

RESISTO:

D2.6_RESISTO platform and tools reference architecture - first



RESISTO

D2.6 – RESISTO PLATFORM AND TOOLS REFERENCE ARCHITECTURE - FIRST

Document Manager:	Alberto Neri	LDO	Editor
--------------------------	--------------	-----	--------

Project Title:	RESilience enhancement and risk control platform for communication infraSTructure Operators
Project Acronym:	RESISTO
Contract Number:	786409
Project Coordinator:	LEONARDO
WP Leader:	BTC

Document ID N°:	RESISTO_D2.6_190516_01	Version:	1.0
Deliverable:	D2.6	Date:	16/05/2019
		Status:	APPROVED

Document classification	PUBLIC
--------------------------------	---------------

Approval Status	
Prepared by:	Alberto NERI (RM3), Annarita DI LALLO (LDO)
Approved by: (WP Leader)	Zhan CUI (BTC)
Approved by: (Coordinator)	Federico FROSALI (LDO)
Advisory Board Validation (Advisory Board Coordinator)	NA
Security Approval (Security Advisory Board Leader)	Alberto BIANCHI (LDO)

CONTRIBUTING PARTNERS

Name	Company / Organization	Role / Title
Alberto Neri, Annarita Di Lallo, Lucio Brighella, Guido Mariotta	LDO	Engineering Technology Dept., Cyber & ICT experts
Mirjam Fehling-Kaschek, Jörg Finger	EMI	Scientific Researcher
Alessandro Neri, Marco Carli, Stefano Panzieri, Cosimo Palazzo	RM3	
Cosimo Zotti, Giuseppe Celozzi	TEI	
Carmen Patrascu	ORO	
Risto Laanoja	GT	
Andrei Avădănei	BSS	
Moisés Valeo, Javier Valera	INT	
Rodoula Makri, Panagiotis Karaivazoglou, Alexandros Kyritsis, Nikolaos Uzunoglu, Apostolos Papafragkakis	ICCS	Senior Researchers / Electrical Engineers
Michael A. Skitsas	ADI	

DISTRIBUTION LIST

Name	Company / Organization	Role / Title
PMT	RESISTO CONSORTIUM	NA
Markus MULLER	EC DG REA	EC Programme Officer
General Public	NA	NA

REVISION TABLE

Version	Date	Modified Pages	Modified Sections	Comments
0.9	07.01.2019	All	All	Release for SAB review
1.0	16.05.2019	All	All	Final release

COPYRIGHT STATEMENT



© 2018-2021 This document and its content are the property of the RESISTO 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 RESISTO Consortium or the Partners detriment and are not to be disclosed externally without prior written consent from the RESISTO Partners. Each RESISTO Partner may use this document in conformity with the RESISTO Consortium Grant Agreement provisions.

The research leading to these results has received funding from the European Union's Horizon 2020 Research and Innovation Program, under the Grant Agreement No 786409.

The views and opinions in this document are solely those of the authors and contributors, not those of the European Commission.

PROJECT CONTACT



LEONARDO

Via delle Officine Galileo 1 – Campi Bisenzio (FI) – 50013 – Italy

Tel.: +39 055 5369640, Fax: +39 055 5369640

E-Mail: frederico.frosali@leonardocompany.com

RESISTO PROJECT – PUBLISHABLE EXTENDED ABSTRACT

Communications play a fundamental role in the economic and social well-being of the citizens and on operations of most of the CIs. Thus they are a primary target for criminals having a multiplier effect on the power of attacks and providing enormous resonance and gains. Also extreme weather events and natural disasters represents a challenge due to their increase in frequency and intensity requiring smarter resilience of the Communication CIs, which are extremely vulnerable due to the ever-increasing complexity of the architecture also in light of the evolution towards 5G, the extensive use of programmable platforms and exponential growth of connected devices. The fact that most enterprises still manage physical and cyber security independently represents a further challenge. RESISTO platform is an innovative solution for Communication CIs holistic situation awareness and enhanced resilience (aligned with ECSO objectives). Based on an Integrated Risk and Resilience analysis management and improvement process availing all resilience cycle phases (prepare, prevent, detect, absorb, etc.) and technical resilience capabilities (sense, model, infer, act, adopt), RESISTO implements an innovative Decision Support System to protect communication infrastructures from combined cyber-physical threats exploiting the Software Defined Security model on a suite of state of the art cyber/physical security components (Blockchain, Machine Learning, IoT security, Airborne threat detection, holistic audio-video analytics) and services (Responsible Disclosure Framework) for detection and reaction in presence of attacks or natural disasters. Through RESISTO Communications Operators, will be able to implement a set of mitigation actions and countermeasures that significantly reduce the impact of negative events in terms of performance losses, social consequences, and cascading effects in particular by bouncing efficiently back to original and forward to operational states of operation.

EXECUTIVE SUMMARY

This document D2.6 “RESISTO platform and tools reference architecture - first” illustrates the reference architecture of the RESISTO platform by specifying its overall static and behavioural structure, its functions, its internal and external interfaces.

From a methodological point of view, the document first provides an overview of the RESISTO system, next describes and refines the RESISTO architecture detailing the system sub-systems and components. Functional modules (e.g., the Risk Predictor, the Workflow Manager, and the Orchestration Controller) and infrastructure elements (e.g., the Data Integration Layer and the Cockpit) are both presented and assessed. The sections of the document related to the Concept of execution, which captures the dynamic behaviour of the system, and to the Interfaces, which instead depict the services provided and requested by each component, are still empty and will be presented in deliverable D2.7 “RESISTO platform and tools reference architecture - final” at month 15.

In fact, the RESISTO reference architecture is defined according to a two-step process.

Aim of the first step is to fix the main system functional flows and to take a snapshot of the system components, so to identify their role and the services/interfaces they mean to provide. The present document represents exactly this first step.

By carrying out the second step, the relationships among the system components will be defined to properly drive the following component development and adaption actions as well as the platform integration phase. On the purpose, the results of further activities (e.g., the definition of the use cases for RESISTO operational validation) that are planned to be performed after the first step will be exploited.

CONTENTS

ABBREVIATIONS	12
1. INTRODUCTION	16
1.1. Document Identification	16
1.2. Document Overview	16
2. RESISTO PLATFORM OVERVIEW	18
3. SYSTEM ARCHITECTURE	20
3.1. Long Term Control Loop Functional Architecture	21
3.2. Short Term Control Loop Functional Architecture	22
3.3. Architecture Components	24
4. DETAILED DESCRIPTION OF ARCHITECTURE COMPONENTS	26
4.1. Long Term Control Loop	26
4.2. Knowledge base	28
4.3. Short Term Control Loop	31
4.4. Cockpit	58
4.5. Physical Resources Monitoring	61
4.6. Network Resources Monitoring	73
5. CONCEPT OF EXECUTION	81
6. LOGICAL INTERFACE DESIGN	82
7. REFERENCES	83

INDEX OF FIGURES

Figure 1 – Deliverables for RESISTO WP2 “Use cases and holistic systems modelling”	16
Figure 2 – Cybersecurity Framework's five Functions.....	18
Figure 3 – RESISTO overall concept.	19
Figure 4 – RESISTO functionalities and control loops.	19
Figure 5 – RESISTO logical architecture.....	20
Figure 6 – Long Term Control Loop Functional Architecture.	21
Figure 7 – Short Term Control Loop Functional Architecture.....	22
Figure 8 – RESISTO architecture components.	24
Figure 9 – Risk and resilience management process for the long term control loop, based on [Ref2].	27
Figure 10 – The layers of KSI Infrastructure.....	29
Figure 11 - Deployment model of the Knowledge Base.	29
Figure 12 – Vulnerability Disclosure Framework programs.....	31
Figure 13 – Mule Environment.	32
Figure 14 – Apache Kafka Logical Architecture.	33
Figure 15 – Correlator components.....	34
Figure 16 – Raw data from a SCADA source.....	35
Figure 17 – Correlator main data flow.	35
Figure 18 – Example of a correlation rule.....	36
Figure 19 – Example of correlation.....	36
Figure 20 – Logical Architecture of the Cyber/Physical Threats Correlator.	37
Figure 21 – Apache Storm: Topology Nodes.	38
Figure 22 – Apache Flume architecture.	40
Figure 23 – Risk Predictor Architecture.....	41
Figure 24 – CISI Apro SQL input data structure.....	42
Figure 25 – CISI Apro SQL output data structure.	42
Figure 26 – CISI Apro user interface.	43
Figure 27 – CISI Apro Module: Layers & Resources.	44
Figure 28 – CISI Apro Module: Entity Maker.....	45
Figure 29 – CISI Apro Module: Modeler.....	46
Figure 30 – CISI Apro Module: State Variables.	47
Figure 31 – CISI Apro Module: Link State.....	48
Figure 32 – CISI Apro Module: Simulation.	49
Figure 33 – URANIUM platform civil protection panel.....	50

