



# SMESEC

**Protecting Small and Medium-sized Enterprises digital technology through an innovative cyber-SECurity framework**

## D3.8 Public Report of SMESEC Framework and Unified Architecture

Document Identification			
<b>Status</b>	Final	<b>Due Date</b>	31/05/2019
<b>Version</b>	1.1	<b>Submission Date</b>	29/10/2019

<b>Related WP</b>	WP3	<b>Document Reference</b>	D3.8
<b>Related Deliverable(s)</b>	D3.3	<b>Dissemination Level (*)</b>	PU
<b>Lead Organization</b>	IBM	<b>Lead Author</b>	Fady Copty
<b>Contributors</b>	Benny Zeltser (IBM) Francisco Hernandez, Olmo Rayón (WoS) Samuel Fricker (FNHW) Jose Francisco Ruiz, Pablo Barrientos (Atos)	<b>Reviewers</b>	Christos Tselios (Citrix)
			Jose Fran. Ruiz (Atos)

### Keywords:

security, system, design, architecture, integration, WP3, requirements, stakeholder, goals, innovation, protection, defence, management, context, concept, pattern, composition, interface, rationale, sequence, response, forensics, orchestration, hub.

This document is issued within the frame and for the purpose of the SMESEC project. This project has received funding from the European Union's Horizon2020 Framework Programme H2020-DS-SC7-2016 under Grant Agreement No. 740787 and supported by Swiss State Secretariat for Education, Research and Innovation (SERI) under contract number 17.00067. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the European Commission.

This document and its content are the property of the SMESEC Consortium. All rights relevant to this document are determined by the applicable laws. Access to this document does not grant any right or license on the document or its contents. This document or its contents are not to be used or treated in any manner inconsistent with the rights or interests of the SMESEC Consortium or the Partners detriment and are not to be disclosed externally without prior written consent from the SMESEC Partners.

Each SMESEC Partner may use this document in conformity with the SMESEC Consortium Grant Agreement provisions.

(\*) Dissemination level.-**PU**: Public, fully open, e.g. web; **CO**: Confidential, restricted under conditions set out in Model Grant Agreement; **CI**: Classified, **Int** = Internal Working Document, information as referred to in Commission Decision 2001/844/EC.

## Document Information

List of Contributors	
Name	Partner
Fady Copty, Benny Zeltser	IBM
Francisco Hernandez, Olmo Rayón	WoS
Samuel Fricker	FHNW
Jose Francisco Ruiz, Pablo Barrientos	ATOS

Document History			
Version	Date	Change editors	Changes
1.0	04/09/2019	IBM	New section 5.3 Description of testing platform in SMESEC
1.1.	29/10/2019	ATOS	Final chek and submission to EC

Quality Control		
Role	Who (Partner short name)	Approval Date
Deliverable leader	Fady Copty (IBM)	29/10/2019
Technical manager	Christos Tselios (Citrix)	29/10/2019
Quality manager	Rosana Valle Soriano (Atos)	29/10/2019
Project Manager	Jose Fran. Ruíz (Atos)	29/10/2019

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	2 of 68
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1
				<b>Status:</b>	Final

# Table of Contents

Document Information .....	2
Table of Contents .....	3
List of Tables.....	5
List of Figures .....	6
List of Acronyms.....	7
Executive Summary .....	9
1 Introduction.....	10
1.1 Purpose of the document .....	10
1.2 Relation to other project work.....	10
1.3 Structure of the document .....	10
2 SMESEC Framework design .....	12
2.1 Composition view.....	12
2.2 Component View.....	15
2.3 Interface View .....	18
2.4 Deployment View.....	24
2.5 Communication bus security .....	26
3 SMESEC Framework user experience.....	27
3.1 Personas.....	27
3.2 Functions .....	29
3.2.1 Overarching User Interface Design Decisions .....	30
3.2.2 View: SME Security Dashboard .....	31
3.2.3 View: SMESEC Tools Dashboard .....	33
3.2.4 Evolved View: SMESEC Tools Overview.....	34
3.2.5 Evolved View: Tool View.....	35
3.2.6 View: Security Configuration View .....	35
3.3 Navigation .....	36
4 Design of SMESEC Framework Hub .....	38
4.1 System Architecture .....	38
4.2 Interface to other components .....	41
4.2.1 Input.....	41

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	3 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final

4.2.2	Output.....	41
4.3	Design rationale.....	42
5	Initial version of the SMESEC Framework prototype.....	43
5.1	Description and objectives .....	43
5.2	Functionalities and characteristics.....	43
5.2.1	My Status.....	44
5.2.2	Security Status Overview .....	45
5.2.3	“SMESEC@” .....	46
5.2.4	“SMESEC Tools”.....	47
5.2.5	“My Plugins” .....	49
5.2.6	“Security Configuration”.....	50
5.2.7	Access to Tools .....	50
5.3	Description of testing platform in SMESEC .....	51
5.3.1	FuzzMon.....	51
5.3.2	ExpliSAT.....	52
5.4	Development and integration environment .....	52
5.5	Integration methodology .....	52
5.6	Technical infrastructure.....	53
5.7	Authentication and security .....	53
5.8	Deployment and configuration.....	54
5.8.1	Deployment and configuration of the SMESEC Framework and client-side applications	54
5.8.2	Updating .....	55
5.9	API for external tools .....	56
5.10	Initial testing.....	58
6	Conclusions.....	60
7	References.....	61
8	Annex A. Detailed level description of the Input JSON Format.....	63

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	4 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final

## List of Tables

<i>Table 1: Component view summary</i>	18
<i>Table 2: Interfaces between SMESEC Framework components</i>	24
<i>Table 3: Deployment of SMESEC components</i>	26
<i>Table 4: Primary persona “Nicolas, the cybersecurity responsible (CISO) in the SME”</i>	28
<i>Table 5: Secondary personae who interact with the persona Nicolas.</i>	29
<i>Table 6: Common UI elements of the SMESEC Framework user interface.</i>	30
<i>Table 7: Elements of the SME Security Dashboard.</i>	32
<i>Table 8: Elements of the SMESEC Tools Dashboard.</i>	34
<i>Table 9: Elements of the SMESEC Tools Overview UI.</i>	35
<i>Table 10: Element Updates of the Tool View UI.</i>	35
<i>Table 11: Element Updates of the Tool View UI.</i>	36
<i>Table 12: Rule example - unwanted geolocation filtering</i>	39
<i>Table 13: Rule example - CPU and process understanding</i>	39
<i>Table 14: Rule example - rise awareness</i>	39
<i>Table 15: Alert JSON format to be used for reporting alerts to the SMESEC Hub.</i>	41
<i>Table 16: Detailed description of MISP format</i>	64

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	5 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final

## List of Figures

<i>Figure 1: High-level view on the methodology for designing and developing the SMESEC Framework</i>	10
<i>Figure 2: SMESEC composition view</i>	13
<i>Figure 3: Partial SMESEC setup</i>	14
<i>Figure 4: Basic SMESEC setup</i>	14
<i>Figure 5: SMESEC dashboard for the SME's CISO providing an actionable overview of the SME's security status.</i>	31
<i>Figure 6: Alert visualisation in the Dashboard</i>	33
<i>Figure 7: SMESEC Tools Dashboard providing overview of the detailed status and access to the SMESEC tools.</i>	33
<i>Figure 8: SMESEC UI navigation, providing alternative dashboard views and launching SMESEC framework tools (XL-SIEM and CYSEC provided as illustrative examples).</i>	36
<i>Figure 9: Architecture overview of the SMESE Hub</i>	38
<i>Figure 10: Rule configuration</i>	40
<i>Figure 11: Alert and recommendation example</i>	42
<i>Figure 12: My Status view</i>	45
<i>Figure 13: Security Status Overview view</i>	45
<i>Figure 14: SMESEC@ view</i>	47
<i>Figure 15: SMESEC Tools view</i>	48
<i>Figure 16: Plugin of alerts view</i>	50
<i>Figure 17: Quick link to tools</i>	51
<i>Figure 18: Data flow</i>	57
<i>Figure 19: External tools API component diagram</i>	57
<i>Figure 20: Example of alert in JSON format</i>	68

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	6 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final

## List of Acronyms

Abbreviation / acronym	Description
ADC	Application Delivery Controller
API	Application Programming Interface
AST	Application Security Testing
AV	Anti-Virus
CEO	Chief Executive Officer
CIRT	Cybersecurity Incident Response Team
CISO	Chief Information Security Officer
CISSP	Certified Information Systems Security Professional
CVE	Common Vulnerabilities and Exposures
DDoS	Distributed Denial of Service (network attack; also seen as DDSA)
DT	Deception Technology
Dx.y	Deliverable number y belonging to WP x
DoA	Document of Action
EC	European Commission
EPP	Endpoint Protection Platform
GRC	Governance, Risk Management and Compliance
HTTP	Hyper Text Transfer Protocol
HTTPS	HTTP Secure
IaaS	Infrastructure as a Service
IC	Innovation Committee
IDSISO	Intrusion Protection System International Organization for Standardisation
IDS	Intrusion Detection System
IoT	Internet of things
IP	Internet Protocol
ISFCISSP	Information Security Forum Certified Information Systems Security Professional
ISFAM	Information Security Focus Area Maturity
ISOISF	International Organization for Standardisation Information Security Forum
IT	Information Technology

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture	<b>Page:</b>	7 of 68
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU
	<b>Version:</b>	1.1	<b>Status:</b> Final

Abbreviation / acronym	Description
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
MISP	Malware Information Sharing Platform
MiTM	Man-in-the-Middle
MQTT	Message Queuing Telemetry Transport
OWASP	Open Web Application Security Project
PaaS	Platform as a Service
PCI-DSS	Payment Card Industry Data Security Standard
SaaS	Software as a Service
SIEM	Security Information and Event Management
SME	Small Medium Enterprise
SOC	Security Operations Centre
SSL	Secure Socket Layer
SUT	System Under Test
SW	Software
SWG	Secure Web Gateways
SWG	Secure Web Gateway
TaaS	Test-as-a-Service
UI	User interface
URL	Uniform Resource Locator
USG	Unified Service Gateway
UX	User Experience
VDI	Virtual desktop infrastructure
VM	virtual machine
VPN	Virtual Private Network
WAF	Web Application Firewall
WP	Work Package
XML	Extensible Mark-up Language
XMPP	Extensible Messaging and Presence Protocol

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture	<b>Page:</b>	8 of 68
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU
	<b>Version:</b>	1.1	<b>Status:</b>
			Final



---

## Executive Summary

---

The scope of this document is to describe the final version of the SMESEC framework architecture. The document will describe the changes and enhancements made to “SMESEC System Design” (D3.1) [1] and to “SMESEC Unified Architecture – First Internal Release” (D3.2) [2].

The SMESEC Framework architecture was enhanced to meet requirements gathered in previous deliverables and in the first-year review and documented in D3.1 and D3.2. Here we present the final design views of the SMESEC architecture.

We detail in this document the architecture of internal SMESEC component. We present the core components that deliver orchestration functionalities: SMESEC Hub and SMESEC extensions. And, we present the architecture of the SMESEC interface.

We describe in this document the enhanced user interface designed with special attention to user-experience and based on iterative discussions with the use-case partners.

Finally, we describe the Initial version of the SMESEC Framework prototype functionalities, integration and deployment.

This document will serve as basis for “SMESEC security Framework Final version” (D3.7).

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	9 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final

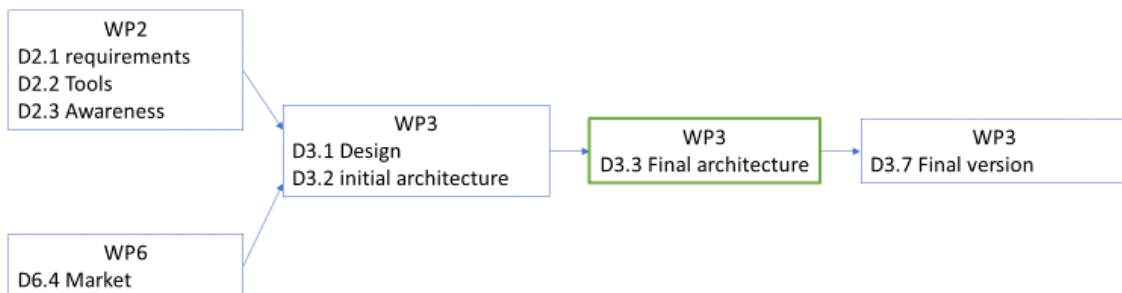
# 1 Introduction

## 1.1 Purpose of the document

The scope of the present document is to describe the final architecture of the SMESEC security framework. The document describes the final design views, the user interface design, the details of the orchestration components, and the SMESEC prototype. The architecture was designed to meet the requirements set in “SMESEC Unified Architecture – First Internal Release” (D3.2) [2]

## 1.2 Relation to other project work

As described in the DoA [9] , this document will report the final version of the SMESEC framework architecture. This document considers as an input the system design and requirements described in D3.2 and report as output the final architecture of the SMESEC Framework. The development of the final version of the SMESEC Framework will continue in task “From the prototype to the final SMESEC security framework.” (T3.4) and results will be reported in “SMESEC security Framework Final version” (D3.7).



**Figure 1: High-level view on the methodology for designing and developing the SMESEC Framework**

Figure 1 shows a high-level diagram of the process we followed, and the future work planned.

## 1.3 Structure of the document

This document is structured in 6 major chapters:

- **Chapter 1** is the introduction which describes the main objectives of this deliverable, relationship to other deliverables, and the following sections.
- **Chapter 2** describes final component, composition and interface views.
- **Chapter 3** describes final user-interface view and user experience.
- **Chapter 4** describes the design of the SMESEC Framework Hub.

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	10 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final

- **Chapter 6** describes the initial version of the prototype.
- **Chapter 5** draws conclusions and summarizes the deliverable.

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	11 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final

## 2 SMESEC Framework design

Several SMESEC design views were reported in D3.2 and D3.2, we hereby describe only the changes to those views.

### 2.1 Composition view

In this section we describe the final composition view of the SMESEC Framework. The composition view is depicted in Figure 2 and can be divided into several layers:

- The SMESEC infrastructure layer. The layer where all SMESEC infrastructure resides. This layer includes all the centralized functionalities like orchestration, authentication, configuration, and user interface.
- The SMESEC tools layers. The layer where all the tools of SMESEC reside. This layer includes all the tools provided by partners. These tools are external to the SMESEC infrastructure and are deployed at the partners' infrastructure. Also, the "external tool" resides in this layer.
- The SME infrastructure layer. The layer where all agents and endpoint security tools reside. This layer is the layer of tools integrated into the SME's infrastructure.

The architecture exposes the following user interfaces capabilities:

- Login. Supported by Keycloak[2] authorization and authentication mechanism. This is used to login into the SMESEC infrastructure and SMESEC tools layers. All components governed by Keycloak are denoted by a blue circle in the figure below.
- View attack chain alerts, recommendations and forensic reports. These are produced by the SMESEC Hub by orchestrating the various tools' results.
- Push notification to the user regarding alerts. These are produced by the SMESEC Hub.
- View alerts, view training, run testing and run patching. These are direct interfaces to SMESEC tool collection that are exposed to the user via the presentation interface of SMESEC.
- Edit SMESEC Hub predefined rules.
- Edit SMSEC Framework configuration, and part of the SMESEC tools' configuration (denoted by a green circle in the figure below).

The SMESEC Framework exposes the following interface categories: presentation interface and data interface. The presentation interface is used to propagate tool interfaces to the SMESEC interface, and the data interface is used for propagation of alerts and info from the tools and components into the SMESEC Hub. More details about the interface and communication module are to be found in the following sections.

The SMESEC infrastructure is composed of five main components:

- Presentation module responsible user interface interactions with underlying capabilities

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	12 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final

- Keycloak module responsible for authorization and authentication
- Configuration module responsible for configuring the infrastructure and tools
- The SMESE-Hub responsible for orchestration of tools
- Communication interface responsible of communication to the SMESEC tools layer

Further details regarding the components of all three layers are to be found in the component view section.

