and a button that redirects to the Keycloak Account Management page where account settings may be configured. In addition, it allows setting a profile picture and changing the user timezone.

home > settings

settings

basic-info

Avatar

time-zone

Username
test-user

Email
testmail@gmail.com

Manage account information

Figure 19: Account settings component

In the dashboard home page, the user is able to monitor Connector traffic at a glance. The following charts are displayed:

- **Access map chart**

A map chart which displays the number of hits per country in the last 30 days. The larger the circle in a specific country, the more requests originated from it. The chart allows zooming with the scroll wheel and panning by clicking and dragging on the chart surface.



Figure 20: Access map chart

- **Last 24 hours (live)**

A bar chart which displays the number of hits per hour. It uses SSEs to receive updates from the back-end approximately every 5 seconds.

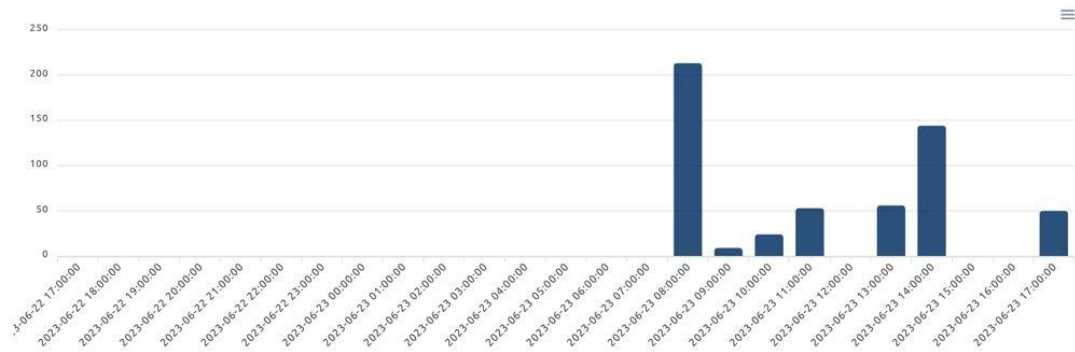


Figure 21: Last 24 hours (live)

- **Last 14 days**

A bar chart which displays the number of hits per day for the last 14 days.

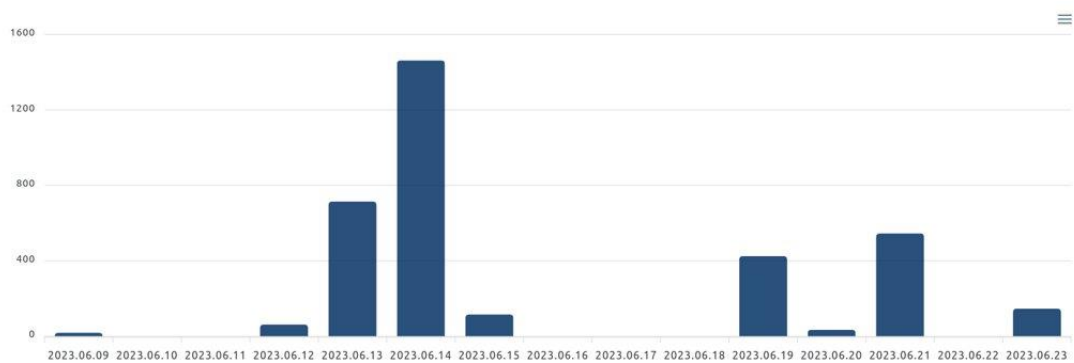


Figure 22: Last 14 days

- **Data sent over time chart**

A bar chart which displays the number of bytes sent per day for the last three days.