Figure 34 – ATENA Platform.	51
Figure 35 – Mitigation Module of the Short Term Control Loop.	52
Figure 36 – Activiti Engine.	53
Figure 37 – BPMN task and process.	53
Figure 38 – Example of process and task.	54
Figure 39 – EWC function architecture.	57
Figure 40 – Functionalities of RESISTO Cockpit.	58
Figure 41 – Leonardo SC2 WebViewer.	59
Figure 42 – Leonardo SC2 Map Viewer.	60
Figure 43 – Item representation in the CROP (Common Relevant Operational Picture).	60
Figure 44 – CISIAview (Risk Predictor HMI).	61
Figure 45 – Orange Romania Smart Site Management System.	63
Figure 46 – Audio Analytics System.	68
Figure 47 – Video Analytics System.	68
Figure 48 – Intelligence Audio/Video Surveillance System.	69
Figure 49 – ORO SOC: Architecture Overview.	76
Figure 50 – ORO SOC: response workflow.	77
Figure 51 – ORO SOC: operational model.	78
Figure 52 – ORO SOC: four-tier hierarchy.	79

INDEX OF TABLES

Table 1 – Framework's five Functions for Communication Infrastructures.	18
Table 2 – RESISTO components and correspondent paragraphs.	25
Table 3 – Deliverables of WP3, defining the long term control loop.	26
Table 4 – Planned interfaces for the Risk Predictor Module.	51
Table 5 – Data exchange between the ICCS detection system and the RESISTO platform.	67
Table 6 – Input and output interfaces for the audio/video analytics component.	70

ABBREVIATIONS

2G, 3G, 4G, 5G	Second, third, fourth and fifth generation of mobile phone systems
AAC	Audio Analytics Component
AC	Alternating Current
ADI	ADITESS
ALB	Altice Labs SA – Telecom Portugal
AP	Access Point
API	Application Programming Interface
APN	Access Point Name
B2B	Business to Business
BLE	Bluetooth Low Energy
BPM	Business Process Model
BPMN	Business Process Model and Notation
BSS	BIT SENTINEL SECURITY
BSSID	Basic Service Set Identifier
BTC	British Telecom – British Telecommunications Public Limited Company
CER	Istituto Superiore delle Comunicazioni e delle Tecnologie dell'Informazione – IT-CERT
CESG	Communications-Electronics Security Group
CI	Critical Infrastructures
CIP	Critical Infrastructure Protection
CISIApro	Critical Infrastructure Simulation by Interdependent Agents
CROP	Common Relevant Operational Picture
CSRF	Common Source Route File
CW	Continuous Wave
DB	Database
DC	Direct Current
DoS	Denial of Service

ECGI	E-UTRAN Cell Global Identifier
EMI	Ernst-Mach-Institut - Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V.
EPL	Event Processing Language
ESB	Enterprise Service Bus
EWCF	Emergency Warning Communication Function
GCS	Ground Control Station
GDPR	General Data Protection Regulation
GIS	Geographic Information System
GPS	Global Positioning System
GPU	Graphics Processing Unit
GT	GUARDTIME AS
HMI	Human Machine Interface
HVAC	Heating, Ventilation, & Air Conditioning
ICCS	Institute of Communication & Computer Systems - National Technological University of Athens
INT	INTEGRASYS S.A
IoT	Internet of Things
IP	Internet Protocol
IT	Information Technology
JMS	Java Message Service
KPI	Key Performance Indicator
LAN	Local Area Network
LDO	Leonardo S.p.A.
LTE	Long Term Evolution
MAC	Media Access Control
MCC	Mobile Country Code
MNC	Mobile Network Code
MHR	Mixed-Holistic-Reductionist

ML	Machine Learning
MSSP	Managed Security Solutions Provider
MTTR	Mean Time to Repair
NFV	Network Function Virtualization
NIST	National Institute of Standards and Technology
OCU	Orchestrator Central Unit
ORO	Orange Romania SA
OSINT	Open Source INTelligence
OSS	Operational Support System
OTE	Hellenic Telecommunications Organization S.A. - (Organismos Tilepikoinonion tis Ellados)
PCI	Physical Cell ID
PIR	Post Incident Report
PSIM	Physical Security Information Management
PTZ	Pan, Tilt and Zoom
QoS	Quality of Service
RAN	Radio Access Network
RCS	Radar Cross Section
RESISTO	RESilience enhancement and risk control platform for communication infraSTructure Operators
RFID	Radio Frequency Identification
RM3	Dipartimento di Ingegneria Università degli Studi Roma Tre
RSRP	Reference Signal Received Power
RSSI	Received Signal Strength Indication
RTSP	Real Time Streaming Protocol
RTV	Retevisión I, S.A. - Cellnex Telecom, S.A
SCADA	Supervisory Control And Data Acquisition
SDN	Software Defined Networking
SDR	Software Defined Radio

SDS	Software Defined Security
SIEM	Security Information and Event Management
SOC	Security Operating Center
SOP	Security Operational Procedure
SSID	Service Set Identifier
TAC	Tracking Area Code
TCP	Transmission Control Protocol
TEI	Ericsson Telecomunicazioni SpA
TIM	Telecom Italia Mobile – TELECOM ITALIA S.p.A.
TLC	Telecommunication
UAV	Unmanned Aerial Vehicle
VAC	Video Analytics Component
VPN	Virtual Private Network
WP	Work Package
WSN	Wireless Sensor Network

1. INTRODUCTION

1.1. Document Identification

The present deliverable D2.6 “RESISTO platform and tools reference architecture - first” illustrates the current status and outputs of Task 2.4 “RESISTO reference architecture for long term preparation and short term disruptions”, which is included in Work Package WP2 “Use cases and holistic systems modelling”. The activities are ongoing and the final results will be presented in deliverable D2.7 “RESISTO platform and tools reference architecture - final” at month 15 (M15), as shown in Figure 1.

Deliverable Number ¹⁴	Deliverable Title	WP number ⁹	Lead beneficiary	Type ¹⁵	Dissemination level ¹⁶	Due Date (in months) ¹⁷
D1.11	Annual Reports - second	WP1	1 - LDO	Report	Confidential, only for members of the consortium (including the Commission Services)	24
D2.1	End user requirements for integrated cyber-physical risk and resilience management, platform and tools	WP2	13 - BUW	Report	Public	10
D2.2	Cyber-physical threat/ risk scenarios and pre-assessment - first	WP2	11 - Fraunhofer	Report	Public	6
D2.3	Cyber-physical threat/ risk scenarios and pre-assessment - final	WP2	11 - Fraunhofer	Report	Public	12
D2.4	Telecommunication system model and interfaces - first	WP2	11 - Fraunhofer	Report	Public	6
D2.5	Telecommunication system model and interfaces - final	WP2	11 - Fraunhofer	Report	Public	12
D2.6	RESISTO platform and tools reference architecture - first	WP2	1 - LDO	Report	Public	8
D2.7	RESISTO platform and tools reference architecture - final	WP2	1 - LDO	Report	Public	15
D2.8	Table-top read teaming results of RESISTO architecture, scenarios and use cases	WP2	4 - OTE	Report	Public	15

Figure 1 – Deliverables for RESISTO WP2 “Use cases and holistic systems modelling”.

1.2. Document Overview

The document is organized as follows:

- section 2 provides an overview of the RESISTO platform;

- section 3 logically and functionally describes the RESISTO architecture;
- section 4 illustrates in detail the components of the RESISTO platform;
- section 5 explains the concept of execution among the software units composing the RESISTO platform;
- section 6 describes the logical interfaces among the RESISTO components;
- section 7 lists the references.

Some of the sections and subsections will be provided with the final issue of the document, which is deliverable D2.7 “RESISTO platform and tools reference architecture - final” at M15. In fact, they need further activities that are planned to be carried out after the issue of the present D2.6 “RESISTO platform and tools reference architecture - first”. For example, sections 5 and 6 need integration work and activities, that have just started in WP6 (M7-M19). Similarly, subsections 4.5.1 and 4.6.1, which respectively describe the physical and network monitoring data provided by the communication operators, need the definition and exhaustive description of the pilot use cases (D2.8 at M15).

2. RESISTO PLATFORM OVERVIEW

The NIST (National Institute of Standards and Technology) Framework for Improving Critical Infrastructure Cybersecurity, commonly referred to as the NIST Cybersecurity Framework, provides private sector organizations with a structure for assessing and improving their ability to prevent, detect and respond to cyber incidents. Version 1.0 was published by the US NIST in 2014 and was aimed at operators of critical infrastructure. The Cybersecurity Framework's five Functions, which represent the five primary pillars for a successful and holistic cybersecurity program, are depicted in Figure 2.



Figure 2 – Cybersecurity Framework's five Functions.

These five functions are not only applicable to cybersecurity risk management, but also to risk management at large. As such, they have been used to define the functionalities of the RESISTO platform, whose main objective is to improve risk control and resilience of modern Communication Infrastructures, against a wide variety of cyber-physical threats, being those malicious attacks, natural disasters or even un-expected faults. The definition of the five core functions with specific reference to Communication Infrastructure security against cyber-physical threats is summarized in Table 1.

Function	Description
Identification	Define and maintain a knowledge base on physical and cyber security risks to systems, assets, data, and capabilities characterizing Communication Infrastructures.
Protection	Develop and implement the appropriate safeguards to ensure delivery of Communication Infrastructure services.
Detection	Early and timely discover the occurrence of physical and cyber security events.
Response	Orchestrate and implement effective response to a detected security event.
Mitigation	Develop and implement the appropriate activities to mitigate the impacts of the threat and to restore as much as possible capabilities or services that were impaired due to a security event.

Table 1 – Framework's five Functions for Communication Infrastructures.