### SMESEC Framework

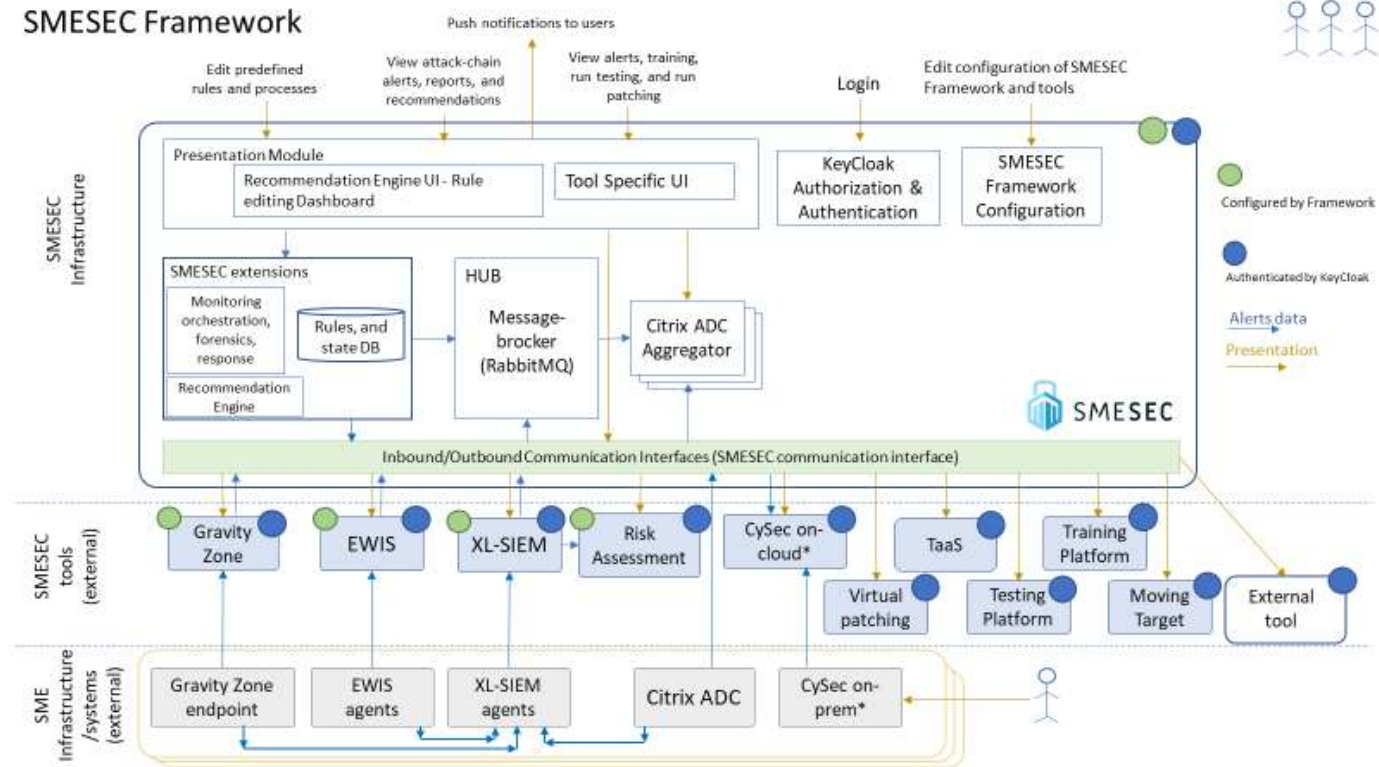
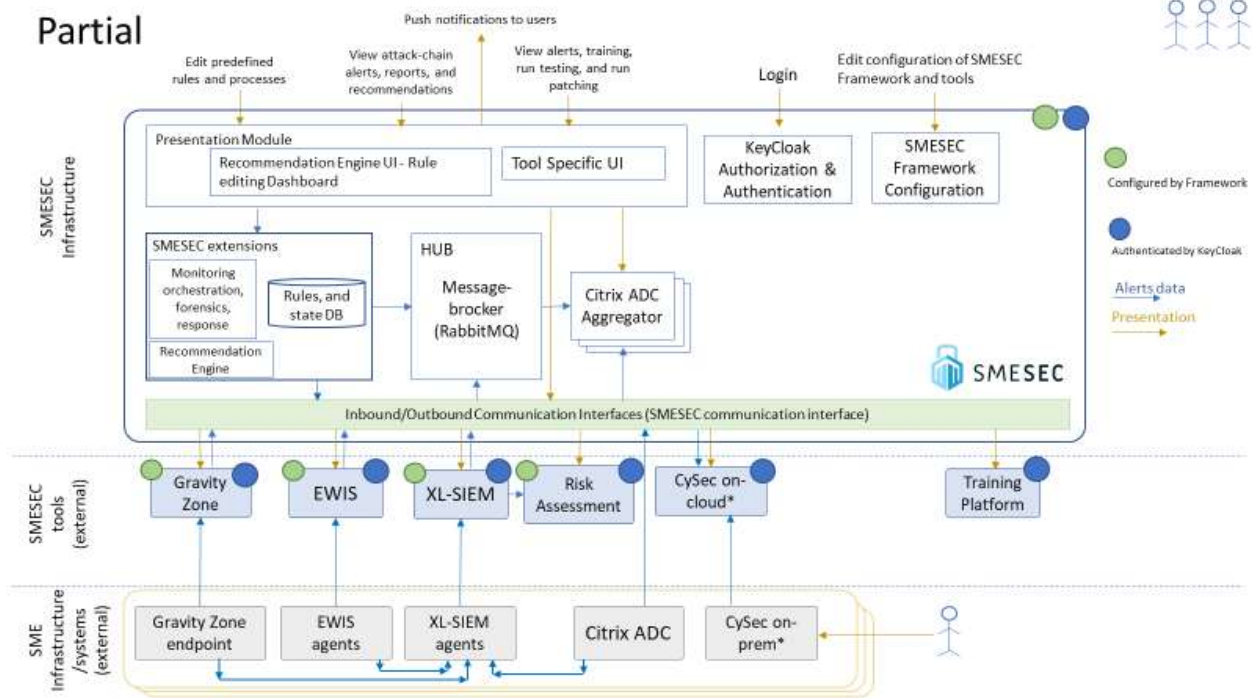


Figure 2: SMESEC composition view

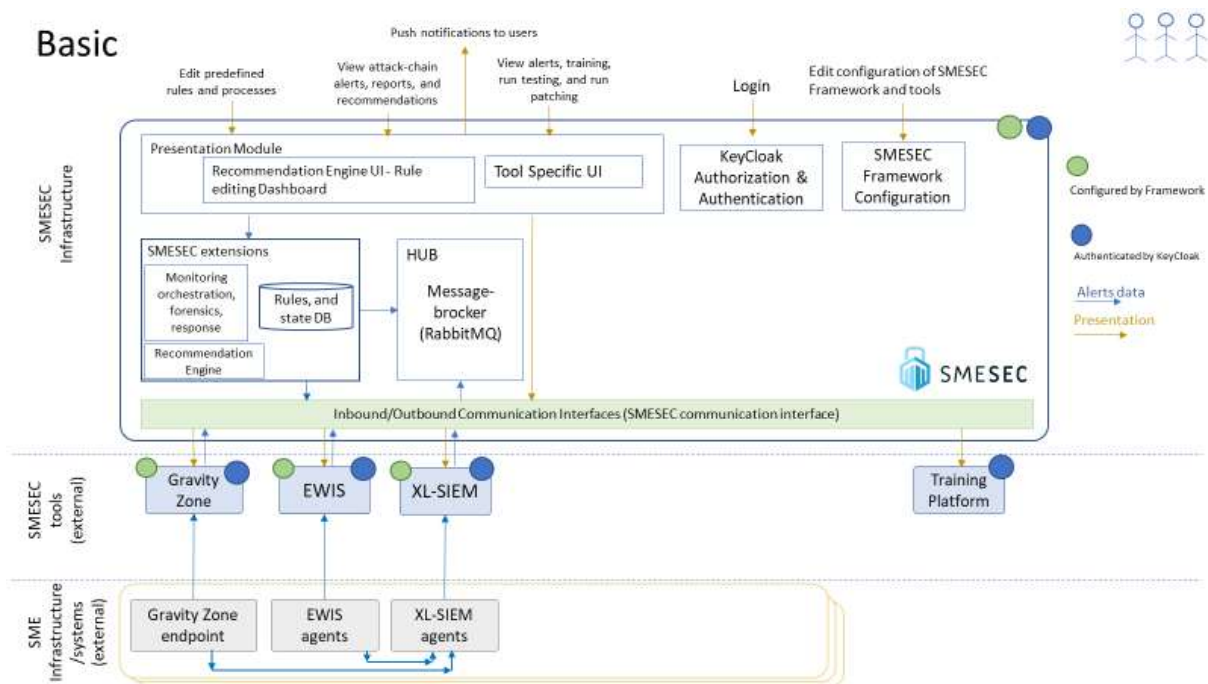
In addition to the above composition view the SMESEC architecture enables two other setups in order to address specific SMESEC business needs which may require the Partial and Basic setups of the SMESEC Framework. The Partial and Basic setups are depicted in Figure 3 and Figure 4. These setups provide partial and limited capability of the overall SMESEC Framework by limiting the availability of SMESEC tools to a single user with access into this setup. The governance of setup per user is done using Keycloak authentication and is the responsibility of the tool owner.

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture	<b>Page:</b>	13 of 68	
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	
	<b>Version:</b>	1.1	<b>Status:</b>	Final



**Figure 3: Partial SMESEC setup**

The main difference between those two setups is the collection of tools deployed at the SME’s premise and the service available for the user. The concept behind the basic setup is that it provides basic security with monitoring, endpoint, training and orchestration. The concept behind the partial setup is to add on top of that the risk assessment, advanced network security, and security expertise assessment tools.



**Figure 4: Basic SMESEC setup**

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture	<b>Page:</b>	14 of 68
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU
	<b>Version:</b>	1.1	<b>Status:</b>
			Final

## 2.2 Component View

The table below provides the detailed description of SMESEC Framework components. For each component description, responsibility, input, and output are described. Further interface details are found in the interface view section. The components can be divided into six main categories:

- *Data collection* contains the tools, XL-SIEM agents, and EWIS agents. These components are responsible to collecting SME's data, which will be analyzed later. The SMESEC Framework supports monitoring of various sources of information, for example, network data is monitored by Citrix ADC.
- *Endpoint protection and offline tools* consists of the following tools, Citrix ADC, Gravity Zone endpoint, TaaS, Virtual patching, Testing Platform, and Moving target. These tools strive to strengthen both the infrastructures located on the SME's premises, and the security of the products developed by the SMEs.
- *Data analysis* category aggregates all data collected by the data collection and endpoint protection tools, analyses the data and prepares it for the orchestration and presentation modules. The following are aggregators in the SMESEC Framework: Citrix ADC Aggregator, Gravity Zone, XL-SIEM, EWIS.
- *Training and security assessment* tools aim to assess both the security level of the SME's infrastructures, and the awareness and knowledge in security of the employees. Furthermore, the SMESEC framework contains tools such as CySec that sets itself a target to raise awareness and give a proper security education to the SME's employees. This category contains the following tools, CySec on-prem, CySec on-Cloud, Training Platform, and Risk Assessment Engine.
- The *orchestration* contains the SMESEC Hub and extensions module. This module consists of various plugins that use hardcoded rules, alongside AI-generated patterns, to analyze all the data collected and produce alerts and recommendations.
- The *presentation module* is the interface of the whole SMESEC Framework with its users. It gives an intuitive and easy to use customizable UI that assists the user is governing over the whole framework.

Follows a table of all components detailing the description and responsibility of each component alongside with a high-level description of the components' input and output.

Component	Description and responsibility	Input	Output
Citrix ADC	Intercepts network communication	Network traffic into SME's system	Information extracted from intercepted communication
Citrix ADC Aggregator	Aggregates information from Citrix ADC and produces alerts	Information extracted from intercepted communication	Aggregated information into data visualization

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture	<b>Page:</b>	15 of 68
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU
	<b>Version:</b>	1.1	<b>Status:</b> Final

XL-SIEM agents	Monitors on-disk log files	Log files from Citrix ADC , EWIS agents, and Gravity Zone endpoint	Information extracted from log files
XL-SIEM	Aggregates information from all XL-SIEM agents and produces alerts	Information extracted from log files from XL-SIEM agents	Alerts in either proprietary or MISP[10] format to the HUB
Risk Assessment Engine	Correlates vulnerability posture with XL-SIEM alerts, and estimates risk possible cost in USD	XL-SIEM alerts, and vulnerability status from user	Prioritization of alerts based on vulnerability posture, and estimate security breach possible cost in USD
Gravity Zone endpoint	Malware detection and vulnerability management	Files on disk	Analysis result of malware detection sent to Gravity Zone and point and to XL-SIEM agents
Gravity Zone	Aggregates information from all Gravity Zone instances and produces alerts	Analysis result of malware detection from Gravity Zone endpoint	Aggregated alerts from all malware detection instances
EWIS agents	Honey-pot integrated into customer premises	Network traffic, files downloaded and every activity in the honeypot	Extracted information sent to XL-SIEM agents and XMPP commands sent to EWIS
EWIS	Aggregates information from all EWIS agents and produce alerts	XMPP commands from honeypot	Based on the monitored communications, syslog information of security events sent to XL-SIEM, and logs to EWIS backend database
CySec on-prem	Create recommendations for SMEs and train SMEs	User input (as answers to questions)	Logs, answers, accounts one-way replication (upon request only) to CySec-on-Cloud
CySec on-Cloud	Create recommendations for SMEs and train SMEs	User input (as answers to questions)	List of Recommendations to SMESEC HUB as MQTT-SMESEC-MISP messages to a statically configured server, and list of recommendations to user
TaaS	Dynamic template-	Information about connected	Test results on TaaS Front End.

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	16 of 68
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1
				<b>Status:</b>	Final



	based testing platform	user from Keycloak.	
Virtual patching	Create a virtual patch based on user data	Labelled samples of inputs to user application	Log scanner to be deployed at user premises
Testing Platform	Test customer's code for security vulnerabilities	Request to download the tool	Sends back the tool that was requested
Moving target	Compiler plugin	Request to download the tool	Sends back the tool that was requested
Training Platform	Provide training for SME's employees	Request to view the online training	Interactive training
HUB	Collect alerts from all online monitoring tools	Alerts in either proprietary or MISP format from XL-SIEM	Alerts sent to Citrix ADC Aggregator
SMESEC extension	Analyse alerts collection to detect possible attack-chains, provide initial forensic and response capabilities, and provide recommendations based on orchestration of alerts and CySec results	(1) alerts collected in HUB (2) Requests from the presentation module for rule editing	Attack-chain alerts, initial forensics & response, and recommendations
Presentation module	Presents results to user and receives user requests	User interaction/input	(1) present results to user (2) forward requests to system SMESEC extensions for rule editing and presentation (3) Requests to SMESEC communication interface for presentation of various tools (4) Requests Citrix ADC Aggregator for data through the available API (5) send notifications to user
Keycloak	Manages authorization and authentication of SMESEC users	Login request	Authentication and authorization to the following components: Gravity Zone, EWIS, XL-SIEM, Risk Assessment, CySec on-cloud, Virtual patching, TaaS, Testing

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	17 of 68
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1
				<b>Status:</b>	Final

			Platform, Training Platform, Moving Target, External tool
Configuration	Update configuration of SMESEC framework	Configuration request	Configuration status for the following components: XL-SIEM, EWIS, Gravity Zone
Communication interface	Delegates communication between SMESEC tools and the SMESEC Framework	presentation requests and data transfer request	Presentation and data requests
External tool	TBD	TBD	TBD

**Table 1: Component view summary**

## 2.3 Interface View

The interface view is used to specify the internal interfaces of the SMESEC Framework. The SMESEC Framework consists of the SMESEC infrastructure, SMESEC tools that run on various cloud providers, and endpoint tools that run on the SME’s premises. The diverse execution environments require a delicate approach to the design of communication between the various entities.

All inbound and outbound communication to and from the SMESEC Infrastructure goes through the SMESEC communication interface. It presents a standardized way of communication with the SMESEC Infrastructure and plays the role of the “gatekeeper” by providing a secure two-way gate to and from the infrastructure. Further details regarding the communication bus are described in section 2.5

The communication between each endpoint tool on the SME’s premises with other tools provided by SMESEC partners is defined solely by the tool owners with the constraint of all communication to be secure to protect both the framework, and the potentially sensitive SME’s data.

Follows a table describing the interfaces between all components of the SMESEC Framework in detail. For each component a list of interfacing components is provided, a description of what requests does this component initiate, a description of what requests does this component serve, and details whether this component provide a presentation interface, data interface, authentication interface, configuration interface, and encryption on-rest.