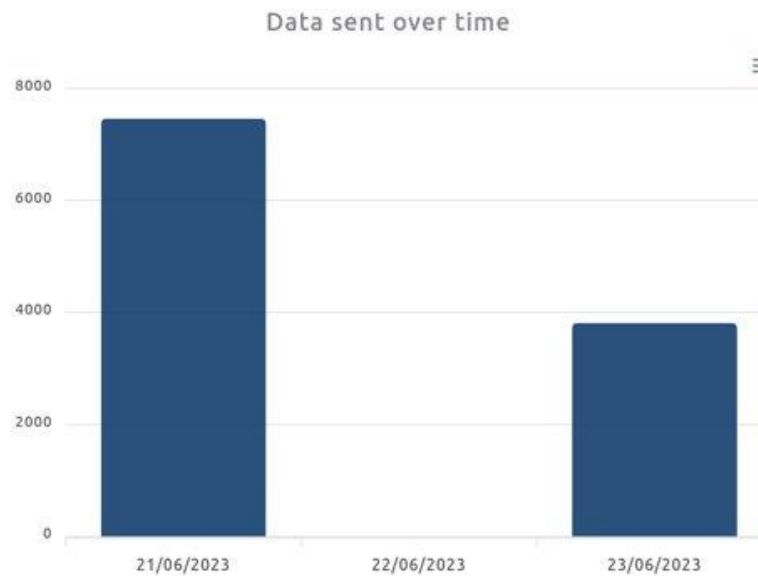


Figure 23: Data sent over time

- **Response codes over time**

A stacked bar chart which displays hits aggregated by response code for the last three days. Clicking on a legend label allows dynamically toggling the chart series that corresponds to a specific response code.

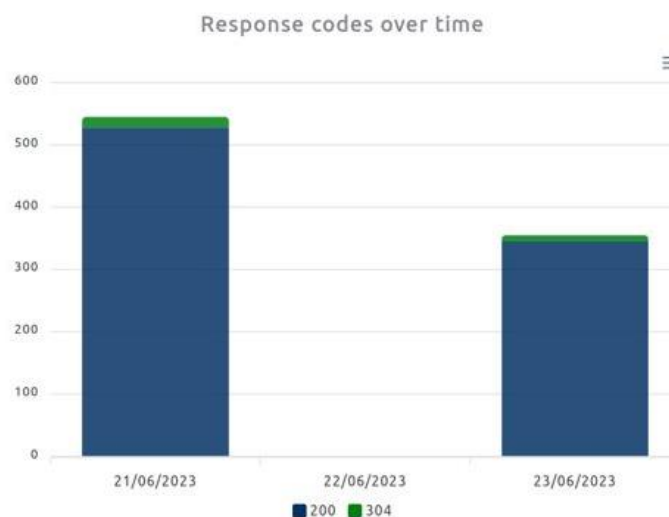


Figure 24: Response codes over time

- **Anomaly detection chart**

A bar chart which always displays at most two bars per date. The number of normal clients is denoted by a blue bar and the number of abnormal clients is denoted by a green bar. Hovering over a bar displays the distinct IPs that have been classified as normal or abnormal.

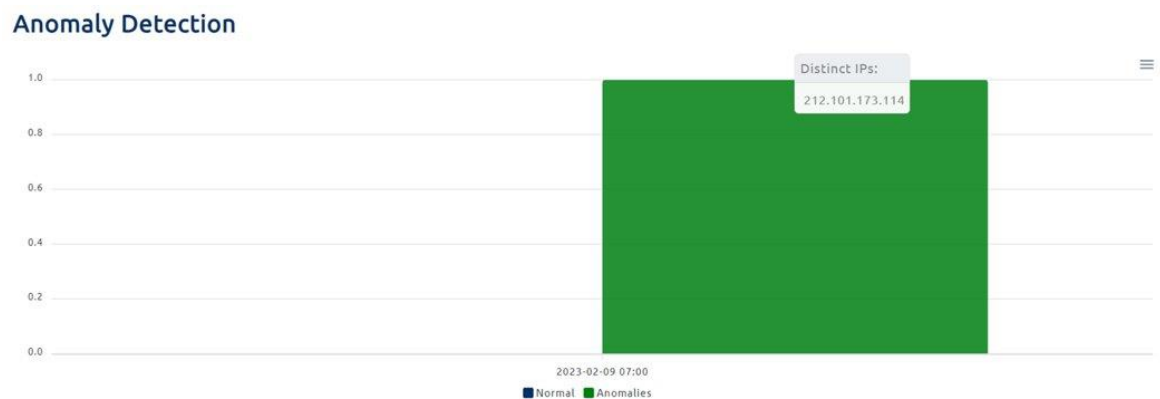


Figure 25: Anomaly detection

At the top of the dashboard, a dropdown menu is available which allows users to select a specific Connector whose statistics they are interested in. By default, all displayed information concerns all Connectors whose logs exist in the Elasticsearch index. By selecting a different Connector, the data is requested from the back-end using the selected Connector identifier and charts are updated with the new data accordingly.