RESISTO is meant to be an innovative solution for Communication Infrastructure holistic (physical, logical) situation awareness and enhanced resilience. It fosters integrated risk-resilience assessment of the Communication Infrastructure, faster detection of threats/attacks, better informed decision making and achievement of holistic understanding of a situation across the cyber and physical domain and interlinked Critical Infrastructures (CI). These capabilities allow for better reaction and more efficient selection of countermeasure and mitigation actions (Figure 3).

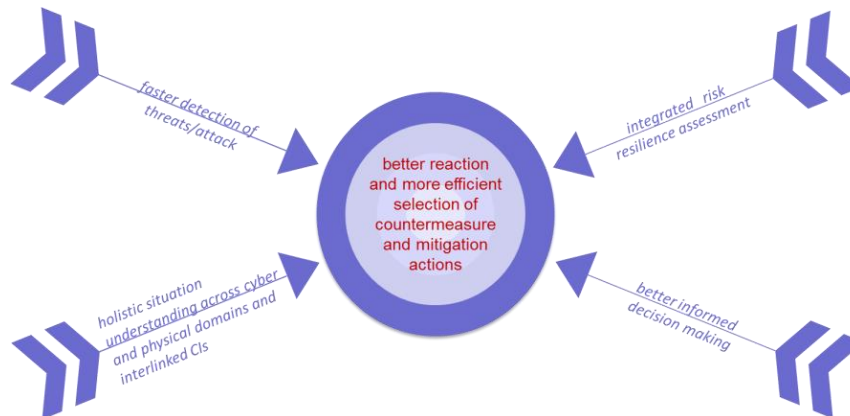


Figure 3 – RESISTO overall concept.

As depicted in Figure 4, RESISTO encompasses two different control loops:

- the Long Term Control loop carries out the “Identification” and “Protection” functions;
- the Short Term Control loop carries out the “Detection”, “Reaction” and “Mitigation” functions.

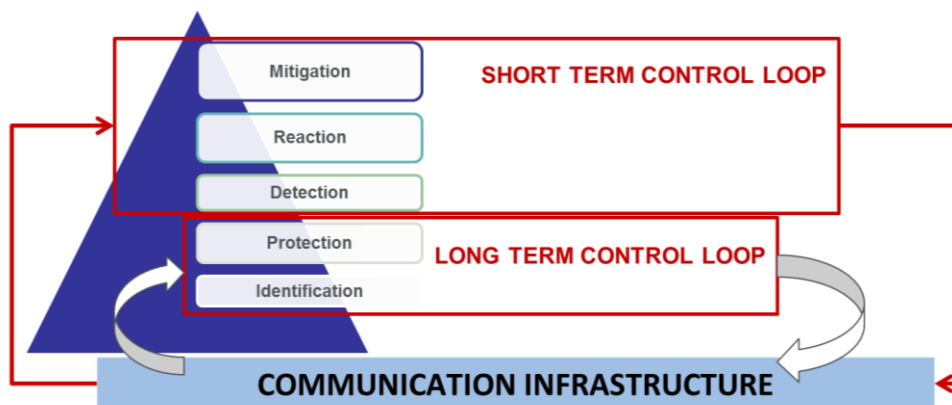


Figure 4 – RESISTO functionalities and control loops.

Both loops run on top of the Communication Infrastructure in order to ensure re-use of already deployed security solutions and deliver a holistic approach to preparation, prevention, detection, response and mitigation of the combination of physical and cyber threats to the critical infrastructure.

The detailed description of RESISTO innovative platform is provided in the following chapters.

3. SYSTEM ARCHITECTURE

The logical architecture of RESISTO is depicted in Figure 5. The platform integrates two control loops both running on top of the Communication Infrastructure and strongly interlinked with each other.

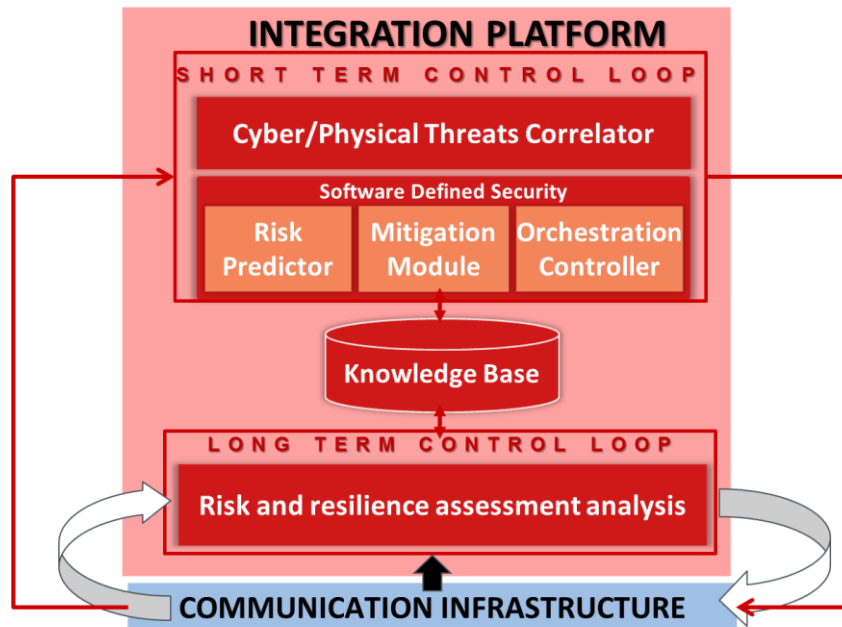


Figure 5 – RESISTO logical architecture.

- The **Long Term Control Loop** is an **offline** component. It assesses asset vulnerabilities and security threats and consequently defines the configuration of the system, updating it on a periodic basis or when particular events take place (new threats or discovery of previously undetected vulnerabilities). The long term risk and resilience assessment and improvement process is typically conducted annually, quarterly or even monthly. It determines criticalities and long term strategies.
- The **Short Term Control Loop** is the **runtime** component (i.e., it responds within minutes down to milliseconds) of the platform. As such, it promptly reacts to attacks and threats that may impact the operational life of the system. It enhances situation awareness, immediate response and bouncing back up to forward to even better states of systems as preselected by the long-term approach. It is built around the concept of Software Defined Security (SDS) that performs a dynamic, flexible reconfiguration of security/resilience mechanisms and relocation (virtualization) of security functions, in a way similar to what currently done in SDN (Software Defined Networking). In particular:
 - the **Cyber/Physical Threats Correlator** provides the timely detection of cyber/physical attacks;
 - the **Risk Predictor** simulates the impact of anomalies and security attacks on the Communication Infrastructure and interlinked CIs;
 - the **Mitigation Module** selects the best response/mitigation strategy and defines the actions to be enforced to mitigate the risks or to recover from a damaged situation;

- the **Orchestration Controller** manages the cyber physical resources needed to apply the security policies stated by the Mitigation Module.

3.1. Long Term Control Loop Functional Architecture

The RESISTO **Long Term Control loop** is in charge of:

- defining the configuration of the Communication Critical Infrastructure (CI) according to the security assessment, and
- updating it on a periodic basis, in the order of months, or when particular events take place (new threats or discovery of previously undetected vulnerabilities).

The functional architecture of the Long Term Control loop is depicted in Figure 6. It substantially performs a “**Risk and resilience assessment analysis**” that **off-line**:

- identifies the context,
- analyzes the interdependencies (physical, cyber, logical and geographical) and the risks,
- evaluates semi-quantitatively and quantitatively the recognized risks,
- suggests the risks treatment and “**Resilience indicators**” as aggregate measures of resilience of the communication CI in its operational phase.

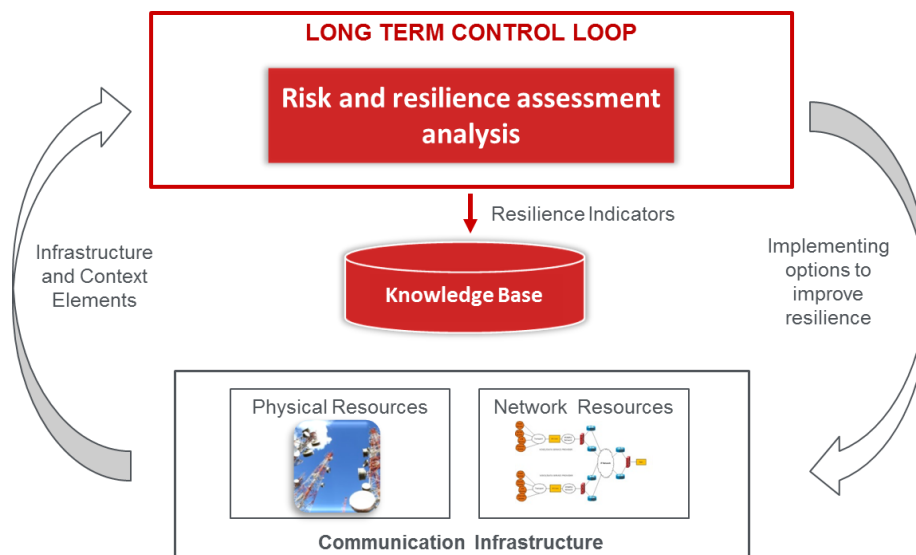


Figure 6 – Long Term Control Loop Functional Architecture.