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	18 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final

Component name	list of interfaces to other components	initiates	serve	presentation interface	data interface	on rest encryption	authentication interface	configuration interface
Citrix ADC	<ul style="list-style-type: none"> <li>SMESEC communication interface</li> </ul>	pushes information to (1) XL-SIEM agent (2) SMESEC communication (3) Citrix ADC Aggregator through the Citrix NITRO API	none	no	yes	no	no	no
Citrix ADC Aggregator	<ul style="list-style-type: none"> <li>SMESEC communication interface</li> </ul>	none	(1) consume Citrix-information routed by communication interface (2) provide data to HUB (3) provide data to presentation module	yes	yes	no	no	no
XL-SIEM agents	<ul style="list-style-type: none"> <li>XL-SIEM (on cloud)</li> </ul>	push information to XL-SIEM	consume information from (1) Gravity Zone endpoint (2) EWIS agent (3) Citrix ADC	no	yes	no	no	no

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture				<b>Page:</b>	19 of 68
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b> Final

XL-SIEM	<ul style="list-style-type: none"> <li>• SMESEC communication interface</li> <li>• XL-SIEM agents</li> </ul>	push alerts to the (1) SMESEC communication module (2) Risk assessment engine	(1) consume information from XL-SIEM agents (2) serve presentation requests from the communication module	yes	yes	no	yes	yes
Risk Assessment Engine	<ul style="list-style-type: none"> <li>• XL-SIEM</li> <li>• SMESEC communication interface</li> </ul>	none	(1) consume alerts from the XL-SIEM (2) serve presentation requests from the communication interface (3) consume survey answers from presentation module	yes	no	yes	yes	yes
Gravity Zone endpoint	<ul style="list-style-type: none"> <li>• Gravity Zone (on cloud)</li> <li>• XL-SIEM agents</li> </ul>	pushes information to (1) XL-SIEM agent (2) Gravity Zone	none	no	yes	no	no	no
Gravity Zone	<ul style="list-style-type: none"> <li>• SMESEC communication interface</li> <li>• Gravity Zone endpoint</li> </ul>	push alerts to the SMESEC communication module	serve presentation requests from the communication module	yes	yes	no	yes	yes
EWIS agents	<ul style="list-style-type: none"> <li>• EWIS (on cloud)</li> <li>• XL-SIEM agents</li> </ul>	syslog information to XL-SIEM, Logs to EWIS backend databases	XMPP commands from EWIS backend.	no	yes	no	no	Yes

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	20 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final

EWIS	<ul style="list-style-type: none"> <li>• SMESEC communication interface</li> <li>• EWIS agents</li> <li>• XL-SIEM</li> <li>• CITRIX (Netscaler)</li> </ul>	push alerts to the SMESEC communication module	serve presentation requests from the communication module	yes	yes	no	yes	yes
CySec on-prem	<ul style="list-style-type: none"> <li>• CySec on-cloud</li> </ul>	push status to CySec on-cloud	serve user requests via UI	yes	yes	no	no	no
CySec Cloud	<ul style="list-style-type: none"> <li>• SMESEC communication interface</li> </ul>	none	(1) serve presentation requests from the communication module (2) serve status update requests from the communication module	no	yes	no	yes	no
TaaS	<ul style="list-style-type: none"> <li>• SMESEC communication interface</li> </ul>	none	serve presentation requests from the communication module	yes	no	no	yes	no
Virtual patching	<ul style="list-style-type: none"> <li>• SMESEC communication interface</li> </ul>	none	serve presentation requests from the communication module	yes	no	no	yes	no

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	21 of 68
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1
				<b>Status:</b>	Final

Testing Platform	<ul style="list-style-type: none"> <li>SMESEC communication interface</li> </ul>	none	serve presentation requests from the communication module	yes	no	n/a	yes	no
moving target	<ul style="list-style-type: none"> <li>SMESEC communication interface</li> </ul>	none	serve presentation requests from the communication module	yes	no	n/a	yes	no
Training Platform	<ul style="list-style-type: none"> <li>SMESEC communication interface</li> </ul>	none	serve presentation requests from the communication module	yes	no	no	yes	no
HUB	<ul style="list-style-type: none"> <li>SMESEC extension</li> <li>Citrix ADC Aggregator</li> <li>SMESEC communication interface</li> </ul>	alert retrieval requests to Citrix ADC aggregator	(1) consume alerts from the communication module (2) serve alert fetch requests from SMESEC extensions module	no	yes	yes	no	no
SMESEC extension	<ul style="list-style-type: none"> <li>HUB</li> <li>SMESEC communication interface</li> <li>Presentation module</li> </ul>	status fetch request to the communication module and draws alerts from the HUB	presentation and configuration request from the presentation module	no	yes	yes	no	no

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	22 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final

presentation module	<ul style="list-style-type: none"> <li>• SMESEC extension</li> <li>• Citrix ADC Aggregator</li> <li>•</li> <li>• SMESEC communication interface</li> </ul>	initiates notifications to users	serve user requests of: (1) rule and process editing (2) presentation of alerts, reports, recommendations (3) presentation of tool specific UI	yes	yes	n/a	no	no
Keycloak	<ul style="list-style-type: none"> <li>• Gravity Zone</li> <li>• EWIS</li> <li>• XL-SIEM, Risk Assessment</li> <li>• CySec on-cloud</li> <li>• Virtual patching</li> <li>• TaaS</li> <li>• Testing Platform</li> <li>• Training Platform</li> <li>• Moving Target</li> <li>• External tool</li> </ul>	none	serve (1) user login requests (2) module authentication and authorization requests	yes	yes	yes	yes	no
configuration	<ul style="list-style-type: none"> <li>• XL-SIEM</li> <li>• EWIS</li> <li>• Gravity Zone</li> </ul>	initiate configuration requests	serve user configuration requests via UI	yes	yes	yes	yes	n/a

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture				<b>Page:</b>	23 of 68	
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final

communicati on interface	<ul style="list-style-type: none"> <li>Gravity Zone (on cloud)</li> <li>EWIS (on cloud)</li> <li>XL-SIEM (on cloud)</li> <li>CySec (on cloud)</li> <li>AngeEye, TaaS</li> <li>Training Platform</li> <li>Moving Target</li> <li>External Tool</li> <li>Citrix ADC</li> <li>SMESEC extension</li> <li>HUB</li> <li>Presentation Module</li> <li>Citrix ADC Aggregator</li> <li>Keycloak authorization and authentication</li> </ul>	(1) presentation requests to Gravity Zone, EWIS, XL-SIEM, Risk Assessment, CySec on-cloud, Virtual patching, TaaS, Testing Platform, Training Platform, Moving target and External tools (2) status fetch requests to CySec on cloud	consume alerts and data from Gravity Zone, EWIS, XL-SIEM, Citrix ADC	yes	yes	no	no	no
External tool	<ul style="list-style-type: none"> <li>SMESEC communication interface</li> </ul>	None	presentation requests from the communication module	TBD	TBD	TBD	yes	no

**Table 2: Interfaces between SMESEC Framework components**

## 2.4 Deployment View

The deployment of SMESEC Framework can be categorized into three categories:

- Deployment of SMESEC infrastructure
- Deployment of SMESEC tools
- Deployment of agents and endpoint tools

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	24 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final



The deployment of agents and end-point-security is necessary for collecting information from the SME systems, thus these tools are always deployed in the SME's premise. In addition to those, a CySec tool deployment into the SME's is optional for SME's who are concerned about privacy.

The SMESEC tools layer includes two categories: (1) online tools that aggregate information from agents and end-point-security tools (2) offline tools that are not dependent on the agents and end-point security tools. The tools in this layer are deployed on tool-providers' premises or on the cloud. One exception for this is the Citrix-aggregator that was deployed inside the SMESEC-infrastructure during the development of the prototype and is planned to become an independent deployment in the future.

The SMESEC infrastructure includes all the components responsible for the tools' collection and orchestration. This is deployed at ATOS premises and it supports multi-tenancy of SME's.

Follows a table describing the deployment details of the SMESEC Framework:

Component	Deployment	Multi-tenancy
Citrix ADC	SME's infrastructure	instance per SME
Citrix ADC Aggregator	SMESEC infrastructure	instance per SME
XL-SIEM agents	SME's infrastructure	multiple instances per SME
XL-SIEM	ATOS infrastructure	yes
Risk Assessment Engine	ATOS infrastructure	yes
Gravity Zone endpoint	SME's infrastructure	multiple instances per SME
Gravity Zone	BD infrastructure	yes
EWIS agents	SME's infrastructure	multiple instances per SME
EWIS	FORTH infrastructure	yes
CySec on-prem	SME's infrastructure	instance per SME
CySec on-Cloud	FHNW	yes
TaaS	EGM infrastructure	yes
Virtual patching	IBM Cloud	yes
Testing Platform	IBM Cloud	instance per SME
moving target	IBM Cloud	instance per SME

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	25 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final

Training Platform	UoP infrastructure	yes
HUB	SMESEC infrastructure	yes
SMESEC extension	SMESEC infrastructure	yes
presentation module	SMESEC infrastructure	yes
Keycloak	SMESEC infrastructure	yes
configuration	SMESEC infrastructure	yes
communication interface	SMESEC infrastructure	yes
External tool	TBD	TBD

**Table 3: Deployment of SMESEC components**

## 2.5 Communication bus security

All SMESEC tools connected to the communication-bus must apply mutual (two way) Keycloak authentication. All communication between the communication bus and the SMESEC tools, regardless of the underlying protocol, must be encrypted using TLS1.2 or above.

The Security responsibilities of the SMESEC communication between SMESEC infrastructure and SMESEC tools are distributed among components as follows:

- Tool security is the tool provider's responsibility.
- It is the communication bus responsibility to apply network security.
- HUB-security: It is the HUB responsibility to validate their input against possible attacks.

The bus must support multi-tenancy and load balancing.

The bus must apply network security measures:

- Install and configure a firewall for hardening
- Input validation for security purposes (i.e. DoS attack detection, discovery and response of potential malicious activity)

All security events reported by bus security (example firewall), must be logged to a central logging service, and saved for 90 days.

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	26 of 68
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1
				<b>Status:</b>	Final

## 3 SMESEC Framework user experience

Usability is a key requirement of the SMESEC Framework. Since our solution aims being used by organizations which employ the full spectrum of professionals, from novice to cybersecurity experts, it is important that the ability of using and understanding the framework to be as high as possible regardless of the level of ones' familiarity with the core principles of cybersecurity as a whole. This is the main reason behind having the SMESEC Framework offer by design a unified interface for all integrated tools.

To design the proper user experience (UX) and identify the target personas of the SMESEC Framework we have extended the previously conducted interviews with the use-case SME partners with joint application design workshops for the definition of the user interfaces in collaboration with the cybersecurity responsible in these SMEs.

Unchanged in comparison to D3.2, the user interface (UI) supports the unification, while offering the needed simplicity, with a tool launcher and an integration hook for the SMESEC tools' information and display of events. Changed is, however, a switch in focus: awareness about SME-specific cyber threats and understanding how to improve the SME's security is now put into the user's focus. This change resulted from workshops with the SMESEC use case SME's and reflect their need for immediate access to the value-creating elements of the SMESEC framework.

This section describes the targeted user personas, gives an overview of the provided UI functions, specifies the navigation, and describes the details of the UI views. The specification refers back to D3.2 section 4.2 *User Interface* [2] and describes additions or modifications to the previously specified user experience design.

### 3.1 Personas

The UI has been designed for use by specific personae in the SME. According to the so far collected survey data and by following the SMESEC fast ramp-up recommendations for cybersecurity capability improvement in the SME, we can expect that in each end-user SME there will be a person appointed for handling cybersecurity in the SME. We call this person the Chief Information Security Officer, or CISO, referring to the corresponding formal job description that is often used in large companies. To describe in a specific way how to use the framework we defined a user called "Nicolas" who has this responsibility.

Table 1 specifies the characteristics and offers background of Nicolas, the SME CISO, who is the main user of the SMESEC framework and the tools that are included in the framework. It is to be noted that personas are not identical to the user roles. User roles represent privileges and responsibilities of a person at a given time, while personas present characteristics, goals, desires, and expectations of a person.

Attributes	Values
Name	Nicolas
Responsibility	Cybersecurity responsible in the SME (Chief Information Security Officer, CISO)

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture	<b>Page:</b>	27 of 68				
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final

Characteristics	Curious about cybersecurity, while being afraid that it might be too complicated. Appointed by the SME management to handle the topic of cybersecurity in the SME. Cybersecurity is a side-topic and not the sole work priority for the person.
Background	Marginal knowledge of cybersecurity, which is improving through the use of the SMESEC Framework. First-time and repeated occasional user of the SMESEC framework without preparatory training.
Tasks	Expected to assess threat, vulnerability, and protection status; decide about and set cybersecurity controls; involve the SME’s employees; and report to the management.
Expectations	Guidance with support of the personal learning of cybersecurity and how to address cybersecurity with the SMESEC framework. Minimal effort to obtain and maintain overview and awareness of cybersecurity in the SME and to report about it.

**Table 4: Primary persona “Nicolas, the cybersecurity responsible (CISO) in the SME”**

Additional users that have other responsibilities are of relevance in the extended SMESEC framework use. Their enablement is the concern of Nicolas’s use of SMESEC and his personal interaction with these users both online and offline. In comparison to D3.2, the current version presented here provides an extension of roles and traits and an explanation of their involvement for protecting the SME under the leadership of Nicolas. Table 2 gives an overview.

Name	Role and Traits	Consideration in SMESEC
Philippe	Chief Executive Officer (CEO) leading the strategy and operations of the SME and being legally responsible for its overall welfare. He understands the importance of cybersecurity but is too busy to manage it sustainably.	Philippe is provided read-only access to the SMESEC framework. Nicolas regularly creates reports for Philippe that are based on the security status information provided by SMESEC.
Claudia	Employee of the SME and expected to be aware of cyber threats and is expected to adhere to safe practices that help to protect the SME from these cyber threats. She wants to do her work well and expects that others are helping her.	Claudia is provided access to trainings offered by the SMESEC tool Securityaware.me and polls generated from CYSEC. Nicolas coordinates the interaction of Claudia with SMESEC online and offline.
Julien	Employee of the SME with a careless attitude and potentially malicious intentions that might hurt the welfare of the SME.	Nicolas works with Julien the same way he works with Claudia. In addition, Nicolas activates and configures monitoring tools of the SMESEC framework, such as GravityZone, the EWIS honeypot, and NetScaler to detect insider attacks and uses IBM AntiROP and TaaS to prevent potential backdoors in the SME’s products and services.

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture	<b>Page:</b>	28 of 68	
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	
	<b>Version:</b>	1.1	<b>Status:</b>	Final

Martin	Cybersecurity expert and consultant offering personalized help and advice for SMEs. Martin’s business is cybersecurity, and he brings in-depth practical experience as a CISO and member of cybersecurity incident response teams (CIRT).	Nicolas works with Martin for receiving specialized advice beyond what the SMESEC framework provides and support for responding to cybersecurity incidents. To assist Martin, Nicolas shares the company profile, maturity information, event logs collected and with the SMESEC framework.
Jose	Cybersecurity reference person and community manager interacting with stakeholders and advancing cybersecurity for SME in Europe.	Nicolas understands that creating industry-wide awareness and advancing cybersecurity technology depends also on his company. For that reason, he opts in to sharing anonymous data about events and capability improvements with the open SMESEC community.
Christos	Cybersecurity external auditor responsible to verify that the SME is compliant with regulations	Nicolas works with Christos for compliance auditing requested by important customers. Nicolas uses the SMESEC framework to implement some of the controls, practices, and trainings that Christos suggests.

**Table 5: Secondary personae who interact with the persona Nicolas.**

While the persona definition is based on results from discussions with the SMESEC use case SMEs and cybersecurity experts, validation of the characterised collaboration between Nicolas and the rest of the SME is subject to the validation trials planned for the year 3 of the SMESEC project, where the SMESEC framework is brought into use by the SMESEC use case SMEs and the third-party SMEs that joined the SMESEC project as third-parties through the open call.

## 3.2 Functions

In the deliverable D3.2, a SMESEC Framework user interface (UI) was proposed that primarily consistent of a launcher and static information about the SMSEC tools that can be accessed through the launcher. To draw advantage of the cybersecurity situation sensed by the tools and cybersecurity knowledge of the SMESEC consortium of what the SME should do in that situation, the Framework UI was extended to be a one-stop dashboard for the Chief Information Security Officer (CISO) of the SME. The dashboard now offers an overview of the sensed situation as well as recommendations of actions that may be useful in the SME’s situation.

The dashboard was developed in collaboration with the use case SMEs members of the SMESEC consortium. To co-design helped eliciting latent tacit needs that have not been discovered earlier. It also allowed taking advantage of the SMESEC tool provider’s expertise and testing of ideas of how an effective workplace can be designed that is usable and useful for the SME CISO.

The SMESEC framework UI offers a comprehensive overview of indicators and events that reflect the status of the SME, provides recommendations for actions that may be useful in the SME’s situation, and provides access to the SMESEC tools. This section describes the enhanced designs of the views and offer tables with function catalogues provided by these views, including targeted benefits as rationales. The tables also offer traceability with the list of functions defined in D3.2 through consistent use of identifiers and motivates the modifications, respectively extensions

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	29 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final

### 3.2.1 Overarching User Interface Design Decisions

The idea of the one-stop dashboard for the SME CISO implied restructuring of the user interface. An enhanced header and footer have been designed and an evolved navigation paradigm defined. Figure 5 illustrates the new user interface paradigm, including the use of the evolved header and footer. Section 3.3 describes the evolved navigation paradigm. The following table describes the functions provided by the common UI elements.