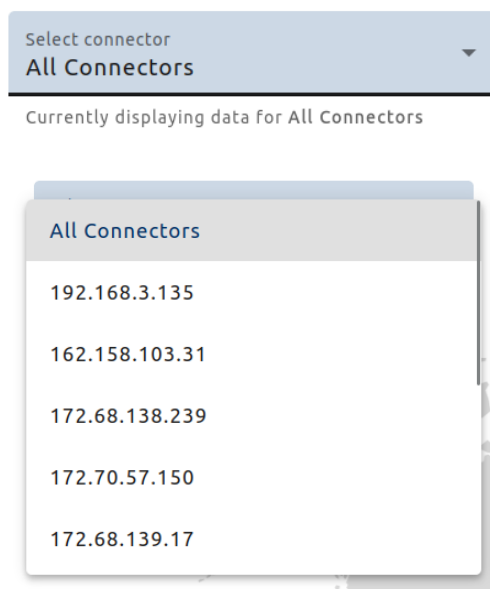


Figure 26: Connectors dropdown menu

4 Conclusions

In conclusion, the measures for legal, regulatory, privacy and cybersecurity compliance employed by the organizations and development teams involved in the implementation of various components of the OneNet Interoperable Network of Platforms have aided in its fortification against potential cybersecurity threats and the preservation of the confidentiality and integrity of handled data.

The requirements provided in the NISTIR 7628 document, introduced in D5.8 [3], were selected as the most relevant cybersecurity guidelines for to be considered and applied during the implementation of the OneNet project due to their relevance with smart grid information system security requirements. Accordingly, a number of methods were established throughout the project which fulfill the legal, regulatory, privacy and cybersecurity requirements. These methods have contributed to our deeper understanding of the security-centric approach required to render the platform resilient to the ever-evolving cybersecurity threats. Through the adoption and implementation of these cybersecurity measures, we have gained increased confidence in our ability to successfully address potential threats. The current report describes in detail which of the NISTIR 7628 requirements are addressed.

This deliverable accompanies the tool-agnostic architecture description which has been provided in D6.4 [4], along with packaging and deployment of the OneNet Monitoring and Analytics Dashboard solution. Furthermore, we delineate the design and implementation of OneNet Monitoring and Analytics Dashboard GUI. By filtering and indexing collected logs, performing the machine learning-assisted anomaly detection process, aggregating and visualizing historical and real-time data, we have implemented a monitoring and analytics dashboard GUI which provides insights on Connector usage patterns and enables us to promptly identify anomalies and potential malicious behavior. In the next and final steps of the project, we will continue with the integration of recently released components of the OneNet Middleware in the OneNet Monitoring and Analytics Dashboard, such as the Clearing House, in order to release the final version of the Dashboard which will augment the existing functionality by analyzing information regarding data exchanges in the OneNet network.

References

- [1] The Smart Grid Interoperability Panel–Smart Grid Cybersecurity Committee, NISTIR 7628 Revision 1, “Guidelines for Smart Grid Cybersecurity”, September 2014. [Online]. Available: <https://csrc.nist.gov/pubs/ir/7628/r1/final>
- [2] OneNet Deliverable D5.2, “OneNet Reference Architecture”. [Online]. Available: https://onenet-project.eu/wp-content/uploads/2022/12/OneNet_D5.2_v1.0.pdf
- [3] OneNet Deliverable D5.8, “Report on Cybersecurity, privacy and other business regulatory requirements”. [Online]. Available: https://onenet-project.eu/wp-content/uploads/2022/10/OneNet_957739_D5_8_v1_final.pdf
- [4] OneNet Deliverable D6.4, “AI, Big Data, IoT Orchestration Workbench”. [Online]. Available: https://onenet-project.eu/wp-content/uploads/2023/08/OneNet_D6.4_v1.0.pdf
- [5] CEN-CENELEC-ETSI Smart Grid Coordination Group, “Smart Grid Information Security (SGIS)”, December 2014. [Online]. Available: https://www.cenelec.eu/media/CEN-CENELEC/AreasOfWork/CEN-CENELEC_Topics/Smart%20Grids%20and%20Meters/Smart%20Grids/7_sgsg_sgis_report.pdf
- [6] OneNet Deliverable D6.5, “OneNet Reference Platform First Release”. [Online]. Available: https://onenet-project.eu/wp-content/uploads/2022/10/OneNet_D6.5_final_v1.1.pdf
- [7] “Horizon 2020 - Work Programme 2018-2020 Secure, clean and efficient energy”, September 2020. [Online]. Available: https://ec.europa.eu/research/participants/data/ref/h2020/wp/2018-2020/main/h2020-wp1820-energy_en.pdf