The risk and resilience assessment analysis is compliant with but extends ISO 31000. It is conducted in 9 steps (see section 4.1 for details). The long-term risk and resilience assessment process for cyber-physical threats results in a guidance application report as well as a support tool, which enforces sufficient consistency and rigor of the assessment and improvement process, as part of RESISTO’s solutions. In particular, the support tool suggests overall telecommunication key performance functions to the operators, possible threats of interest and certainly methods for assessment and improvement.

3.2. Short Term Control Loop Functional Architecture

The Short Term Control Loop is as a typical run-time control loop. It is in charge of detecting potential physical, cyber and physical/cyber combined threats and of promptly reacting to attacks and events that may impact the operational life of the system.

The **Short Term Control Loop**:

- **monitors the physical and cyber security status of the infrastructures**, correlating the physical and cyber domain events and communication KPIs (Key Performance Indicators) to detect anomalies and provide early warnings on security attacks by detecting threats in advance;
- **evaluates the attack impact** with respect to performance degradation of detected anomalies and security attacks on the communication CI, and interlinked CIs if known, based on the cascading effect;
- **supports decision making** providing a qualitative and quantitative What-If analysis tool in order to evaluate the most resilient communication CI reconfiguration;
- **drives reaction and mitigation** by means of action workflows (composed of directives to intervention teams, physical protection devices activation) and, mainly, of orchestrated Communication Network reconfiguration and protection function activation.

The Short Term Control Loop functional control flow is reported in Figure 7.

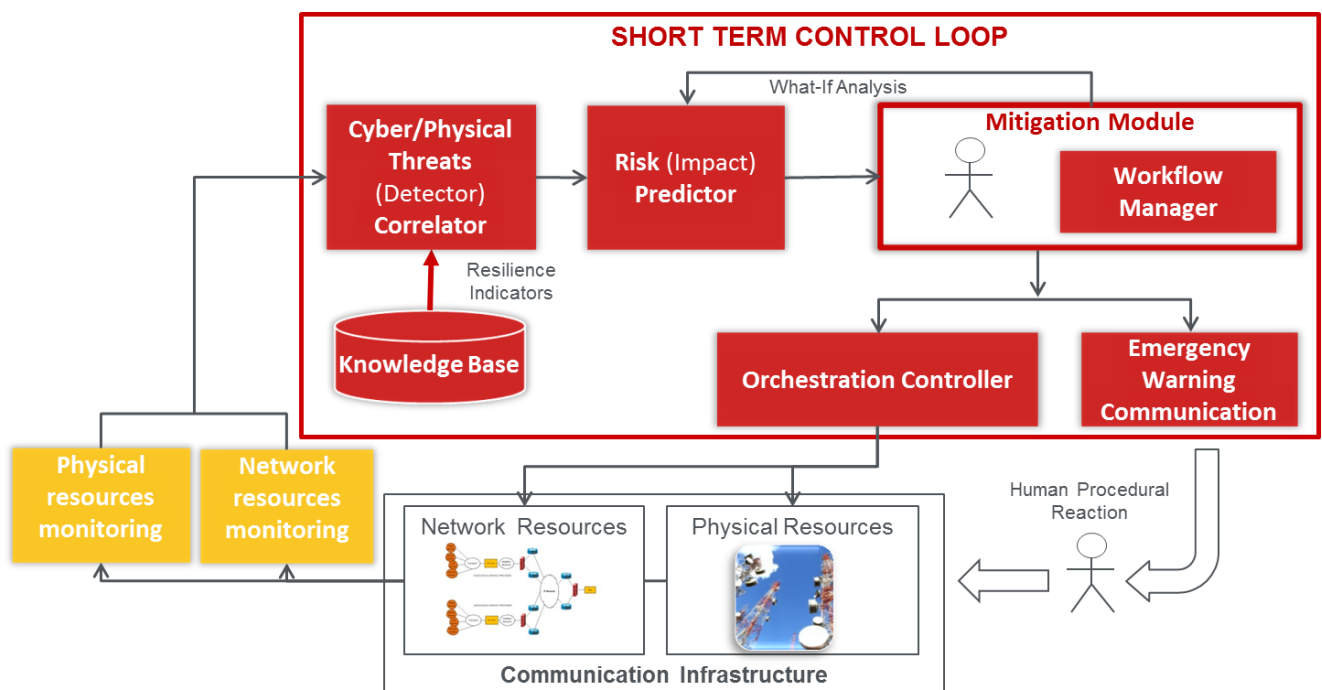


Figure 7 – Short Term Control Loop Functional Architecture.

Input data to the Short Term Control Loop can be grouped into the following categories:

- 1) physical threats;
- 2) cyber threats and attacks;

- 3) physical communication infrastructure monitoring data (e.g., power and energy usage information and faults, intrusions, attacks on telecom sites);
- 4) Communication Network monitoring data (e.g., traffic, alarms, faults).

The sources of such data and information are:

- legacy PSIM (Physical Security Information Management) system(s) of the communication infrastructure,
- legacy SOC (Security Operating Centers) of telecommunication operators,
- RESISTO additional physical threat detectors (e.g., airborne threats detection systems, smart spectrum surveillance),
- RESISTO additional cyber threat detectors such as OSINT (Open-Source Intelligence) based detectors.

From a functional point of view input data are collected by the **Cyber/Physical Threats Correlator**. This is a rule based engine applying customized rules based on the Long Term Control Loop off-line analysis. The Correlator not only propagates the detected threats but generates alarms from apparently harmless events and monitoring data as well, to detect anomalies to be processed. The latter action is performed by using several event correlation techniques, such as logical, causal and temporal correlation based on event time. The Correlator is equipped with a machine learning based module. Machine learning algorithms can be applied to communication monitoring data and provide parameters and thresholds to the rule based engine. In particular, the algorithm may be trained with typical traffic data either on geographical or time bases and allow early warnings about anomalies (e.g., unusual heavy network traffic on a specific cell at a specific time on a certain day of the week). For a more complete description of the Cyber/Physical Threats Correlator see section 4.3.2.

Anomalies detected by the Correlator trigger the **Risk (Impact) Predictor**. It evaluates and highlights the impacts of the detected anomaly on the communication infrastructure and, mainly, on the services provided by the infrastructure. The Risk Predictor Engine acts at run-time on a model of the communication infrastructure. The communication infrastructure is modelled off-line interlacing different points of view.

- Under a reductionist perspective, each infrastructure is decomposed into a web of interconnected physical elementary entities and their behaviour depends on the (mutual or not) interactions with the other reductionist elements;
- Applying a holistic approach, each infrastructure is modelled as a (logical) reality with its own identity, functional properties and recognizable boundaries. It interacts with other similar entities according to reduced identifiable set of relationships. With such perspective it is easy to identify roles that each infrastructure plays in a specific context.
- From a Service point of view, a Service Entity represents a logical element, organizational or real, that provides an aggregate resource such as a QoS (Quality of Service) level.

Moreover the Risk Predictor supports the decision making process allowing a “What-If analysis” and thus simulating the application of countermeasures and reconfiguration and their impact on the system resilience. This is a way to evaluate the most resilient answer to the threat according to the “Risk and resilience assessment analysis” performed by the Long Term Control Loop. For a more complete description of the Risk (Impact) Predictor see section 4.3.3.

The reaction phase is supported by a **Workflow Management** software engine. The most appropriate workflow is selected and executed. A workflow is a conditional sequence of steps. Each step can specify a procedural action (e.g., alert a reaction team with an emergency message), drive a physical actuator (e.g., lock a physical gate), carry out a complex action on the Communication Network (e.g.,

activate a Virtual Network Security Function, isolate a faulty or attacked component, reconfigure a part of the network, disable a 5G slice, etc.). For a more complete description of the Workflow Management see section 4.3.4.

Complex actions on the Communication Infrastructure are performed by the **Orchestration Controller**. It is built around the concept of Software Defined Security (SDS) taking advantage of the Network Function Virtualization (NFV) and Software Defined Networking (SDN) paradigms of the underlying communication network. The Orchestrator Controller implements complex security functions and services composing less complex/primitive security mechanisms/functions acting on physical resources (i.e. network physical equipment) as well as Virtual Network Functions in a NFV/5G perspective. The Orchestration Controller operates on a communication infrastructures already controlled by a telecommunication operator, so it works on top of a simple SDN Controller, on the northbound side, or on top of a more complex network Operational Support System (OSS). For a more complete description of the Orchestration Controller see section 4.3.5.

The **Emergency Warning Communication Function** is activated when there is a need for sending instant messages, targeted alerts and operating instructions to specific categories of users that are present in a certain area where events like natural disasters, physical or cyber-attacks are occurring. In particular, rescue teams called to execute actions on the infrastructure can leverage on the received information, both textual and visual. For a more complete description of the Emergency Warning Communication Function see section 4.3.6.

3.3. Architecture Components

The components of the RESISTO platform are presented in detail in Figure 8.