Functions	Targeted Benefits	Implementation
FWUI-UC01: Single Sign-on (unchanged)	Allow access to all SMESEC-protected information and tools with one effort.	Keycloak-based authentication and authorisation.
FWUI-P01.1: Quick Links (unchanged)	Support exploration of tools. Support visual inspection and correlation of tools' settings and outputs.	Integrated tool display with header indicating chosen tool, tool display, and accordion with compact tool launcher.
FWUI-UC03-V2: Display cybersecurity KPI and alarms for the SME (replacing FWUI-UC03)	Awareness of current threats and protection status, and guidance of the CISO with little expertise with recommended actions.  The SMESEC tools report the following information: real-time security-related events, discovered vulnerabilities, the SME's security maturity, alerts, and trends.  Several overviews are provided: SME-centric security status overview, SMESEC tools-centric security status overview, overview of SMESEC tools and plugins (FWUI-UC02), and the SME's security configuration for parametrising the SMESEC framework.  Flexibility for consortium to add and remove SMESEC tools	Mashup of UI controls rendered by the various SMESEC tools.
FWUI-P01.4: Header Bar (modified)	The human end-user knows he is using the SMESEC framework.  The human end-user can navigate across the views: a personal view with favourite indicators, the security status overview of the SME, the status of the SMESEC tools, the introduction and selection of the SMESEC tools, a selection of the framework plugins, and the security configuration of the SME.	HTML always shown on top of screen.
FWUI-P01.5: Footer (added)	The human end-user knows that the SMSEC framework is delivered by trustworthy parties.	HTML with logo and disclaimers at the bottom of the page.

**Table 6: Common UI elements of the SMESEC Framework user interface.**

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	30 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final

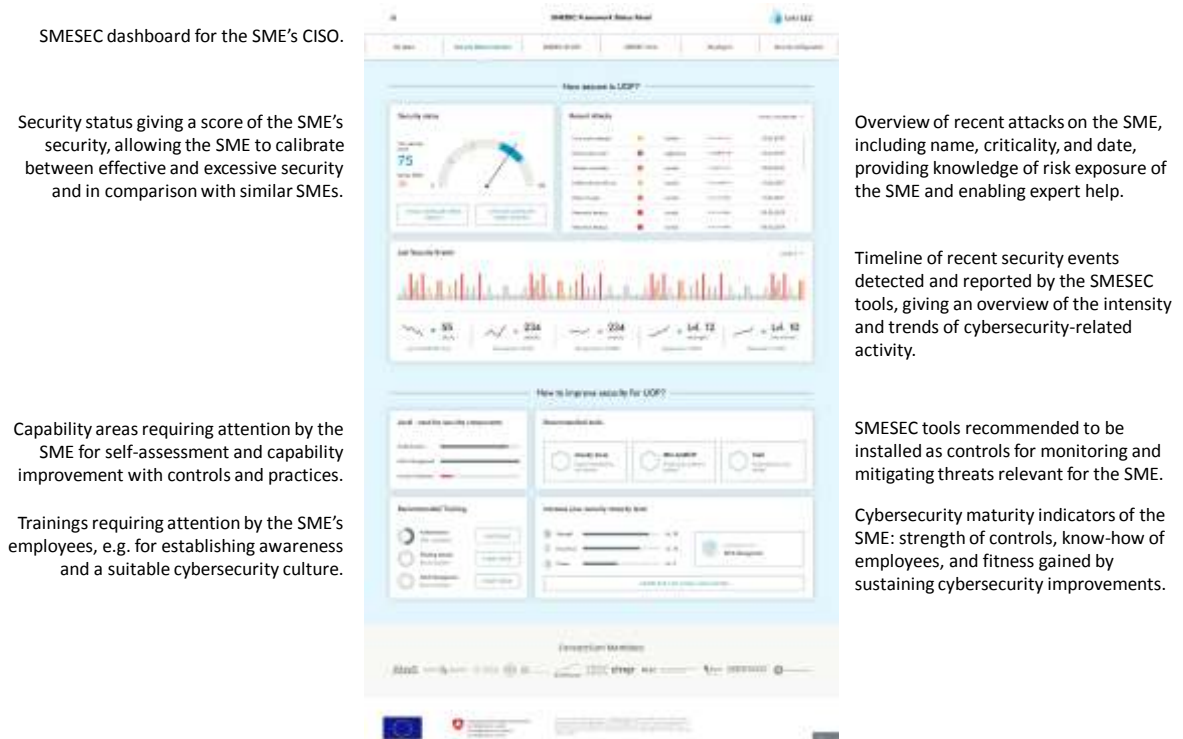
Unchanged is the entry point to the SMESEC framework. The UI will be accessible by signing in on the SMESEC homepage and used as a public-cloud service offered by the SMESEC consortium. The UI may also be deployed on-premise and used in conjunction with locally deployed tools.

Unchanged are the compatibility requirements with the end user’s machines: the UI of the SMESEC framework can be used with a browser and offers visual and textual interfaces and allows display of dynamic tool-rendered information with an iframe-based approach. Firefox [4]v63 and Chrome [5] v70 on Windows 10 [6] and Safari v12 [7] MacOS High Sierra [8] planned for acceptance tests. This allows integration of the SMESEC Framework tools and integration of security for the SMESEC Framework.

The following subsections describe the views of the SMESEC framework, including the SME Security Dashboard, the SMESEC Tools Dashboard, the SMESEC Tools view, the Tool view, and the Security Configuration view. The personal view follows the principle of the SMESEC Tools Dashboard view but displays only those tools that were selected by the human end-user. The Framework Plugins view follows the principles of the SMESEC Tools view but lists framework plugins for activation.

### 3.2.2 View: SME Security Dashboard

Changed is the presentation of the information display and access to the SMESEC tools. The use case SMEs that participated in the design activities underlined the importance of the one-stop information display for awareness about the SME’s cybersecurity status and recommendations of what should be done to improve the status. Figure 5 shows the re-designed screen and describes its elements that put actionable information into the focal point and explanation of the SMESEC tools into the background (accessible through a menu item).



**Figure 5: SMESEC dashboard for the SME’s CISO providing an actionable overview of the SME’s security status.**

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	31 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final

The SME Security Dashboard shown in Figure 5 offers the SME’s CISO a one one-page overview with indicators that give answers to the questions *how secure is the SME* and *how to improve the security of the SME*. The page is needed by the SME’s CISO to obtain awareness of currently relevant threats, assess the SME’s security status, and plan actions for improving the security. The dashboard is composed of widgets that are rendered the plugins of the SMESEC framework, hence offers flexibility for the SMESEC providers to adapt and evolve the view based on lessons-learned from validation or new capabilities that emerge, e.g. as a result of integrating third-party capabilities into the framework through the open call.

The following table specifies the sections of the view. FWUI-P01.4-5 had been specified in D3.2 already. FWUI-P03.1-3 are new or modified elements.

View	Section	Targeted Benefits	Implementation
FWUI-P03: Security Dashboard	FWUI-P01.4: Header Bar (modified)	The human end-user knows he is using the SMESEC framework. The human end-user can navigate across the views.	HTML always shown on top of screen.
FWUI-P03: Security Dashboard	FWUI-P01.5: Footer	The human end-user knows that the SMSEC framework is delivered by trustworthy parties.	HTML with logo and disclaimers at the bottom of the page.
FWUI-P03: SME Security Dashboard	FWUI-P03.1: Dashboard (replacing FWUI-P01.3)	The human end-user is aware of the threat exposure and protection of the SME and know recommended actions for improving the SME’s security.	Integration of plugin-rendered HTML.
FWUI-P03: SME Security Dashboard	FWUI-P03.2: Tool-Launching Recommendations	The human end-user knows recommended actions and can launch Securityaware.me, respectively CYSEC with the right context to implement the action.	Integration of tool-rendered HTML and links to the matching tool context.
FWUI-P03: SME Security Dashboard	FWUI-P03.3: Alert Display	The human end-user is aware of alerts.	Integration of plugin-rendered HTML and link to the matching tool for resolving the alert.

**Table 7: Elements of the SME Security Dashboard.**

Changed is also the presentation of alerts. These are placed at the top of the security overview and presented in a way that capture the immediate attention of the CISO. Figure 6 illustrates the presentation of an alert. The CISO can acknowledge alerts, drawing attention to the tool that was generating the alert.

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	32 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final



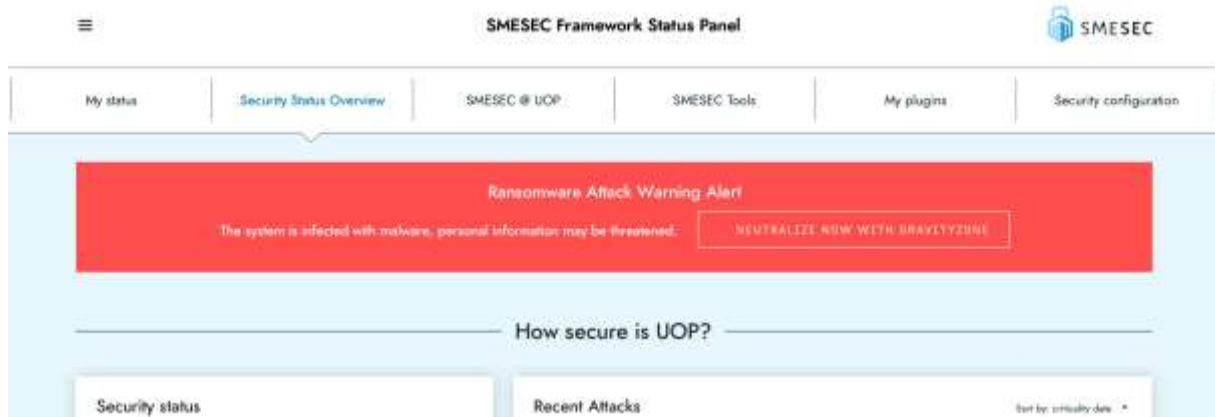


Figure 6: Alert visualisation in the Dashboard

### 3.2.3 View: SMESEC Tools Dashboard

The SMESEC framework also provides the CISO with the ability to switch from the SME-centric view shown in Figure 5 to the SMESEC framework tool-centric view shown in Figure 7. In this tool-centric view, each tool has available a frame for visualizing its most relevant KPI, events, alerts, and other information. This content is intended to provide the CISO with the ability to develop awareness of what information can be gained with each of the tools that are activated in his SMESEC framework configuration.

Dashboard and launcher providing overview of the detailed status and access to the SMESEC tools.

Trends of recent security events detected and reported by the SMESEC tools.

Overall threat level and timeline of recent security events detected and reported by XL-SIEM.

Overview of recent attacks detected by GravityZone.

Cybersecurity maturity indicators of the SME provide by the CYSEC coach.

Histogram of events detected and reported by NetScaler.

Overview of test results reported by TaaS for multiple products, including a timeline indicating test result trends.



Timeline of attacks detected by the honeypot EWIS.

Capability areas requiring attention by the SME for self-assessment and capability improvement.

Status of training delivery to the SME's employees reported by Securityaware.me.

Access to the AntiROP compiler plugin.

Figure 7: SMESEC Tools Dashboard providing overview of the detailed status and access to the SMESEC tools.

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture	<b>Page:</b>	33 of 68	
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	
	<b>Version:</b>	1.1	<b>Status:</b>	Final

The SMESEC Tools Dashboard shown in Figure 7 offers the SME’s CISO a one one-page overview with indicators that give answers to the questions *what the cybersecurity status is according to the tools*. The page is needed by the SME’s CISO to obtain awareness of the SME’s capabilities and cyber threats for the cybersecurity themes represented by the SMESEC tools activated by the CISO. Also, this dashboard is composed of widgets that are rendered the plugins of the SMESEC framework, hence offers flexibility for the SMESEC providers to adapt and evolve the view based on lessons-learned from validation or new capabilities that emerge, e.g. as a result of integrating third-party capabilities into the framework through the open call.

The following table specifies the sections of the view. FWUI-P01.4-5 had been specified in D3.2 already. FWUI-P03.1-3 are new or modified elements.

View	Section	Targeted Benefits	Implementation
FWUI-P04: SMESEC Tools Dashboard	FWUI-P04.1: Dashboard (replacing FWUI-P01.3)	The human end-user is aware cybersecurity status according to the activated SMESEC tools.	Integration of plugin-rendered HTML.
FWUI-P04: SMESEC Tools Dashboard	FWUI-P03.2: Tool-Launching Recommendations	The human end-user knows recommended actions and can launch any SMESEC tool through the respective widget used for information display.	Integration of tool-rendered HTML and links to the matching tool.
FWUI-P04: SMESEC Tools Dashboard	FWUI-P03.3: Alert Display	The human end-user is aware of alerts.	Integration of plugin-rendered HTML and link to the matching tool to fix the alert.

**Table 8: Elements of the SMESEC Tools Dashboard.**

The remaining tabs offer access to configuration of SMESEC, including the activation of tools and plugins and the security configuration of the SME that SMESEC uses to adapt the functionality of the tools and recommendations.

### 3.2.4 Evolved View: SMESEC Tools Overview

As specified in D3.2, the One-page Overview offers an introduction with quick-links allowing to understand the scope of the page, a section with hierarchical structuring and explanation of the SMESEC framework, and a dashboard with SMESEC tool KPIs and alerts. The view is renamed to *SMESEC Tools Overview* to communicate the intention of the view in comparison to the other views. Further, the SMESEC Tools Overview is extended to provide the CISO with the ability to activate and deactivate tools, affecting the dashboard views. The function catalogue is updated follows.

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	34 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final

View	Section	Targeted Benefits	Implementation
FWUI-P01: SMESEC Tools Overview	FWUI-P01.2: Tool Launcher (unchanged)	The human end-user gets introduced into the topic of cybersecurity through the categorical grouping of SMESEC tools into sections and subareas that offer short explanations.  The human end-user can launch a tool with a full understanding of the tool's scope.	HTML with cross-page links.
FWUI-P01: SMESEC Tools Overview	FWUI-P01.3: Dashboard (removed)	This section is replaced by the SME Security Dashboard view FWUI-P03.	-
FWUI-P01: SMESEC Tools Overview	FWUI-P01.6: Activation (added)	The human end-user is able to activate a tool to be considered in the Dashboards or to deactivate it.	Checkboxes.

**Table 9: Elements of the SMESEC Tools Overview UI.**

### 3.2.5 Evolved View: Tool View

The Tool View FWUI-P02 is adapted to the new visual framework that includes the Header Bar FWUI-P01.4 and Footer FWUI-P01.5. The iframe used for integrating the tool's user interface is adapted accordingly. The function catalogue is updated as specified in the following table.

View	Section	Targeted Benefits	Implementation
FWUI-P02: Tool View	FWUI-P02.1: Tool UI (unchanged)	The human end-user uses the launched SMESEC tool without distracting cluttering.	iframe integration of tool front-end.

**Table 10: Element Updates of the Tool View UI.**

### 3.2.6 View: Security Configuration View

The new Security Configuration View FWUI-P05 provides the CISO with the ability to configure the SME's preferences and profile. Upon the installation or activation of the SMESEC framework in the SME, the Security Configuration is the first view shown to the user. Once the security configuration is complete enough, the first view changes to be the SME Security Dashboard.

A CYSEC coach is used for guiding the user through the security configuration, allowing the user to understand what the settings imply for managing cybersecurity in the SME. The deliverable D3.4 section 3.5.6 describes functionality and visual appearance of the CYSEC coaches.

The function catalogue of the Security Configuration View is as specified in the following table.