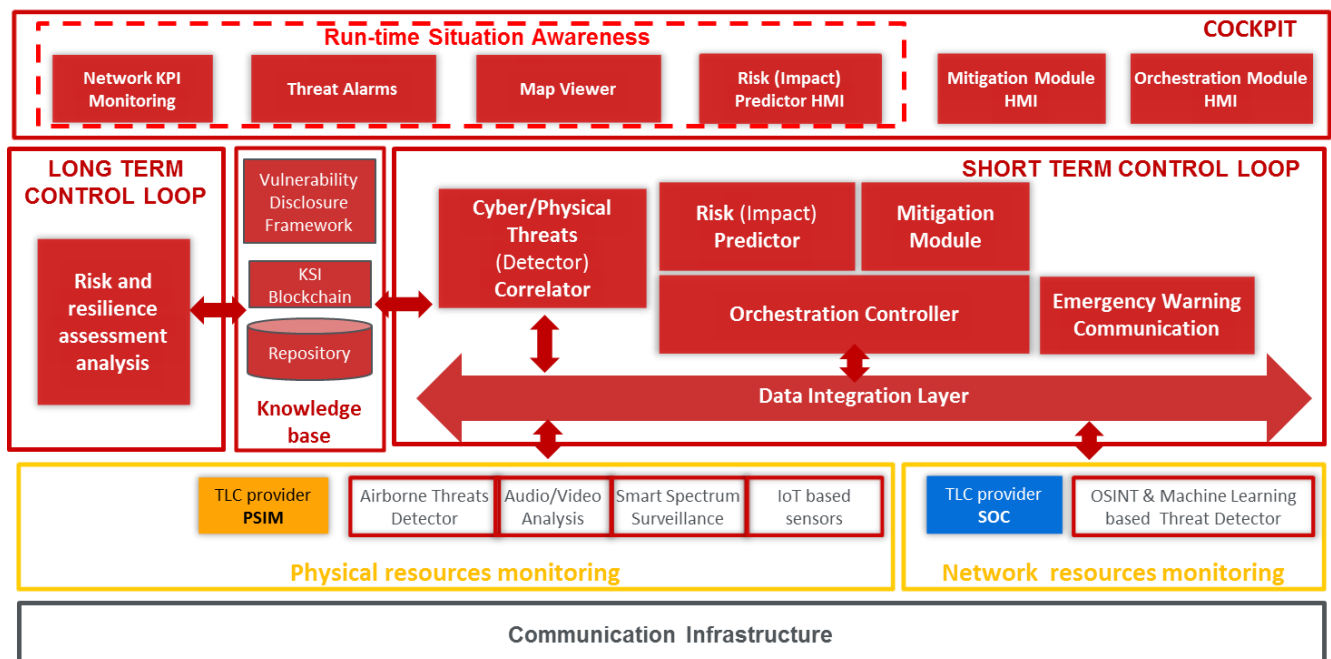


Figure 8 – RESISTO architecture components.

The detailed description of each component is provided in chapter 4. The correspondence between each component and the specific paragraph devoted to its description is reported in Table 2.

Some components, for example the Cockpit and the Data Integration Layer, provide common services to the functional components and as such have not been illustrated in the functional description of the architecture. Moreover, from a functional point of view, the components for Physical and Network Resources Monitoring are RESISTO specific Physical and Cyber Threats detectors.

Component	Sub-component	Paragraph
Long Term Control Loop		4.1
Knowledge Base		4.2
	Repository	4.2.1
	KSI Blockchain	4.2.2
	Vulnerability Disclosure Framework	4.2.3
Short Term Control Loop		4.3
	Data Integration Layer	4.3.1
	Cyber/Physical Threats Correlator	4.3.2
	Risk (Impact) Predictor	4.3.3
	Mitigation Module	4.3.4
	Orchestration Controller	4.3.5
	Emergency Warning Communication	
Cockpit		4.4
Physical Resources Monitoring		4.5
	PSIM	4.5.1
	Airborne Threats Detector	4.5.2
	Audio/Video analysis	4.5.3
	Smart Spectrum Surveillance	4.5.4
	IoT Based Sensors	4.5.5
Network Resources Monitoring		
	SOC	4.6.1
	OSINT and Machine Learning Threat Detector	4.6.2

Table 2 – RESISTO components and correspondent paragraphs.

4. DETAILED DESCRIPTION OF ARCHITECTURE COMPONENTS

4.1. Long Term Control Loop

The definition of the long term control loop is the main objective of WP3. Therefore, more detailed descriptions of the long term control loop and its tools will be provided in the deliverables of WP3, summarized in Table 3.

Task	Deliverable	Description
Task 3.1 Long term learning cyber-physical risk and resilience management	D3.1	Risk and resilience management process for cyber-physical threats of telecom CI - first
	D3.2	Risk and resilience management process for cyber-physical threats of telecom CI - final
Task 3.2 Methods/Plans for joint cyber-physical security management process	D3.3	Methods for cyberphysical security management for telecom CI - first
	D3.4	Methods for cyberphysical security management for telecom CI- final
Task 3.3 Physical protection and prevention methods: assessment and cyber-physical interaction	D3.5	Damage/ Vulnerability models for physical and cyber threats of telecom CI - first
	D3.6	Damage/ Vulnerability models for physical and cyber threats of telecom CI - final
Task 3.4 Risk and resilience quantities and related KPIs for telecommunications infrastructure	D3.7	KPIs, quantities and metrics for cyberphysical risk and resilience of telecom CI - first
	D3.8	KPIs, quantities and metrics for cyberphysical risk and resilience of telecom CI - final
Task 3.5 Desktop application to use case scenarios for second use cases refinement	D3.9	Analytical security assessment application to use cases and their refinement

Table 3 – Deliverables of WP3, defining the long term control loop.

The main output of the long term control loop is the risk and resilience analysis management process, which shall help to identify and evaluate risks and suggest treatment and mitigation strategies.

While the short term loop provides tools for direct reaction on attacks in real-time, the long term loop should be conducted on a periodic basis or in case of specific events, e.g. after changes in the setup of the infrastructure or detection of a new type of threats. As a result the long term loop leads to the identification of criticalities and definition of long term strategies.

The implementation of the long term control loop is based on a sophisticated risk and resilience management process extending the ISO 31000 standard [Ref2]. The process is structured into nine sequential steps to be followed in a closed loop, as shown in Figure 9. Specific input about the system, and possible threats and counter-measures is needed at different execution steps of the loop. This information is collected by a tabular Excel template. A web-application is developed to allow a fast and facile browsing through the tables and to infer further information, such as critical combinations of system functions and threats and a ranking of the threats, which serve as additional input to other steps of the resilience management process. This method allows a semi-quantitative assessment of critical risks. The resilience quantification of the critical risks is further supported by network simulation approaches, which can be performed by two simulation tools: the platform integrated simulator CISIApro of the short term control loop and the offline simulator CaESAR. In addition, the simulators allow to assess and rank possible mitigation strategies. It should be noted, that these simulations are not performed directly at the response to an attack but rather on a periodic basis to identify weak points of the current setup of the infrastructure. The results of the simulation processes will be summarized tabularly and integrated as output in the web-application. In Figure 9 the support of the tools to the different steps of the risk and resilience management is visualized. The usage of the tabular inputs and graphical network representations is indicated by orange boxes and the support by the Shiny app and simulation tools by blue or green boxes, respectively.

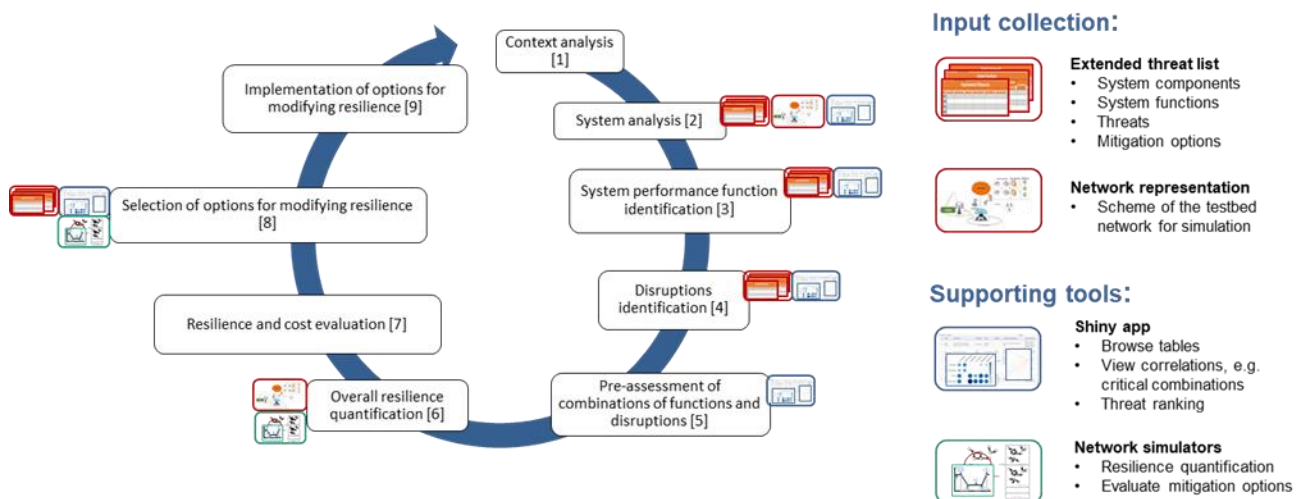


Figure 9 – Risk and resilience management process for the long term control loop, based on [Ref2].

In summary, two main tools to support the nine-steps of the risk and resilience management process were identified:

- 1) A web-application to browse the tabular inputs and inferred quantities thereof. The application is based on the Shiny package of the free programming language for statistical computing R.
- 2) Network simulators to quantify the resilience of the infrastructure for various setups, i.e. different classes of threats considering various mitigation options. The CISIApro simulator is described in detail in section 4.3.3. A description of the C++ based simulator CaESAR is given in D3.5. The simulations do not need to run in real-time on the server of the platform but may be outsourced on a periodic basis to a computing cluster or cloud to allow for the intensive computations.

It should be noted, that further methods and strategies are currently evaluated and discussed in WP3 leading to a possible extension of the list of tools for the long term control loop.

4.2. Knowledge base

The Knowledge base is composed of information about Communication CI configuration, security procedures, protection-reaction/mitigation strategies, recovery procedures and vulnerabilities. Stored data are secured and their integrity is monitored and enforced via blockchain technology.

4.2.1. Repository

The repository is a database for storing Knowledge Base items. Its actual vendor, product, configuration and schema will be decided based on interfaces to short term control loop components and long term control loop components; and other practical considerations, defined in collaboration with other consortium partners. In order to provide guaranteed data integrity and accountability the data stored in Repository is signed using KSI Blockchain. The integration is at higher abstraction level which is aware of 'documents', i.e. a set of related database entries forming a logically connected entity. Each version of a 'document' is signed using KSI Blockchain and cryptographically linked with previous versions, and with source data if available, and with attribution information of source and responsible person or system, thus establishing a provenance chain of documents. In order to provide lower level transparency, accountability and insider threat detection all transactions are recorded in trusted transaction log where all entries are linked and signed using KSI Blockchain. In order to provide privacy of extracted proofs, the signed log is redactable.

4.2.2. KSI Blockchain Infrastructure

KSI Blockchain is built and accessed through the KSI Infrastructure (Figure 10).

KSI Infrastructure is layered and hierarchical. KSI Blockchain is created and maintained by the KSI Core cluster; requests are accepted, and responses are distributed using a hierarchy of Aggregator nodes. The blockchain is distributed using a hierarchy of Extender nodes. Lowest layer, customer-facing Aggregator and Extender are packaged into the so-called Gateway server. The core and aggregation, extender networks are operated as a permissioned blockchain, by Guardtime. Branches of aggregation and distribution hierarchies can be operated by third parties.

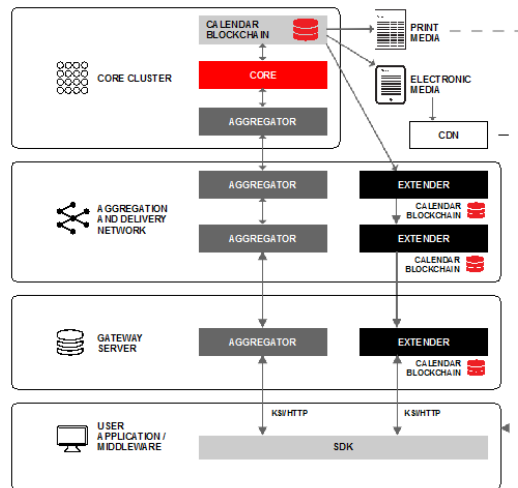


Figure 10 – The layers of KSI Infrastructure.

In RESISTO project Guardtime operates all KSI Infrastructure components, providing access to pre-configured endpoints (Figure 11). The gateway layer should be hosted on-premises for the best possible security and service quality. KSI signatures are server-based, meaning that signing is only possible with online access to the KSI service. The verification of the signatures can be done both offline and online.

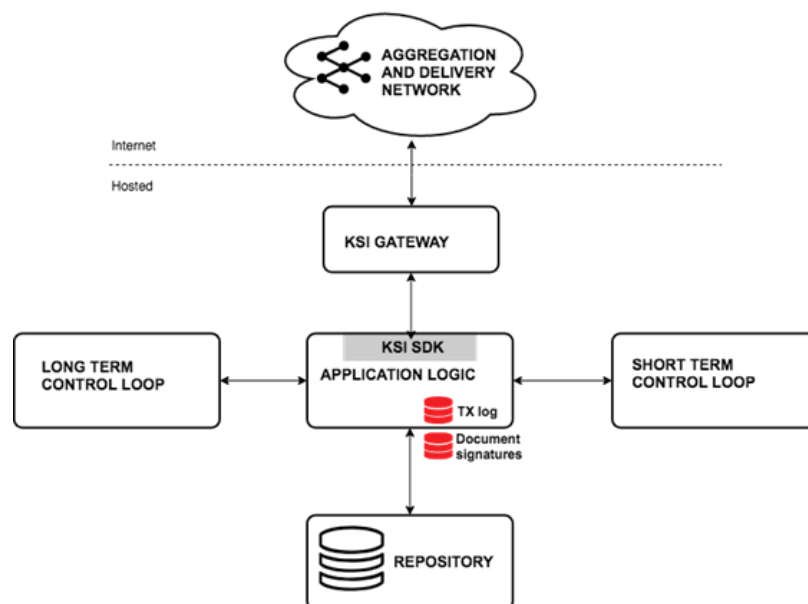


Figure 11 - Deployment model of the Knowledge Base.

4.2.3. Vulnerability Disclosure Framework

The Vulnerability Disclosure Framework aims at integrating vulnerability management functionalities within RESISTO framework by:

- fetching required information from the:
 - Knowledge Database, in order to define the potential vulnerabilities,
 - Network Resource Monitoring, in order to be able to know what systems and applications are in the scope;
- providing all the required functionalities for the internal teams in order to:
 - define the scope and rules for testing, for example:
 - IP range 192.168.0.0/24 is out of scope,
 - *.site.com are in the scope of testing,
 - CSRF (Common Source Route File) vulnerabilities are out of scope;
 - define the rewards for testing, for example:
 - finding an SQL injection vulnerability in a web app will be rewarded with 1000 €;
 - report potential vulnerabilities, for example:
 - the Security Researcher M.J. found a SQL Injection vulnerability on the accessible from target.company.com;
 - validate the potential vulnerability, for example:
 - based on the report received, the responsible team concluded that the M.J. report showcase a valid security issues, a unique ticket id is opened and the stakeholder was notified;
 - monitor the vulnerability through the whole cycle:
 - report the finding,
 - confirm/reject/request additional information from the security researcher,
 - notify the stakeholders,
 - patch the finding,
 - confirm from the security researcher that the issue was fixed,
 - reward the security researcher, if appropriate;
- Hall of Fame module that enables the company to motivate both internal and external security specialists to find bugs in order to both receive rewards but also remain on the Hall of Fame page, where the best security researchers are displayed;
- Rewarding module, that enables the company to reward the security researchers by swags, points for Hall of Fame or cash (fiat or cryptocurrencies).

The company can define both Public and Private programs in order to have a balance between the quality of reports and the risks they are exposing the networks. The company can decide if a vulnerability will be publicly disclosed or if they will keep it private, only as an internal future reference (Figure 12).

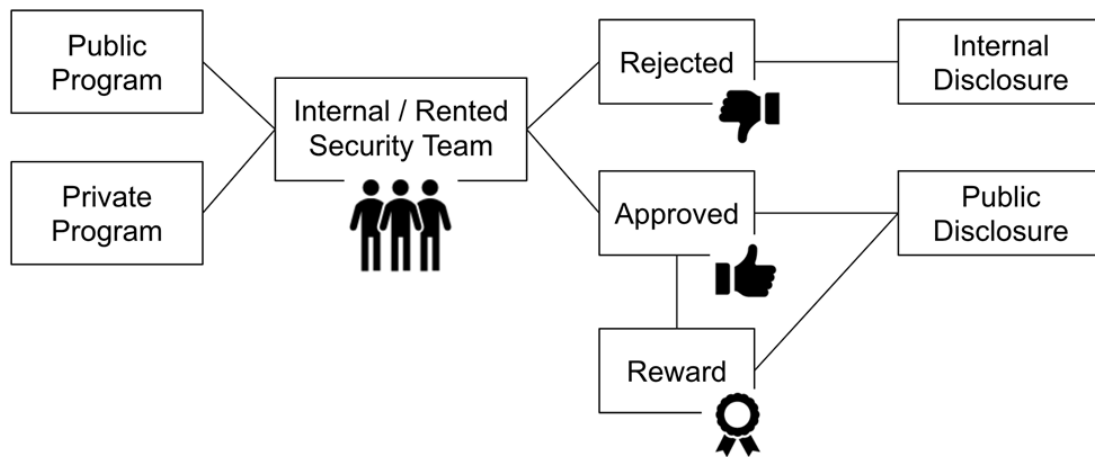


Figure 12 – Vulnerability Disclosure Framework programs.

The key innovation in this field is the combination of different modules from this framework with blockchain concepts in order to increase:

- the privacy of the security researchers,
- the rewards, by enabling ERC20 tokens / Ethereum rewards,
- the accounting, by providing a way to validate that a certain bug was previously reported.

4.3. Short Term Control Loop

4.3.1. Data Integration Layer

The Data Integration Layer is meant to receive and dispatch all information between the platform and the external systems. It is responsible of data transformation and normalization, message routing, data persistence.

The solution adopted for the Data Integration Layer is mainly based on Mule ESB (Enterprise Service Bus). The ESB module interfaces the different adapters (connectors) that make visible the peripheral systems. The implemented adapters allow to exchange data and information, diagnostics, commands with the interconnected vertical systems.

The thorough description of Mule is available at <https://www.mulesoft.com/resources/esb/what-mule-esb>. For sake of convenience, some key features are reported below.