View	Section	Targeted Benefits	Implementation
------	---------	-------------------	----------------

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	35 of 68
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1
				<b>Status:</b>	Final

FWUI-P05: Security Configuration View	FWUI-P02.1: Tool UI	The human end-user uses the launched CYSEC without distracting cluttering. The visual appearance is as in D3.4 Section 3.5.6.	iframe integration of tool front-end.
--	------------------------	---	---------------------------------------

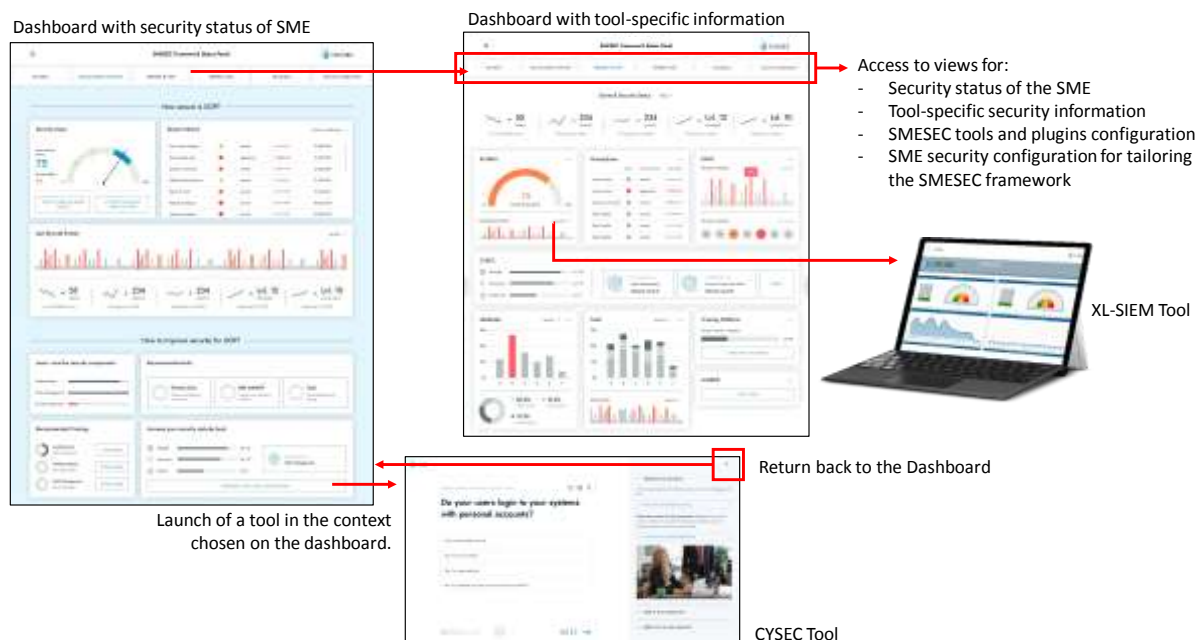
**Table 11: Element Updates of the Tool View UI.**

### 3.3 Navigation

To account for the complexity of cybersecurity monitoring and management, the SMESEC framework UI offers a simple navigation approach based on two paradigms: a) menu bar to switch among views that offer rich information display and access to tools, plugins, and configurations, and b) a launcher allowing to run a tool from the specific context provided by the widget in the view’s mashup.

In comparison to D3.2, the navigation paradigm was adapted due to the relative importance of the information display for awareness over explanations of the SMESEC tools’ capabilities.

The following figure shows the screens and navigation pathways.



**Figure 8: SMESEC UI navigation, providing alternative dashboard views and launching SMESEC framework tools (XL-SIEM and CYSEC provided as illustrative examples).**

Users who visit the framework, will get on the main page a one-page overview the cybersecurity status of their SME (1) that answers the questions of *how secure is your SME* and *how to improve the security of your SME*. The first question is answered with a score of the SME’s security, an overview of recent attacks, and the timeline of recent security events detected by the SMESEC framework. The second question is answered with the current status of self-assessment, capability improvement, and training provision and recommendations of next steps.

The user is offered the choice through a top-level menu bar to switch to the SMESEC tools and drill down into the detailed statuses reported by the SMESEC tools and to see how each of the tools has contributed to the security status assessment (2). The view (2) also allows inspecting the status of tools

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture	<b>Page:</b>	36 of 68
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU
	<b>Version:</b>	1.1	<b>Status:</b> Final

that did not report their measurements or results for the aggregate overview of (1). For example, TaaS is a tool used to manage cybersecurity as a quality aspect for product or service development and does not require the immediate reaction, e.g. of a cybersecurity incident response team (CIRT), to resolve observed problems.

The view (2) can also be used to launch any of the SMESEC tools. Shown as an illustrative example in Figure 8 is the launch of the XL-SIEM tool (3). The view (1) can also be used for launching tools but is restricted to specific training actions with Securityaware.me or self-assessment and capability improvement actions with the cybersecurity coach CYSEC (4). Any of the SMESEC tools runs standalone from the end-user's perspective and can be opened and closed in parallel to the SMESEC dashboard. Tools that represent plugins into other frameworks, such as the IBM AntiROP that is used as a compiler plugin, offer download instructions and *how to use* guidelines.

As a final option, the user can use the menu bar for accessing the remaining views (5).

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	37 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final

## 4 Design of SMESEC Framework Hub

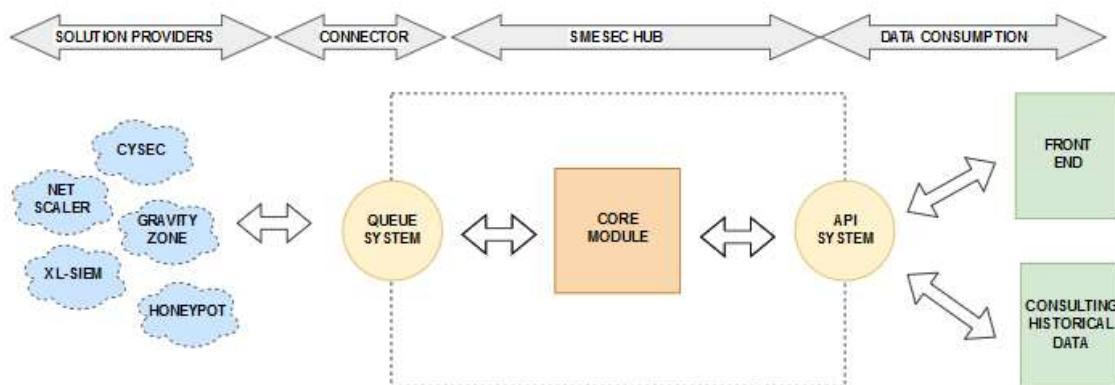
The real added-value of the SMESEC Framework should go beyond the simple addition of different tools. On the other hand, the expected lack of expertise on cybersecurity of the end users makes essential to have an intuitive system to raise alarms and suggest corrective actions when needed. At this juncture, the SMESEC Framework Hub has been designed to cross link heterogeneous data coming from the cybersecurity tools and provide simple insight into the SME infrastructure. The cybersecurity events are here filtered and reframed to render them understandable to any person.

From a practical point of view, this means that data are centrally collected, processed and forwarded to the SMESEC end user, providing clear advice of what actions to take at any time. In the end, the Hub will contribute to attaining a friendly and intuitive front end, and a major cybersecurity awareness of the end users.

### 4.1 System Architecture

In this section, the Hub internal architecture is presented, going into details of the different modules and technologies upon which the functionalities rest. The general overview is shown in Figure 9.

The whole system relies on four stages: (i) data acquisition from the tools, (ii) data concentration and aggregation (queue system), (iii) data processing (core module) and (iv) data extraction (API system).



**Figure 9: Architecture overview of the SMESEC Hub**

In the first stage, the tools deployed at the SME acquire the security events and any additional data necessary to evaluate the overall infrastructure status. Later, some of this information is sent to a queue system based on Rabbit MQ technology by using a predefined JSON format. This is the real entrance door to the Hub. Here, the input information is processed within the core module applying the so-called business rules, which are the cornerstone to attain the expected functionalities. It should be pointed out that this element is agnostic and independent on specific use case restraints. Hence, it can be easily adapted to provide the expected output.

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture	<b>Page:</b>	38 of 68	
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	
	<b>Version:</b>	1.1	<b>Status:</b>	Final

The business rules implement the logic that substantiates the Hub response, and they are intended to provide meaningful and straightforward information to the end users, as well as a corrective action against attacks. Their implementation is being implemented in a sequential fashion in difficulty level. At this stage, only “simple” business rules are operative by processing single data sources. This will be however enriched with “enhanced” ones once two or more data sources can be correlated. The operative business rules at M24 are indicated below. They have been designed as proof-of-concept elements.

<b>BR_01_LOC: Unwanted geolocation filtering</b>	
Objective:	Identify connections from non-allowed locations
Partners involved:	Citrix, Atos, FHNW
Schema:	Citrix ADC (IP list) → XL-SIEM → SMESEC Hub → Front end
Input:	Alarm over suspicious IP address
Processing:	The system has a list of allowed IPs. The system also calculates the location of the IP that is being sent from the firewall
Output:	If the algorithm detects any location different from the allowed ones, an alert is launched

**Table 12: Rule example - unwanted geolocation filtering**

<b>BR_02_PRC: CPU and processes understanding</b>	
Objective:	Identify over working from our systems due to malicious processes on them
Partners involved:	All Providers
Schema:	Any Provider → XL-SIEM → SMESEC Hub → Front End
Input:	Form all the servers that are being monitored, we obtain CPU usage and running processes. Data is sent periodically
Processing:	There are two different events that trigger an alarm: the CPU goes over a predefined threshold or any of the processes matches any of the prestored as malicious in the system
Output:	If any of this happens an alert will be risen

**Table 13: Rule example - CPU and process understanding**

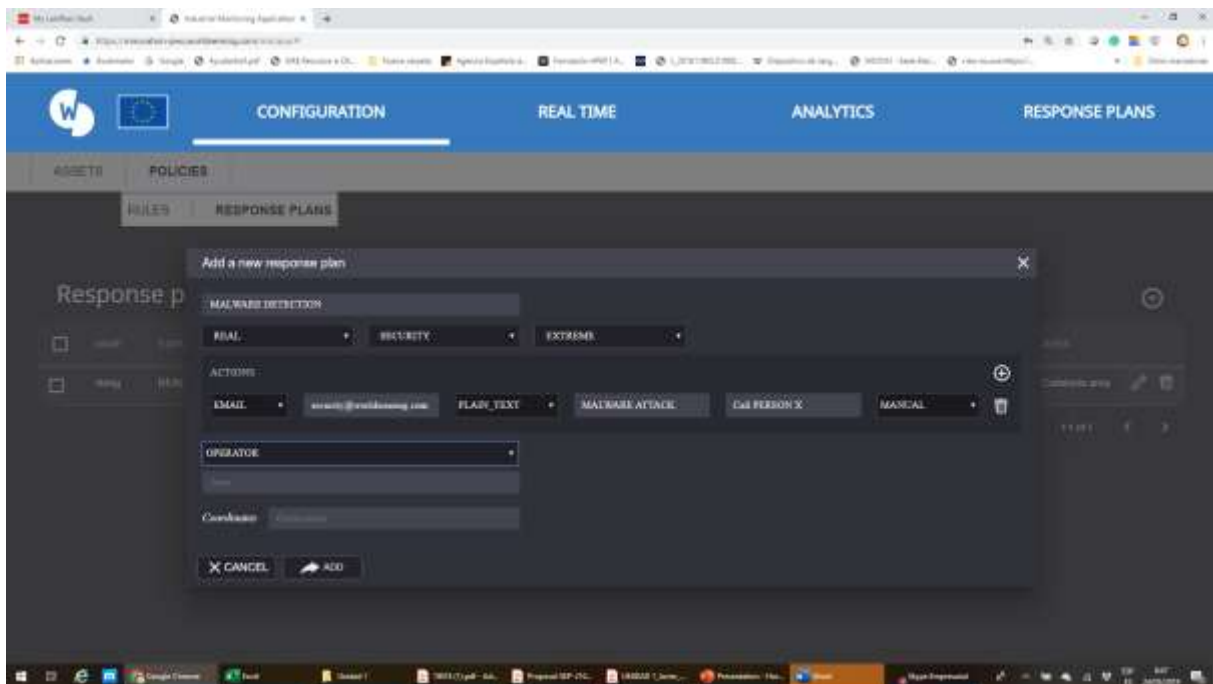
<b>BR_03_REC: Rise awareness and recommend measures</b>	
Objective:	Capture all the information from the attacks that the honeypot is collecting to provide to the end user with details from the attack and actions to mitigate it.
Partners involved:	Forth, Atos, FHNW
Schema:	Honeypot → XL-SIEM → SMESEC Hub → Front End
Input:	An attack that has reached the honeypot is sent to the Hub
Processing:	From a prestored attacks database, there will be a search launched to gather all the data available regarding the attack that has being detected by the honeypot to transfer this information to the user
Output:	The information retrieved from the database containing best practices or actions to perform on the solutions iframe is sent to the front end as an advice for the user

**Table 14: Rule example - rise awareness**

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture	<b>Page:</b>	39 of 68				
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final

As far as the SMESEC business rules configurations are concerned, some of them will be hardcoded in the system to thus provided a minimum level of functionalities after considering the security solutions deployed in each SME environment. This initial set-up is needed once the framework is initially adopted. Nevertheless, more advanced and ad-hoc business rules are expected to be added through a user-friendly interface by SMESEC end-users to respond to a specific alert or event when it is detected. The idea is to provide the capability to design response plans against cybersecurity attacks which are fully aligned with the risk appetite of the company. For example, in the figure below, a simple action is proposed to a platform operator once a malware attack is detected by the framework: sending an email to the security manager.

For the sake of clarity, it must be pointed out that this second mechanism for business rules configuration is still in its infancy and the first functional proofs are just starting.



**Figure 10: Rule configuration**

After the Hub core stage, the output is sent to the SMESEC framework front end so that it can be further processed. The selected methodology to consume these data, both from the perspective of the visualization but also for historical management will be done through an API service.

In this approach, the HUB provides a direct response to some of the initial requirements that the SMESEC framework should fulfil:

1. **Correlation of alerts and tools:** the SMESEC Hub correlates inputs from different solutions to provide more advanced functionalities than those offered by a single one (i.e. BR3);
2. **Response:** business rules are envisaged to suggest a response plan adapted to the specific user needs.
3. **Forensics:** the SMESEC Hub stores the historical events in a dedicated database so that they can be used at any time to conduct a forensics exercise.

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	40 of 68
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1
				<b>Status:</b>	Final



## 4.2 Interface to other components

Specific details of each one of the inputs and outputs elements of the SMESEC Hub are described below.

### 4.2.1 Input

The selected entity to gather information from the tools is a queues system. In this point, solution providers will periodically post the data coming from their deployments and then the SMESEC Hub automatically captures them. These alert data will be placed in the queue by means of a JSON file, whose format has been standardized to unify the communication methodology. Actually, a tailor-made MISP message is proposed for all the solutions to be connected to the Hub (Table 15).

Field	Location custom MISP
Source of the data (e.g. Citrix Firewall, Forth Honeypot, FHNW CYSEC)	Event → Attribute → PluginID and PluginSID
Timestamp	Event → Date
Attacker (IP, port, host name, ...), if applicable	Event → Attribute → Source IP
Attack recipient (IP, port, host name, ...), if applicable	Event → Attribute → Destination IP and Port
Severity/reliability/risk numeric indicator	Event → Attribute → Risk value
Additional info (e.g. for CPU usage business rule, the list of processes running in the machine and their corresponding %CPU)	Event → Attribute → User data

**Table 15: Alert JSON format to be used for reporting alerts to the SMESEC Hub.**

A more detailed description of the JSON format and a practical example is shown in Annex A. Detailed level description of the Input JSON Format

### 4.2.2 Output

Refined results from the processing of the business rules are available for consulting from an API. The two expected consumers are the front-end and any service working with historical data. Figure 11 shows a generic alert and recommendation action to mitigate the potential attack is shown:

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	41 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final

```
{  
  "title": "Forbidden network authentication detected",  
  "description": "Attack type [DDoS], recommendations: [Close ports 5555 and 6666]",  
  "source": "Forth Honeypot",  
  "timestamp": "2019-06-16 14:13:00"  
}
```

**Figure 11: Alert and recommendation example**

### 4.3 Design rationale

The entire architecture of the Hub has been envisaged to provide an output useful for any user, regardless of the technical knowledge. To accomplish this goal, the system capacity to raise alarms, suggesting corrective actions becomes crucial. The unified system for data ingestion and the modular and agnostic concept of the core module is also crucial to tailor the system to the specific SME's needs as well as to extend its use beyond the SMESEC framework. The SMESEC Hub has been conceived as a modular solution in which more functional extensions can be easily added in the future with a two-sided approach: (i) provide advanced functionalities to cybersecurity experts and (ii) enrich the non-technical actions capabilities so that SMEs are more willing to use it. In short, the SMESEC Hub is just the first step in a continuous improvement strategy to approach cybersecurity to different users.

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	42 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final

## 5 Initial version of the SMESEC Framework prototype

### 5.1 Description and objectives

This section describes the prototype of the SMESEC Framework implemented at M24. The first implemented version, which was achieved at M12 was only an initial one where all tools were accessible in a unified portal. This was complemented by a very initial version of the authentication module.

This updated version of the SMESEC Framework has been improved from the last one focusing in the dashboard and internal components. We followed the refined version of the architecture in order to create better and improved communication, storage of information, data processing, security, etc. Also, we created a new dashboard as the initial entry point of the SMESEC Framework where we show quick-access information about the cybersecurity status of the system. This was possible thanks to all the information compiled from the tools and the extremely useful feedback of the use cases. After checking the initial version they highlighted how the first thing they wanted for access was “how is my system” and not a long list of tools that they have to directly access for information.

Additionally, we worked in the development and refinement of internal components that provide storage, authentication, etc. The authentication system was integrated in all tools, the framework, the training platform, etc. following the list of roles identified previously.

Look & feel and user-experience is very important for us. SMESEC aims to provide a specialized and unified cybersecurity solution for SMEs. Therefore, and bearing in mind the low-level expertise of most of the employees of these organizations, we had to go through many iterations for refining the usability of the SMESEC Framework. Also, it was important to provide the information in the easier and more accessible way.