Mule, the runtime engine of Anypoint Platform, is a lightweight Java-based enterprise service bus (ESB) and integration platform that allows developers to connect applications together quickly and easily, enabling them to exchange data. It enables easy integration of existing systems, regardless of the different technologies that the applications use, including JMS, Web Services, JDBC, HTTP, and more. The ESB can be deployed anywhere, can integrate and orchestrate events in real time or in batch, and has universal connectivity.

The key advantage of an ESB is that it allows different applications to communicate with each other by acting as a transit system for carrying data. Mule and other ESBs offer real value in scenarios where there are at least a few integration points or at least three applications to integrate. They are also well suited to scenarios where loose coupling, scalability and robustness are required.

Some Mule's capabilities are depicted in Figure 13:

- Service creation and hosting — expose and host reusable services, using the ESB as a lightweight service container;
- Service mediation — shield services from message formats and protocols, separate business logic from messaging, and enable location-independent service calls;
- Message routing — route, filter, aggregate, and re-sequence messages based on content and rules;
- Data transformation — exchange data across varying formats and transport protocols.

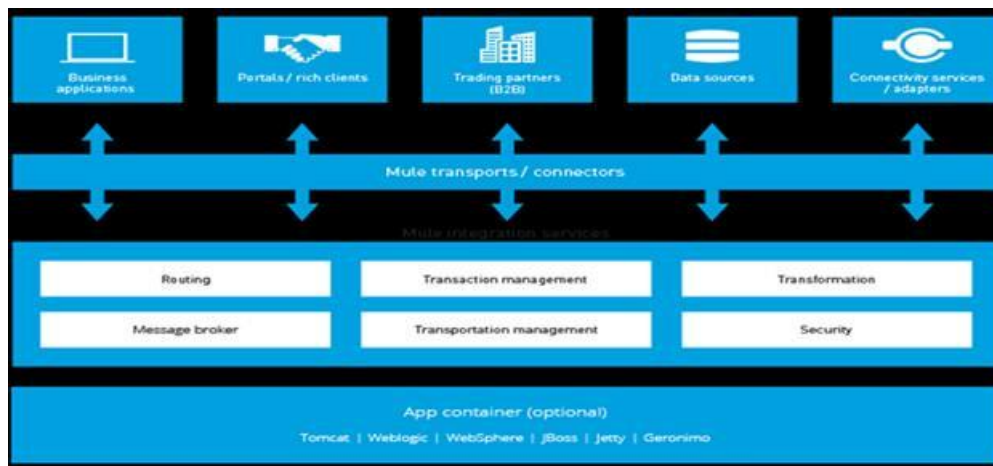


Figure 13 – Mule Environment.

There are two editions of Mule: Community and Enterprise. This enterprise-class version of Mule provides additional features and capabilities that are ideal for production deployments requiring performance, high availability, resiliency and technical support.

Another main element of Data integration Layer is the Message broker. A Message broker is an intermediary program that translates a system's language from one internationally suitable language to another via a telecommunication medium. It is primarily used for message validation, transformation and routing. A message broker is also known as middleware or integration broker. There are three types of message brokers – point-to-point, publish-subscribe, and a hybrid of both. The purpose of a broker is to:

- route or transfer messages to one or more destinations with an alternative representation,
- perform message aggregation, and to decompose them to multiple messages,
- send messages to their destination,
- recompose the responses into one message,
- respond to the messages received / errors,
- communicate with external repository to augment a message.

The Community and Enterprise editions of Mule provide native connectors for some common Message brokers such ActiveMQ and Kafka.

ActiveMQ is an open source message broker written in Java and Java Message Service (JMS) client. It consists of Enterprise Features fostering the communication from more than one client or server. Kafka is an open-source stream processing platform written in Scala and Java.

Apache Kafka is designed for high volume publish-subscribe messages and streams for a fast, durable, and scalable transmission. It is a distributed system implementing high-performance message management. Kafka is a system designed for real-time message exchange among distributed applications residing in clusters. The Kafka architecture is depicted in Figure 14.

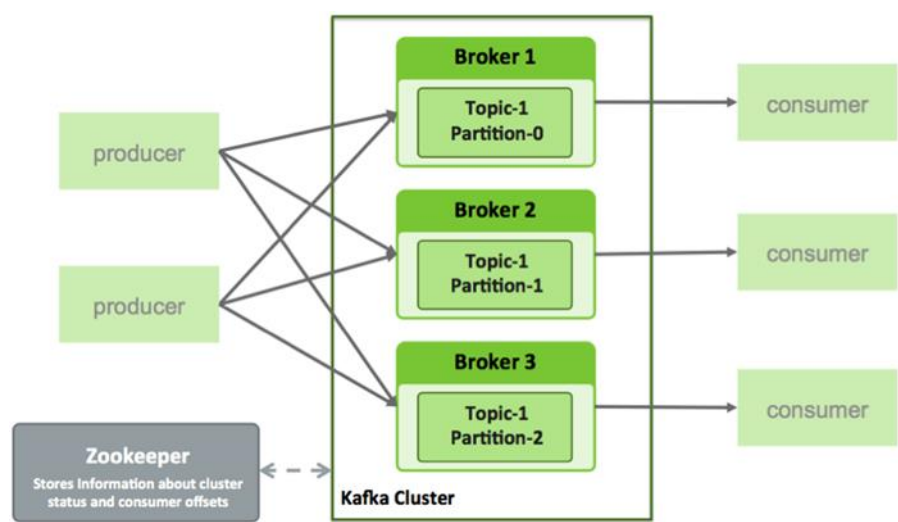


Figure 14 – Apache Kafka Logical Architecture.

Kafka architecture is a distributed system for optimized management of a high number of producer-consumer chains:

- each chain represents a stream of messages known as Topic (or feed);
- each Topic is managed by a certain number of machines belonging to the cluster (broker) and can be associated with an arbitrary number of Producers and Consumers.

The fundamental element in Kafka is the Topic representing a stream of messages; it can also be seen as a “topic” characterizing a group of messages.

In order to enhance system performance, each Topic can be further divided into partitions where each partition constitutes an ordered and unchanging sequence of messages.

Each message can be inserted into only one partition and receives a numerical ID (offset) uniquely identifying it within the partition.

Partitions are a mechanism used by Kafka to build scalability:

- each partition is associated to a file;
- all messages of a partition must be located on the same machine;
- data of different partitions can be located on different machines.

In this manner, a single log file can be divided into more than one physical file and distributed within a cluster thus reaching a considerable size.

Furthermore, each Topic is associated with a replication factor that obliges data to be replicated on more than one machine in order to ensure that the system will keep on working even if one of the machines fails.

4.3.2. Cyber/Physical Threats Correlator

The Cyber/Physical Threats Correlator is composed by the following main components, as depicted in Figure 15.

- Correlator Engine: component correlating data source events and identifying potential threats based on a list of rules set by skilled operators and the rules originating from a machine-learning expert system: the Machine Learning Module.
- Machine Learning (ML) Module: component based on the application of machine-learning algorithms allowing identification of standard behavioral models for the traffic originating from network data sources and the automatic generation of correlation rules.
- Interconnection Layer: dedicated to data exchange between data originating from network data sources and the results produced by the Correlator modules performing data analysis. The component is implemented using a distributed system for message management based on a producer/consumer model.

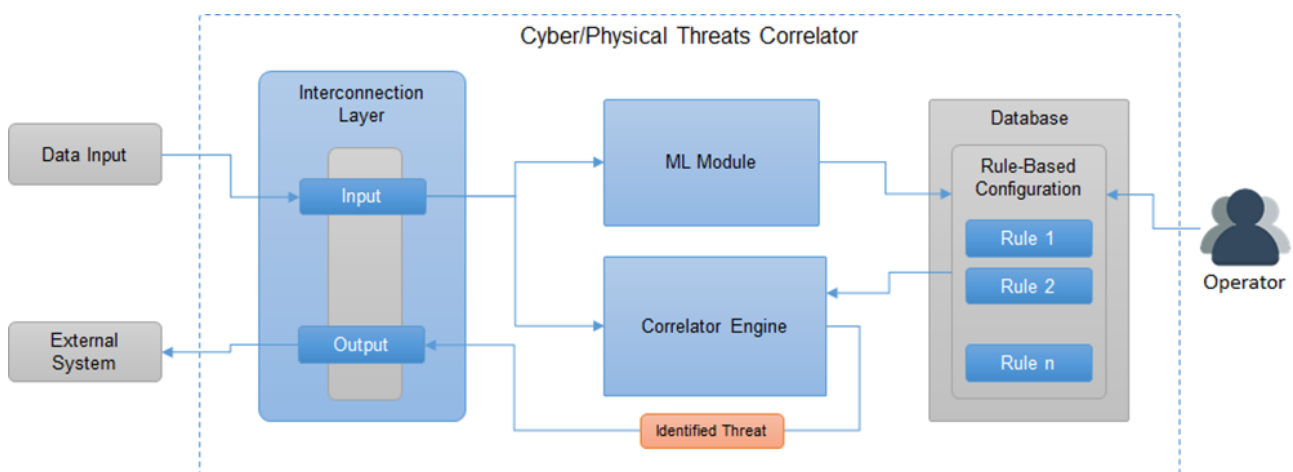


Figure 15 – Correlator components.