Finally, we are working in providing a third-party API for external providers of cybersecurity solutions so they can integrate their solutions into our framework, making it a “cybersecurity market” where SMEs can promote their applications, do business and take advantage of the information compiled from the tools for creating plugins.

### 5.2 Functionalities and characteristics

The SMESEC Framework offers a series of functionalities to the users. There are on the one hand visualization of results and on the other hand access to tools and specific functionalities.

When the user accesses the system, the initial interface shown depends of her role. If the role is admin then she will be redirected to “Security Status Overview” and if it is a normal user then “My cybersecurity status”. The idea behind this is that the admin of the system needs access to the general

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	43 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final

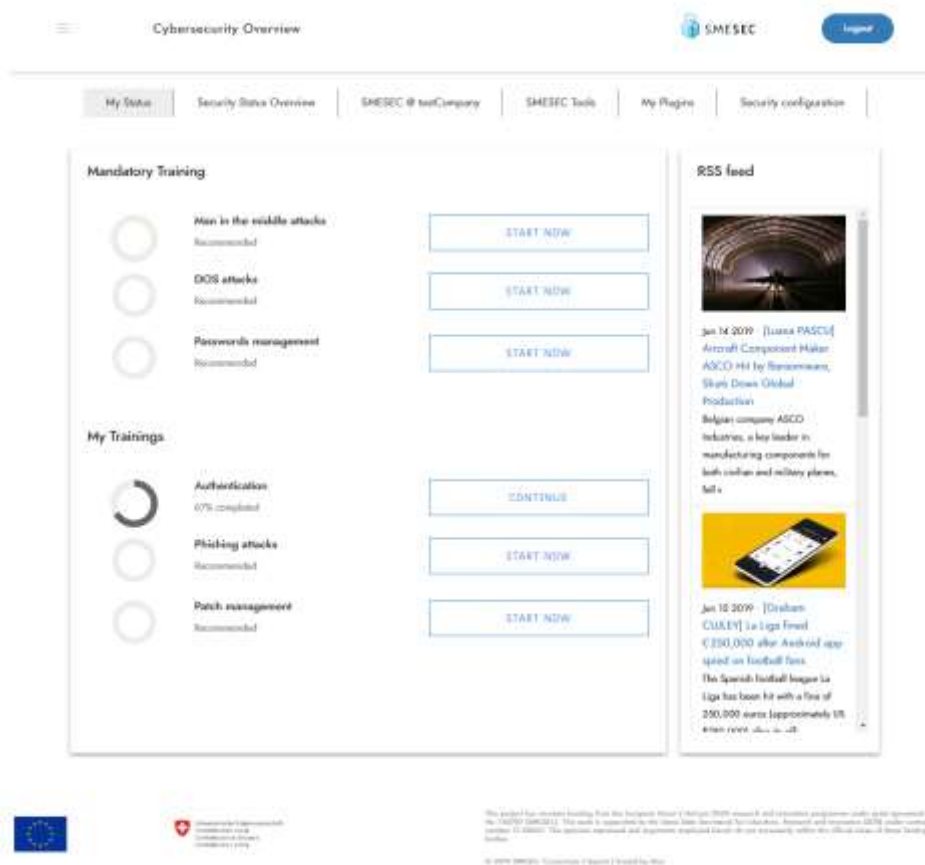
cybersecurity overview of the whole organization (alarms, courses done by all, recommendations of solutions to add, etc.) while if it is a normal user then she only needs to see information about its training, awareness, news of cybersecurity, etc. Bearing in mind each type of role could have different needs we think it is important to show only the necessary information to each person, avoiding showing complex or unnecessary information. This is another way to help SMEs improving their cybersecurity needs. Each role can access only the specific information they need. Apart from these we have more roles, as defined previously. Each role has access to the following tabs:

- Normal user: “My Status”
- Admin: “My Status”, “Security Status Overview”, “SMESEC@”, “SMESEC Tools”, “My Plugins”, “Security Configuration”
- Security Analyst: “My Status”, “Security Status Overview”, “SMESEC@” and “My Plugins”
- Auditor: “My Status”, “Security Status Overview”, “SMESEC@”, “SMESEC Tools”, “My Plugins”, “Security Configuration” (only read)

Regarding each tab, following we present them with a description of their functionality and goal. The tabs are “My Status”, “Security Status Overview”, “SMESEC@”, “SMESEC Tools”, “My Plugins”, “Security Configuration” and the list of tools to be accessed directly.

### 5.2.1 My Status

Figure 12 shows the current version of the “My Status” tab.



<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	44 of 68
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1
		<b>Status:</b>	Final		

**Figure 12: My Status view**

This view focuses in providing information of the cybersecurity status of the employee. Being this a personal space, it shows only information about the training and the results obtained. This way the user can grow its cybersecurity knowledge (which covers not only understanding some basics of cybersecurity but also how to better protect herself and the company). This panel will be later expanded to show also information about cybersecurity news, alerts, etc. coming from the SMESEC website. The training information comes from the training platform of SMESEC.

### 5.2.2 Security Status Overview

Figure 13 shows the current implementation of the “Security Status Overview” tab. This one focuses in presenting an overview of the cybersecurity status of the system.



**Figure 13: Security Status Overview view**

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture	<b>Page:</b>	45 of 68
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU
<b>Version:</b>	1.1	<b>Status:</b>	Final

As we can see the first diagrams shown are about the status of the system. The first one, on the left, shows a high-level analysis of the status of the system. This number is calculated by means of all the events received by the monitoring tools and with an internal classification we are working on right now. The idea is that this number can be adapted automatically depending on the level of cybersecurity of other SMEs or according to the vulnerabilities of the system. This way we aim to show a good way for non-cybersecurity experts to understand how good or bad they are doing in terms of cybersecurity. The more secure the system is the better number they will have.

The second graph shows the different attacks the system has been receiving and information about them. Since attacks are critical for organizations we think it was the best approach to show this information in a short way and an identifier of the risk level according to our expertise, so they would know in which attacks to focus first.

The next one, “Last Security Events”, shows the different events in a calendar with their level of risk and total in the last day, week and month. The idea with this is for the admin to have a good overview of what is happening in the system in a long view, so to know the evolution of the actions they are doing in their organization from a cybersecurity point of view (both technical and human such as training).

Finally, the last part of the overview for admins of the system is recommendations about how to improve the security of the organization. In here it shows the cybersecurity level of the organization (using information of cybersecurity level of all employees), recommended tools and training. The idea of this sub-section is to give the admins general information of how all the employees are improving their cybersecurity knowledge and what tools could benefit them given their needs and business. These recommendations will also use information from the awareness tool (CySec), which will compile information about the existing tools of the organization and the needs they have. This way the recommendations will be updated automatically every time courses or new cybersecurity tools are used in the system.

### 5.2.3 “SMESEC@”

This tab shows more specific information of each tool used in the system. Figure 14 shows current implementation of the dashboard. The idea is to show more data than the overview one but not as specific as the specific portal of each tool. In the example shown here we can see a quick overview of the XL-SIEM, GravityZone and EWIS tools (monitoring information) and CySec (training of the organizations). This information is dynamic so depending on the tools installed it will be more or less graphs.

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	46 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final



Figure 14: SMESEC@ view

## 5.2.4 “SMESEC Tools”

This tab shows the status of each tool in the system, access to clients and agents for installing in the target system and access to the documentation for installing, configuring and using each tool. Figure 15 shows a short overview for three of the tools running in the current version.

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	47 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final

Cybersecurity Overview SMESEC [Logout](#)

---

[My Status](#) | 
 [Security Status Overview](#) | 
 [SMESEC @ testCompany](#) | 
 [SMESEC Tools](#) | 
 [My Plugins](#) | 
 [Security configuration](#)

**Atos XL-SIEM**

Download material

Go to status

**FORTH EWIS**

Download material

Go to status

**BitDefender GravityZone**

Download material

Go to status

**CITRIX NetScaler**

Go to status

**Atos Risk Assessment Engine**

Go to status

**FHNW CySec**

Go to status

**Training Platform**

Go to status

**IBM Anti-ROP**

Download material

Go to status

**IBM Angel Eye**

Go to status

**EGM TaaS**

Go to status

**Figure 15: SMESEC Tools view**

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture	<b>Page:</b>	48 of 68
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU
	<b>Version:</b>	1.1	<b>Status:</b> Final



All the tools offered in the SMESEC Framework are found here. Depending on the subscription of the client some of them will be available and others not. We also show information regarding the status of each tool in the system, using icons, such as “installed”, “need update”, “not available”, etc. We show a short description of all the different possibilities in Figure 15.

Regarding the options for each tool, the material for downloading will include the agents (in the case of the XL-SIEM, etc.) or the clients (in the case of the NetScaler, etc.). This allow us to update them easily when a new version is released and inform/allow the users to download and install these new versions.

The material is provided in the training platform. It includes information about how to install, configure and run the clients together with an overview of how to understand the information provided by the tool in each tool-specific dashboard. The material can be not only text but also videos or instructions. That is why we thought it is better to have everything in a single place (so it is always easier to find).

Finally, third-party applications (external to SMESEC consortium) will also be found here. They will follow the same approach for the clients, documentation, etc. This way all tools would be accessible here and allow a previsualization of their functionality for users to check before deciding to install them or not. This dashboard will be worked more in-depth in the next stage for linking with the subscription process and semi-automatic deployment.

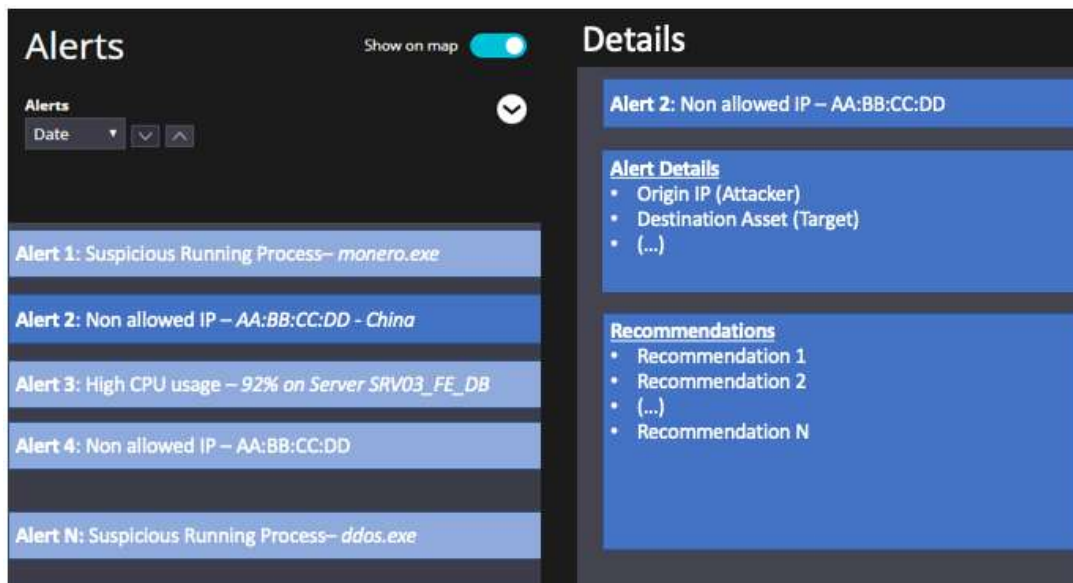
### 5.2.5 “My Plugins”

SMESEC Plugins are special functionalities that take advantage of the data provided by the tool owners via their internal APIs. The main objective of the plugins is to allow either SMESEC partners or external providers to develop extra elements that can support an SME in a specific need, not covered by the cybersecurity solutions of SMESEC. They can be understood as configurable processes that users will have access to.

The plugins are developed and integrated as individual elements and the plan is that users would be able to decide to use them selecting from a list of them. We will provide more information about this functionality in the next version of the SMESEC Framework.

This tab shows the different plugins implemented in SMESEC. Plugins are specific functionalities done by users in order to take advantage of the information compiled in SMESEC by the different monitoring tools. This way we offer a way to have “something more” than just a list of tools available in a unified framework and create functionalities that go beyond the current work. So far we have implemented one plugin and plan to develop more using the data of the tools. This can also benefit from external tools as they could bring additional functionalities and using the data of some tools provide very useful information for area-specific SMEs. A first version of the interface for the existing plugin is shown in Figure 16.

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	49 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final



**Figure 16: Plugin of alerts view**

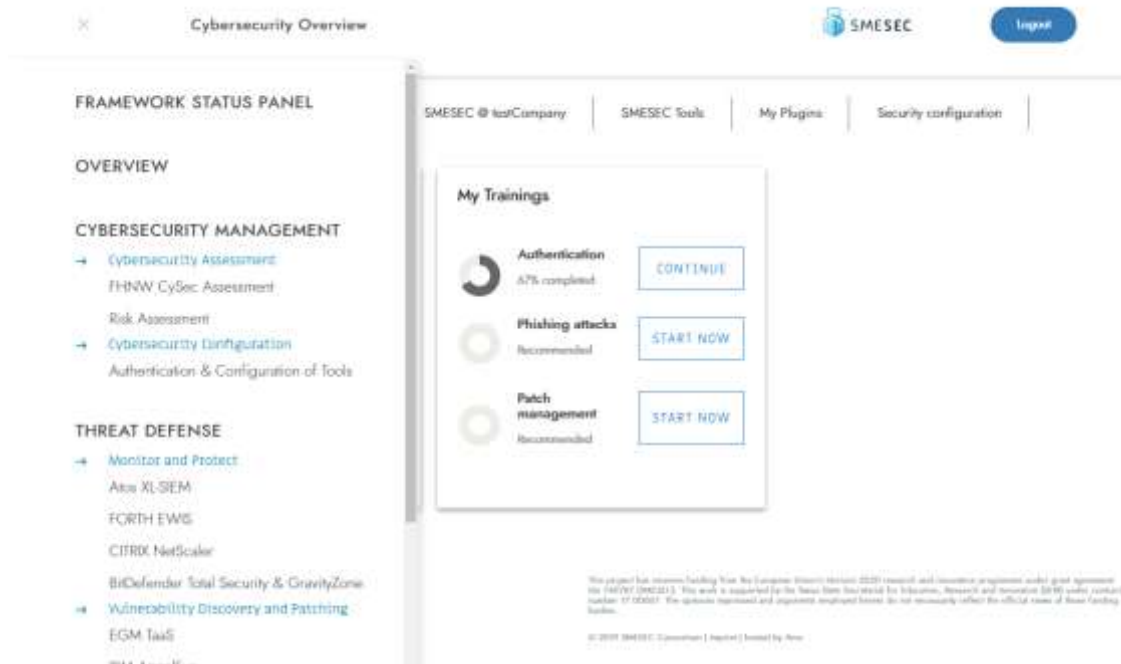
### 5.2.6 “Security Configuration”

We are currently working in this tab. It will contain configuration information of general aspects of the SMESEC Framework and other aspects (general) of tools of SMESEC. Still, as each tool is a product, the configuration of each one is integrated in its system. Therefore, we tried to extract general options for each one and make it available here.

### 5.2.7 Access to Tools

As a difference from the first year, the access to the SMESEC tools is done via a quick link in the SMESEC Framework. Figure 17 shows the link to the tools.

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	50 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final



**Figure 17: Quick link to tools**

## 5.3 Description of testing platform in SMESEC

The testing platform consists of two separate tools that serve the same practical purpose from different perspectives – given source code, the following tools implement various testing strategies to discover bugs and security vulnerabilities in the given source code,

### 5.3.1 FuzzMon

JavaScript is one of the most popular programming languages in the world. Only in GitHub there are more than 2.3 million unique projects written in JavaScript. Amongst other applications, JavaScript is used for server-side development using the NodeJS infrastructure.

Nowadays, there are some static analysis tools that alert developers of bad, or potentially vulnerable places in their code. These tools mainly focus on better coding style, and trivial best practices for secure development. Currently, there is no solution for more complex vulnerability analysis of server-side JavaScript code. This is an important issue to address.

FuzzMon is a pluggable fuzzing infrastructure prototype that uplifts a state-of-the-art fuzzing paradigm (AFL) into a high-level, interpreted, dynamic, and weakly typed language (JavaScript), with focus on server-side JavaScript code (NodeJS) to detect vulnerabilities, find attack vectors and exploits, and discover various bugs, ranging from simple OS command injections to more complex logical errors that can lead to dangerous exploits.

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	51 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final

### 5.3.2 ExpliSAT

ExpliSAT is a powerful software verification tool based on methods of symbolic interpretation. Given source code files, ExpliSAT builds a model of each explicit control flow path with symbolic interpretation of each instruction. Then, it determines whether there exists an instantiation of a path that violates correctness assertions by invoking a decision procedure (SAT). Solving the symbolic formula not only gives an indication of error but generates a set of concrete input values that trigger that error.

## 5.4 Development and integration environment

---

For the development of the SMESEC Framework, it has been decided to use the Java programming language as main technology, because of its flexibility and level of support.

The SMESEC Framework uses as a core Spring Boot, an open source framework sponsored by Pivotal [11].

For the visualization part, the Thymeleaf template engine [12] is used, in conjunction with open source CSS and Javascript frameworks, such as jQuery, Chart.js and Bootstrap [13].

In order to support the development, a continuous integration environment was deployed. This environment is composed of a continuous integration server, using Jenkins [14]. This server automates the necessary tasks to compile the code, perform the tests, analyze code for bugs and possible vulnerabilities in third-party dependencies (described below), creating the docker image and deploy it to a container, so a test instance is always up and running with the latest changes ready to perform integration tests.

More information about the integration environment can be found below, under section 5.5.

For performing these tasks described above, we use Maven [15] as build system, known for its stability and available plugins for extending the functionality.

Besides this infrastructure, a Nexus Repository Server [16] is deployed to store the different snapshots and versions for both the SMESEC Framework compiled code and the Docker images used for deploying it.

## 5.5 Integration methodology

---

The integration of the different tools composing SMESEC in the SMESEC Framework have been done in two different ways, depending on the existing capabilities of each tools.

For the XL-SIEM, GravityZone, and EWIS, the tool's own dashboards are showed in the SMESEC Framework as iframes. This is done due to the impossibility to recreate the complete functionality of the tool with API calls. The approach taken here comes with the downside of showing many different tools, each with its own look and feel, in the same website. This has been overtaken updating each tool style, so they adapted to the general SMESEC look and feel.

For the rest of the tools, dashboards have been created from scratch using API calls, and displaying the required information to the user.

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	52 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final

A special case is the “Security Status Overview” dashboard. In this part information coming from all the tools have been combined with the goal of providing intelligent insights to the SMESEC customers. This integration is provided in both the XL-SIEM and the SMESEC HUB. These tools expose an API, from which the data is retrieved and displayed.

## 5.6 Technical infrastructure

To support the development and integration environment, four different virtual machines have been allocated at Atos premises.

These virtual machines provide support for the following functionalities:

- **Authentication.** This contains the Keycloak server, along with a PostgreSQL database and a LDAP server that serves as backend for user storage.
- **Monitoring.** A Zabbix [17] instance in charge of collects data from all the agents deployed in the rest of the servers of the infrastructure. This tool is able to warn about possible problems before they cause an outage of any of the services. The Zabbix server is also configured to monitor the SMESEC tool’s availability.
- **Artifact storage.** A Nexus Repository Server, configured with a Maven repository and a Docker registry. This server is in charge of storing a copy of the jar file containing the SMESEC Framework code, along with the Docker image used as a base for the running container for each version.
- **Continuous integration.** The CI server is composed of a Jenkins instance, a Sonarqube instance, and a Docker CE installation, that serve to continuously test, build, analyse and deploy the code of the SMESEC Framework.

The technical description of the hardware used for supporting the infrastructure can be found below:

SERVER	CONTENT	vCPU	RAM (GB)	Disk (GB)	OS
<b>Authentication server</b>	Keycloak, PostgreSQL, LDAP Server	2	16	70	CentOS7
<b>Monitoring</b>	Zabbix, OpenVAS	4	8	30	CentOS7
<b>Artifact storage</b>	Nexus Repository	2	8	200	CentOS7
<b>Continuous integration</b>	Jenkins, Sonarqube, Docker	2	32	100	CentOS7

## 5.7 Authentication and security

The authentication of the SMESEC Framework is provided by Keycloak [3], using the OpenID protocol [18] for both authentication and authorization.

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	53 of 68
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1
				<b>Status:</b>	Final

For each request, the access token of the user is checked against the Keycloak server for its validity. It also checks if the user has the necessary permissions to perform the request. For these actions, we use the official Spring Boot adapter [19], provided by Keycloak. The roles we defined for accessing the SMESEC Framework are the ones defined in the previous deliverable. Also, what can be accessed in the SMESEC Framework is described in the previous section.

The Keycloak Spring Boot adapter uses the Spring Security [20] framework under the hood, which is also used to assist in the protection against XSS or CSRF attacks.

Nevertheless, the protection against XSS attacks that Spring Security provides relies in the browser capability to understand the X-XSS-Protection header [21], so every input of the SMESEC Framework needs to be sanitized. At this moment this is not implemented, since the SMESEC framework does not expect any user input. In order to prepare for providing this security measure, the SMESEC Framework is making use of the OWASP HTML Sanitizer Project [22], which is already configured and ready to use.

Also, Content Security Policy [23] is planned to be implemented so only trusted sources are allowed to execute scripts in the SMESEC Framework. This security measure will help us preventing clickjacking attacks.

Also, to ensure that the code of the SMESEC Framework is free of vulnerabilities, we run static code analysis with Sonarqube [24], using the FindBugs Security Audit [25] profile. Besides this analysis, and given that we are using many third-party dependencies, OWASP Dependency Checker [26] is being used to analyze possible vulnerabilities in the dependencies used, so we are able to upgrade those dependencies as soon as possible.

Finally, we plan to have a red-team (thanks to the open call of the project) for checking the resilience and security of the framework. The idea is that they perform several exercises and in each iteration, give us feedback for improving the system. Having different tools in a unified framework means the communication and data storage is critical so this will be one of the main points of action.

## 5.8 Deployment and configuration

### 5.8.1 Deployment and configuration of the SMESEC Framework and client-side applications

The deployment of the SMESEC Framework requires the installation and configuration of its sub-components, which are shown in the diagram of the architecture. There exist two categories of deployment that we describe here: the server of the SMESEC Framework and the clients of the tools. Each of these deployment have different technologies, roles, and usage.

Regarding the server, as we have presented in the architecture, the following items are part of the SMESEC Framework “package”:

- Back-end: contains the communication network, data storage, APIs, etc.
- Dashboard: shows the interface of the SMESEC Framework, which was presented in the previous section
- Supporting components: includes the authentication component (and login), external API, etc.

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	54 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final

- Training platform: the training platform of SMESEC, including existing and available courses created in the project
- SMESEC tools: this is an optional package depending if the tools are used as a service or on premises. On premises means that the organization deploying the framework, and which is going to offer it, provides servers with their own tools instead of being accessed in the servers of the tool providers

For the clients it is only required to install the client of each tool in the target system. This client can be an agent (e.g. XL-SIEM) or instance (e.g. GravityZone). They run in the system to be protected and are configured independently as they are application-specific.

Regarding the role that install each of the types of SMESEC we have identified two different sets, each of them with a different functionality and need:

- SMESEC clients: done by end-users (SMEs or SME associations)
- SMESEC Framework package: done by large organizations, SMEs or SME associations

Following we present for each type more information of the deployment process.

### *SMESEC Clients*

According to the selected tools, end-users will download and install their corresponding clients or agents. The process for installing each of the clients are described in the SMESEC Tools site. The information comes in the form of documentation, videos, examples, etc. Also, there users can find information for the configuration of the tools and how to adapt them to their own system.

### *SMESEC Framework Package*

The users download the package, which includes all the elements previously described. We plan to provide a docker version of the package in order to facilitate its installation. This package will automatize the process as much as possible but still, and as we understand it could be complex to deploy all these elements, we will include material for its correct installation. This would also include information about technical requirements, technologies, communication, security, etc. For example, one of the main activities to perform would be the configuration of Keycloak in their system.

Regarding the tools, as we described before, there are two possibilities: to deploy the tools in their own system or to use them as-a-service. Each of them has pros and cons. Having the tools in their own system means they have better and faster access to them but this would increase the cost of the system (together with the maintenance). For installing the tools in their system they would need to follow, for each tool, the instructions for their deployment and configuration. For example, the way to deploy and configure the XL-SIEM and NetScaler are very different. If they want to use the tools as-a-service then they have to configure the communication between the framework and the tools following the guidelines of the tool owners.

## 5.8.2 Updating

We identified two different cases for updating: the update of the SMESEC Framework or the update of the SMESEC tools. As we think this is a critical process we have designed an initial methodology about this functionality.

### *SMESEC Framework*

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	55 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final

When a new version of the SMESEC Framework is released all the customers that are providing it are notified. The new update will be released as a package (e.g. docker), which will bring the new updates and information (documentation if necessary) about this new version and changelog so it is easier to understand the changes. The organization would then deploy the new version in their system as a whole. The data storage should remain the same except if it is necessary a change of the data model used. In this case we plan to include a functionality for exporting/importing the new information for different versions of the SMESEC Framework. This way, the sub-components of the SMESEC Framework will be divided into micro-services and only the relevant ones will be updated. This will allow us to not lose important information such as alerts, configuration, etc.

The SMESEC website will play an important role here, as will notify always about the last stable version of the SMESEC Framework released so users can always check if they are using the last one and the improvements it brings.

The abovementioned process is specific for the version deployed on-premises. For the cloud-based (as-a-service) one the updating is transparent to the users. Also, we plan to use a blue-green update process for this last one, setting up two instances of the infrastructure and gradually re-direct users to the new versions.

### ***SMESEC Tools***

The process we plan to follow for updating tools is as follow: when a new version of a tool is released owners of the SMESEC Framework receive a notification. If the tool they are using is accessed as a service then they would need to update the configuration for connecting with the new update service. Due to the criticality of this situation we are working in providing a synchronization of all tools for this process (e.g. having a common package for the clients with the last versions of each one).

If the tools are managed on premises, then the admins would deploy the new version internally following the instructions of the developers. After this they would finish the new configuration and provide the updated clients to the users. If a vulnerability is discovered in a tool, while a new version is released, we make possible to deactivate its use in the framework, so it doesn't affect the whole system.

### ***SME-clients***

The SMEs interact with the SMESEC Framework by installing the clients/agents of the tools. When these applications are updated they users have to download or update them. Due to the criticality of this process, we plan to follow onewith unique information of all the clients for the end-users so they can automatically update to the last versions in the more easier and transparent way.

## **5.9 API for external tools**

The external data API is intended for including data and functionality coming from tools that are not part of the initial version of the SMESEC Framework. We thought it could benefit greatly SMESEC as a mean for providing valuable information and extra cybersecurity functionalities that are not covered or supported with the core tools of SMESEC.

This API will act as a bridge for normalizing the information coming from those external tools into the format that the SMESEC HUB can comprehend and process along with the data provided by SMESEC tools. A high-level picture of the flow is provided below:

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	56 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final





**Figure 18: Data flow**

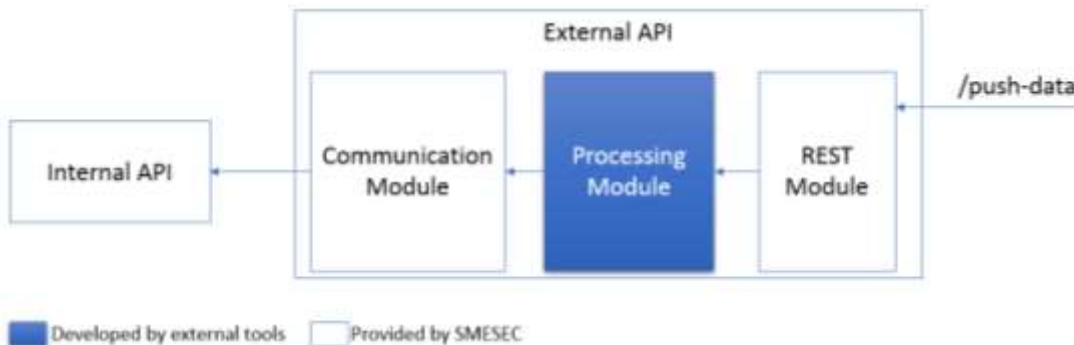
To be able to normalize the data, an interface of this API must be implemented by the external tool, providing a set of different data that will enhance the framework with more insights.

Because of the implementation of the external API components will be made by different organizations that are not part of the SMESEC consortium, and thus are not aware of the architecture and the internal components of the framework, we need to provide guidelines and code support with examples. We plan to cover this by including a couple of third-party applications, so we can refine the process and data management.

Although this guidance is not yet defined and can vary in the future, it is planned to provide Java interfaces to the users with methods that needs to be implemented so we are able to transform the data these tools produce to the internal SMESEC format.

Together with this we will provide supporting methods such as communication to send transformed data to SMESEC data API and entry points to the external API, so external partners can know where to send the data. This can be provided as a REST API, with known endpoints so they can send data to the framework. Other solutions, as queues or messaging systems, could replace the REST approach.

A draft of this architecture is shown in the picture below:



**Figure 19: External tools API component diagram**

Other options, that can either substitute Java interfaces or provided alongside them, are Python abstract classes or C# interfaces, providing a wider set of technologies that can be used by different external tools. This would allow to have a long list of possibilities for external tools to be integrated, aiming to cover as much technology as possible.

This way, the integration of external tools, being as beneficial as it is for SMESEC, needs to be supported at architecture and framework level. The benefits it brings are, among others:

- Provide extra functionalities not supported by the core of SMESEC. This allows for extending in the future when new threats may appear or take advantage of new technologies, making SMESEC a living platform that can adapt to the dynamic needs of organizations and technologies

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture	<b>Page:</b>	57 of 68	
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	
	<b>Version:</b>	1.1	<b>Status:</b>	Final

- Allow for more extra data to be available in the SMESEC Hub. This would allow the creation of more plugins that have access to more information. The more information of solutions working, the better service for the SMEs
- Create new business opportunities. By allowing external organizations of SMESEC to participate and integrate their solutions we could transform our framework into a marketplace, where cybersecurity solution providers could sell their work

Additionally, one of the categories of the open call is the integration of tools by means of this third-party API so we plan to have a good feedback and contribution in the next phase of the project that will help to extend the integration and validation of this component.

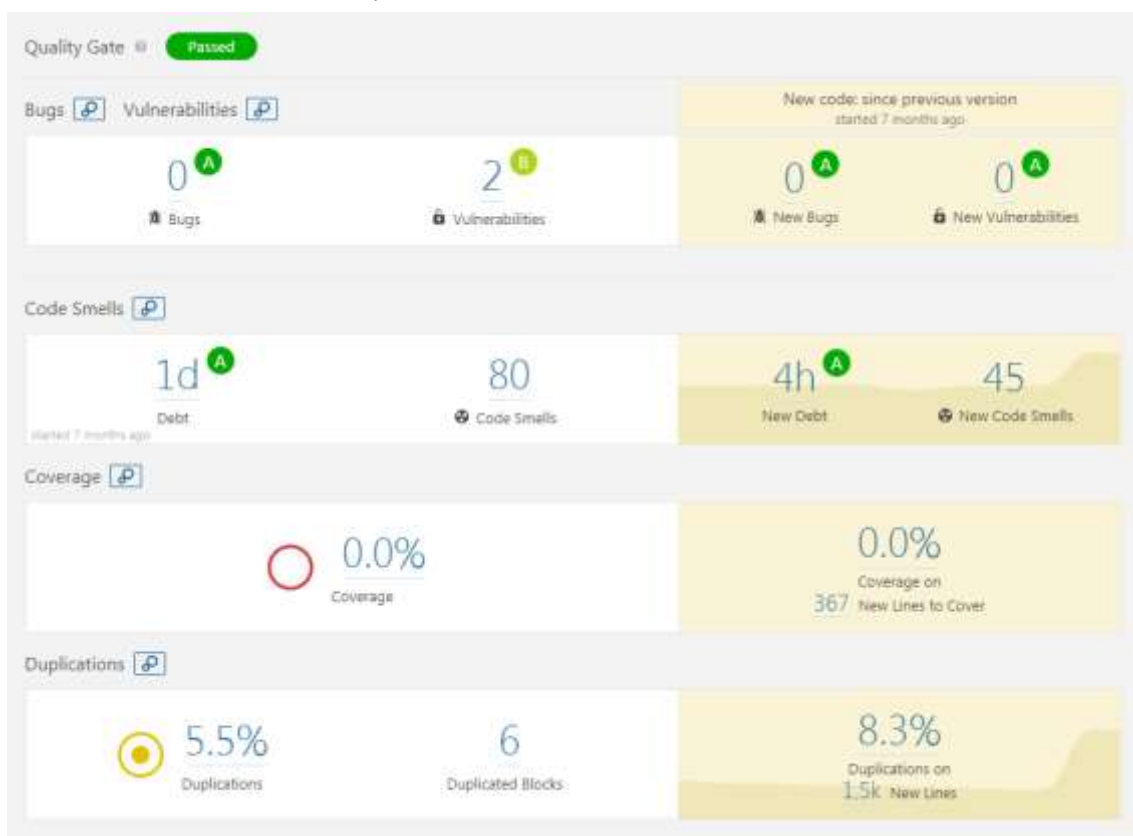
## 5.10 Initial testing

As the framework is just a placeholder for the functionality provided by SMESEC, unit testing has not been performed, since is not necessary at this stage. This is due to our internal planning of focusing on extending the framework and its capabilities in the next iteration, focusing this one in the final integration of the tools at all levels (e.g. functionality, authentication, data provided, etc.).

Most of the code of the SMESEC Framework is made on the controller layer, with basic logic to return the required page.

On the other hand, as it is stated above in section 5.6, static and dynamic analysis is performed in the code, with help of open source tools as Sonarqube or ZAP Proxy.

Below are the results of these analysis.












<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	58 of 68
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1
				<b>Status:</b>	Final

The two vulnerabilities discovered are referred to the current version of Keycloak, and are planned to be removed during an upgrade of the server.

This upgrade is not critical, since the vulnerabilities cannot be exploited in our environment:

- CVE-2019-3868: With a CVSS score of 3.8, the attacker must have access to the server instance to perform the attack, resulting in steal of browser’s session.
- CVE-2019-3875: With a CVSS score of 4.8, the attack that exploits this vulnerability needs to be performed on unsecured protocols, that are not allowed in SMESEC.

As of the results of the ZAP analysis, there are some security issues, as stated in the results shown below:

- 
- ▶  CSP Scanner: Wildcard Directive (2)
  - ▶  Multiple X-Frame-Options Header Entries (2)
  - ▶  X-Frame-Options Header Not Set
  - ▶  Absence of Anti-CSRF Tokens (2)
  - ▶  Cookie No HttpOnly Flag (5)
  - ▶  Cookie Without Secure Flag (2)
  - ▶  Incomplete or No Cache-control and Pragma HTTP Header Set (31)
  - ▶  Web Browser XSS Protection Not Enabled
  - ▶  X-Content-Type-Options Header Missing (14)

There are no high-risk vulnerabilities present in the framework, and a more in-depth analysis of the report shows that most of the issues discovered are present in requests to the Keycloak instance, so we expect that those problems will be mitigated after the upgrade and the final configuration for the production environment during the third year.

Other issues, such as the ones referred by “No Cache-control” are false positives, as they are referencing the CSS classes that we want to be cached for performance.

During the third year, further testing will be performed to ensure that no vulnerabilities are found in the frontend of the application. These tests will use the before-mentioned open source tool ZAP Proxy.

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	59 of 68	
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b> Final

## 6 Conclusions

We have developed the final architecture based on the system design and initial architecture proposed earlier in “SMESEC Unified Architecture – First Internal Release” (D3.2) [2] and “SMESEC System Design” (D3.1)[1].

We have finalized the architecture description of design views including: composition view, component view and interface view. We developed a new version of user-interface view with higher focus on user experience.

We have described, in depth, the architecture of internal components that are the core of the SMESEC Framework. We explain how the SMESEC Hub can collect alerts and information from various tools, how the extensions can correlate and orchestrate between those alerts and produce high quality attack indication, and how we provide response and forensics capabilities. Further, we describe the detailed requirements of the SMESEC communication bus.

We have developed the SMESEC prototype and described its: objectives, functionalities, integration environment, integration methodology, infrastructure, authentication mechanism, communication model, deployment, and configuration. Further, we describe the initial testing that we have conducted for this prototype.

This document will serve as basis for further development of SMESEC Framework, and results will be reported in “SMESEC security Framework Final version” (D3.7) public document.

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	60 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final

## 7 References

- [1] SMESEC deliverable 3.1 “SMESEC System Design”, Fady Coptly, 2018.
- [2] SMESEC deliverable 3.2 “SMESEC Unified Architecture – First Internal Release”, Fady Coptly, 2018.
- [3] Keycloak, <https://www.keycloak.org/>, 2019/04/06
- [4] Firefox, <https://www.mozilla.org/en-US/firefox/>, 2019/04/06
- [5] Chrome, <https://www.google.com/chrome/>, 2019/04/06
- [6] Windows, <https://www.microsoft.com/en-us/windows>, 2019/04/06
- [7] Safari, <https://www.apple.com/lae/safari/>, 2019/04/06
- [8] MacOS, <https://www.apple.com/macOS/>, 2019/04/06
- [9] SMESEC Grant Agreement no. 740787 – Annex I Description of the Action (Part B), April 2017.
- [10] MISP data models – MISP core format, MISP taxonomies, <https://www.misp-project.org/datamodels>, 2018/12/10
- [11] Pivotal. Last visited: June 15<sup>th</sup> 2019. Link: <https://pivotal.io/>
- [12] Thymeleaf. Last visited: June 15<sup>th</sup> 2019. Link: <https://www.thymeleaf.org/>
- [13] Bootstrap. Last visited: June 15<sup>th</sup> 2019. Link: <https://getbootstrap.com/>
- [14] Jenkins. Last visited: June 15<sup>th</sup> 2019. Link: <https://jenkins.io/>
- [15] Maven. Last visited: June 15<sup>th</sup> 2019. Link: <https://maven.apache.org/>
- [16] Nexus Repository. Last visited: June 15<sup>th</sup> 2019. Link: <https://www.sonatype.com/nexus-repository-sonatype>
- [17] Zabbix. Last visited: June 15<sup>th</sup> 2019. Link: <https://www.zabbix.com/>
- [18] OpenID. Last visited: June 15<sup>th</sup> 2019. Link: <https://openid.net/>
- [19] Spring-Boot. Last visited: June 15<sup>th</sup> 2019. Link: <https://spring.io/projects/spring-boot>
- [20] Spring-Security. Last visited: June 15<sup>th</sup> 2019. Link: <https://spring.io/projects/spring-security>
- [21] XSS header protection. Last visited: June 15<sup>th</sup> 2019. Link: <https://docs.spring.io/spring-security/site/docs/5.0.x/reference/html/headers.html#headers-xss-protection>
- [22] Java sanitizer. Last visited: June 15<sup>th</sup> 2019. Link: [https://www.owasp.org/index.php/OWASP\\_Java\\_HTML\\_Sanitizer\\_Project](https://www.owasp.org/index.php/OWASP_Java_HTML_Sanitizer_Project)

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	61 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final

- [23] Content security policy. Last visited: June 15<sup>th</sup> 2019. Link: <https://content-security-policy.com/>
- [24] Sonarqube. Last visited: June 15<sup>th</sup> 2019. Link: <https://www.sonarqube.org/>
- [25] Find security bugs. Last visited: June 15<sup>th</sup> 2019. Link: <https://find-sec-bugs.github.io/>
- [26] OWASP dependency check. Last visited: June 15<sup>th</sup> 2019. Link: [https://www.owasp.org/index.php/OWASP\\_Dependency\\_Check](https://www.owasp.org/index.php/OWASP_Dependency_Check)

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	62 of 68		
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b>	Final

## 8 Annex A. Detailed level description of the Input JSON Format

This section defines the input format of the coming alerts from the security tools to the SMESEC Hub. It is based on the MISP format [10].

DETAILS	IN MISP FILE	FORMAT	VALUES/EXAMPLES
Source of the alert (provider that sent the information to the Hub)	ID = misp[“Event”][“Attribute”][i][“value”]  at <i>i</i> where:  misp[“Event”][“Attribute”][i][“comment”] == “PluginID”  AND  SID = misp[“Event”][“Attribute”][i][“value”]  at <i>i</i> where:  misp[“Event”][“Attribute”][i][“comment”] == “PluginSID”	2 strings containing numeric values corresponding to ID and SID	- Citrix Firewall - “1” - “1” - Process CPU exceeded - “2” - “2” - FHNW Cysec - “3” - “3” - Forth Honeypot - “110000” - “5”
Timestamp	misp[“date”]	Datetime string following the YYYY-MM-DD HH:MM:SS format	E.g: “2019-05-15 14:45:00”
Attacker (IP address)	IP = misp[“Event”][“Attribute”][i][“value”]  at <i>i</i> where  misp[“Event”][“Attribute”][i][“comment”] == “Source IP associated to the detected alarm.”	1 string containing an IP address in v6 format	E.g: “aaaa::1”
Attack recipient (IP address, port)	IP = misp[“Event”][“Attribute”][i][“value”]  at <i>i</i> where  misp[“Event”][“Attribute”][i][“comment”] == “Destination IP associated to the	2 strings containing an IP address in v6 format and a numeric port	E.g: “aaaa::2” and “716”

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	63 of 68
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1
				<b>Status:</b>	Final

	<p>detected alarm.”</p> <p>AND</p> <p>PORT =</p> <p>misp[“Event”][“Attribute”][i][“comment”]</p> <p>at <i>i</i> where</p> <p>misp[“Event”][“Attribute”][i][“comment”]</p> <p>== “Destination Port associated to the detected alarm.”</p>		
(Optional) Geolocation	misp[“Geolocation”]	Dictionary with keys “latitude”, “longitude” and “altitude”, values are strings corresponding to floats	E.g: { "latitude": 46.9412", "longitude": 9.0456", "altitude": "1326.4" }
Severity	misp[“Cap-info”][“severity”]	String	“Extreme”, “Severe”, “Moderate”, “Minor” or “Unknown”
Validity of the alert	misp[“validity”]	String corresponding to the expiry date of alert expressed as UNIX epoch (number of seconds that have elapsed since January 1st 1970)	E.g: “1992638251”
(Optional) Additional info, such as list of processes running in the machine and their corresponding %CPU	<p>misp[“Event”][“Attribute”][i][“value”]</p> <p>at <i>i</i> where</p> <p>misp[“Event”][“Attribute”][i][“comment”]</p> <p>== “Userdata<sub>j</sub>”</p> <p>0 &lt; j &lt; 11</p>	String containing additional info	E.g: “chrome 85\nscp 20\n\n”

**Table 16: Detailed description of MISP format**

From a practical point of view, the JSON looks as follows in the example (alert coming from the XL-SIEM to the SMESEC Hub):

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	64 of 68
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1
				<b>Status:</b>	Final



```

{
  "Event":
  {
    "id": "69123",
    "orgc_id": "11",
    "org_id": "11",
    "date": "2018-09-12",
    "threat_level_id": "4",
    "info": "AAA Probe - Forbidden Network Authentication",
    "published": false,
    "uuid": "5c87a305-4b88-4d2c-ada2-354a0a00020f",
    "attribute_count": "9",
    "analysis": "2",
    "timestamp": "1552392965",
    "distribution": "0",
    "proposal_email_lock": false,
    "locked": false,
    "publish_timestamp": "0",
    "sharing_group_id": "0",
    "disable_correlation": false,
    "extends_uuid": "",
    "event_creator_email": "pablo.barrientoslobato@atos.net",
    "Org":
    {
      {
        "id": "11",
        "name": "Atos-SMESEC",
        "uuid": "5c800639-b840-4870-8b6b-61480a00020f"
      },
      "Orgc":
      {
        {
          "id": "11",
          "name": "Atos-SMESEC",
          "uuid": "5c800639-b840-4870-8b6b-61480a00020f"
        },
        "Attribute": [
          {
            "id": "452847",
            "type": "other",
            "category": "Network activity",
            "to_ids": false,
            "uuid": "5c87a306-ded8-40b8-8463-354a0a00020f",
            "event_id": "69123",
            "distribution": "5",
            "timestamp": "1552392966",
            "comment": "Source IP associated to the detected alarm.",
            "sharing_group_id": "0",
            "deleted": false,
            "disable_correlation": true,
            "object_id": "0",
            "object_relation": null,
            "value": "aaaa::1",
            "Galaxy": [],
            "ShadowAttribute": []
          },
          {
            "id": "452848",
            "type": "target-location",
            "category": "Targeting data",
            "to_ids": false,
            "uuid": "5c87a306-9340-425a-99f7-354a0a00020f",
            "event_id": "69123",
            "distribution": "5",
            "timestamp": "1552392966",
            "comment": "Destination Port associated to the detected alarm.",
            "sharing_group_id": "0",
            "deleted": false,
            "disable_correlation": true,
            "object_id": "0",
  
```

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	65 of 68
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1
				<b>Status:</b>	Final

```

    "object_relation": null,
    "value": "716",
    "Galaxy": [],
    "ShadowAttribute": []
  },
  {
    "id": "452849",
    "type": "target-machine",
    "category": "Targeting data",
    "to_ids": false,
    "uuid": "5c87a306-54b0-450a-b37b-354a0a00020f",
    "event_id": "69123",
    "distribution": "5",
    "timestamp": "1552392966",
    "comment": "Destination IP associated to the detected alarm.",
    "sharing_group_id": "0",
    "deleted": false,
    "disable_correlation": true,
    "object_id": "0",
    "object_relation": null,
    "value": "aaaa:2",
    "Galaxy": [],
    "ShadowAttribute": []
  },
  {
    "id": "452850",
    "type": "other",
    "category": "External analysis",
    "to_ids": false,
    "uuid": "5c87a306-a53c-4c95-8a99-354a0a00020f",
    "event_id": "69123",
    "distribution": "5",
    "timestamp": "1552392966",
    "comment": "Risk value evaluated by XL-SIEM",
    "sharing_group_id": "0",
    "deleted": false,
    "disable_correlation": true,
    "object_id": "0",
    "object_relation": null,
    "value": "4",
    "Galaxy": [],
    "ShadowAttribute": []
  },
  {
    "id": "452851",
    "type": "other",
    "category": "Internal reference",
    "to_ids": false,
    "uuid": "5c87a306-1c74-44b2-8e4c-354a0a00020f",
    "event_id": "69123",
    "distribution": "5",
    "timestamp": "1552392966",
    "comment": "Organization where the XL-SIEM Agent has been deployed",
    "sharing_group_id": "0",
    "deleted": false,
    "disable_correlation": true,
    "object_id": "0",
    "object_relation": null,
    "value": "ATOS",
    "Galaxy": [],
    "ShadowAttribute": []
  },
  {
    "id": "452852",
    "type": "other",
    "category": "External analysis",
    "to_ids": false,
    "uuid": "5c87a306-3b64-4cf5-aece-354a0a00020f",
    "event_id": "69123",

```

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	66 of 68
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1
				<b>Status:</b>	Final

```

    "distribution": "5",
    "timestamp": "1552392966",
    "comment": "Reliability value evaluated by XL-SIEM",
    "sharing_group_id": "0",
    "deleted": false,
    "disable_correlation": true,
    "object_id": "0",
    "object_relation": null,
    "value": "6",
    "Galaxy": [],
    "ShadowAttribute": []
  },
  {
    "id": "452853",
    "type": "other",
    "category": "Other",
    "to_ids": false,
    "uuid": "5c87a306-8de4-4e0b-bdce-354a0a00020f",
    "event_id": "69123",
    "distribution": "5",
    "timestamp": "1552392966",
    "comment": "Userdata1",
    "sharing_group_id": "0",
    "deleted": false,
    "disable_correlation": true,
    "object_id": "0",
    "object_relation": null,
    "value": "PAA",
    "Galaxy": [],
    "ShadowAttribute": []
  },
  {
    "id": "452854",
    "type": "other",
    "category": "Other",
    "to_ids": false,
    "uuid": "5c87a306-e4b0-4429-823e-354a0a00020f",
    "event_id": "69123",
    "distribution": "5",
    "timestamp": "1552392966",
    "comment": "PluginID",
    "sharing_group_id": "0",
    "deleted": false,
    "disable_correlation": true,
    "object_id": "0",
    "object_relation": null,
    "value": "70000",
    "Galaxy": [],
    "ShadowAttribute": []
  },
  {
    "id": "452855",
    "type": "other",
    "category": "Other",
    "to_ids": false,
    "uuid": "5c87a306-f590-4031-a73b-354a0a00020f",
    "event_id": "69123",
    "distribution": "5",
    "timestamp": "1552392966",
    "comment": "PluginSID",
    "sharing_group_id": "0",
    "deleted": false,
    "disable_correlation": true,
    "object_id": "0",
    "object_relation": null,
    "value": "5",
    "Galaxy": [],
    "ShadowAttribute": []
  }
}],

```

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	67 of 68	
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b> Final

```

"ShadowAttribute": [],
"RelatedEvent": [],
"Galaxy": [],
"Object": [],
"Tag": [
  {
    "id": "10",
    "name": "xl-siem:category=\"authentication\"",
    "colour": "#340900",
    "exportable": true,
    "hide_tag": false,
    "user_id": "0",
    "numerical_value": null
  },
  {
    "id": "68",
    "name": "xl-siem:sub-category=\"bruteforce\"",
    "colour": "#5f1100",
    "exportable": true,
    "hide_tag": false,
    "user_id": "0",
    "numerical_value": null
  },
  {
    "id": "344",
    "name": "smesec:tool='xl-siem'",
    "colour": "#55a7f2",
    "exportable": true,
    "hide_tag": false,
    "user_id": "0",
    "numerical_value": null
  }
]}
}

```

Figure 20: Example of alert in JSON format

<b>Document name:</b>	D3.8 Public Report of SMESEC Framework and Unified Architecture			<b>Page:</b>	68 of 68	
<b>Reference:</b>	D3.8	<b>Dissemination:</b>	PU	<b>Version:</b>	1.1	<b>Status:</b> Final