The main objective of the correlator is to correlate events and identify potential threats. Events are acquired through an interconnection layer that deals with managing input/output messages. All of the sources will pour log data into the messaging system in raw format, to be analysed and correlated. An example of RAW data of a SCADA source is reported in Figure 16.

```
May 4 15:18:33 192.168.1.1 n2osevents[0]: CEF:0| Networks|OS|17.5.2-1746F|VI:NEW-
NODE|New node appeared|3|app=other dvc=192.168.1.1 dvchost=nozomi-sg.local cs1=2.5
cs2=true cs1Label=Risk cs2Label=IsSecurity dmac=54:1f:15:1a:11:3c dpt=902 msg=New
other node 192.168.10.10 smac=01:0c:19:1e:1d:14 spt=60162 proto=TCP
start=1525447113163
```

Figure 16 – Raw data from a SCADA source.

All data acquired from the data sources will be subject to normalization, operation necessary to make the data comparable to each other. This operation allows the correlation of heterogeneous sources. The normalized data are sent to the previously described modules, i.e. the Correlator Engine and the ML Based Correlator.

The module that deals with identifying threats is the Correlator Engine (see Figure 17). To identify such threats the module must be provided with a list of well-defined rules, which are set:

- manually by skilled operators, or
- automatically by the ML Module. In particular, this module, through traffic analysis and the use of expert algorithms, generates deterministic rules with thresholds and parameters derived from the analysed behaviour.

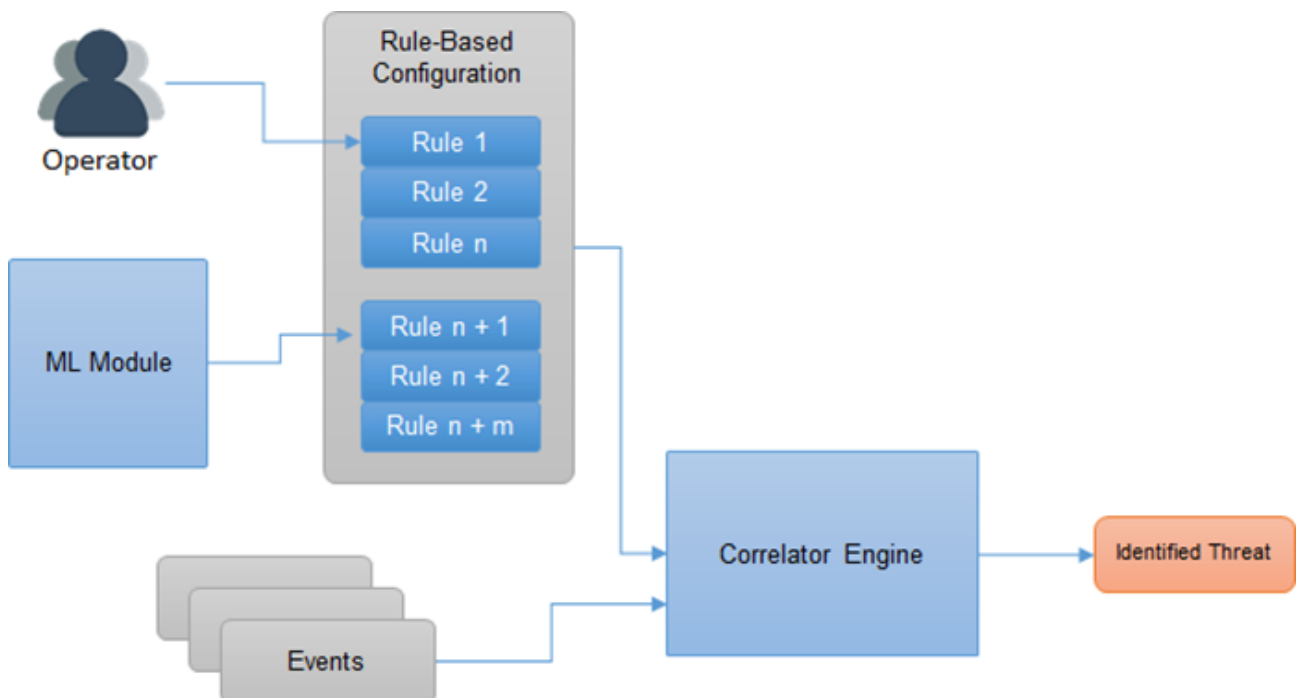


Figure 17 – Correlator main data flow.

The Correlator Engine, realized with the below described technologies, will always be running to identify when a rule matches. When the rules are updated, either manually by the operators or automatically by the expert module, the engine adapts itself to the changes in real time.

An example of a correlation rule is depicted in Figure 18 (details in section 4.3.2.1).

```
SELECT scadaEV1.msg AS scadaEvent1Msg, scadaEV2.msg AS scadaEvent2Msg
FROM PATTERN[every scadaEV1=SCADAEvent(signature='VI:NEW-ARP') ->
scadaEV2=SCADAEvent(signature='SIGN:DHCP-OPERATION')
WHERE timer:within(10 sec)]
```

Figure 18 – Example of a correlation rule.

The list of rules and real-time events are the inputs for the correlation engine. The output of the engine is a potential threat that the rule implements. In the example in Figure 19, a motor generates an alarm when two SCADA events, with particular characteristics, occur within a defined time window.

```
{
  "ts": 1544540170540,
  "ruleName": "Intrusion attempt",
  "ruleSeverity": "MEDIUM",
  "newEvents": [
    {
      "scadaEV1": {
        "signature ": "VI:NEW-ARP",
      },
      "scadaEV2": {
        "signature ": "SIGN:DHCP-OPERATION",
      }
    }
  ]
}
```

Figure 19 – Example of correlation.

The correlation shows:

- rule-related information (i.e., name, severity and timestamp of the match);
- information related to the correlation original events.

The alarm is normalized in a standard format and pushed into the message manager of the interconnection layer, so that it can be retrieved by all external systems. The detailed architecture is shown in Figure 20.

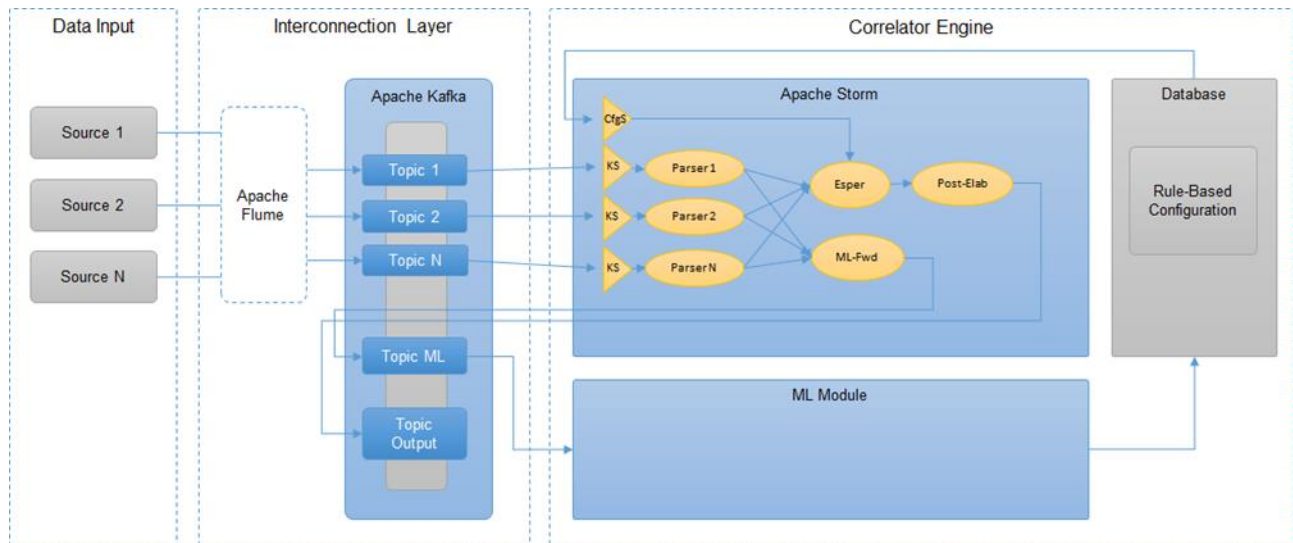


Figure 20 – Logical Architecture of the Cyber/Physical Threats Correlator.

4.3.2.1. Correlator Engine

The Correlator Engine is mainly based on Apache Storm and Esper technologies.

Apache Storm is a free open source software for distributed computing of real-time processes. The Storm architecture is of type Master-Slave:

- Nimbus, the master node, is tasked with distributing software code on the cluster, assigning tasks to the different machines and monitoring potential failures on the part of the tasks;
- the Supervisor slaves wait for the tasks to be assigned by Nimbus and can instruct workers to perform the assigned tasks;
- the work calculated by a single topology is divided and allotted to many workers distributed on different machines belonging to the same cluster.

Zookeeper supports coordination of the tasks among the various demons.

A topology is a process graph (Figure 21). Each node of the topology contains a piece of processing logic and the links between the nodes specify how data must be transferred from one node to another. A topology is always active as Storm is able to manage hardware failures and re-assign failed tasks to other available machines belonging to the cluster with no loss of data.

Storm provides the elementary commands required for construction of topologies:

- spout: a topology node capable of capturing an input stream or, in other words, a data source of tuples;
- bolt: a processing step (similar to a map of the map-reduce job) that:
 - receives input one tuple at a time;
 - processes it;
 - possibly emits one or more tuples which could generate new streams.

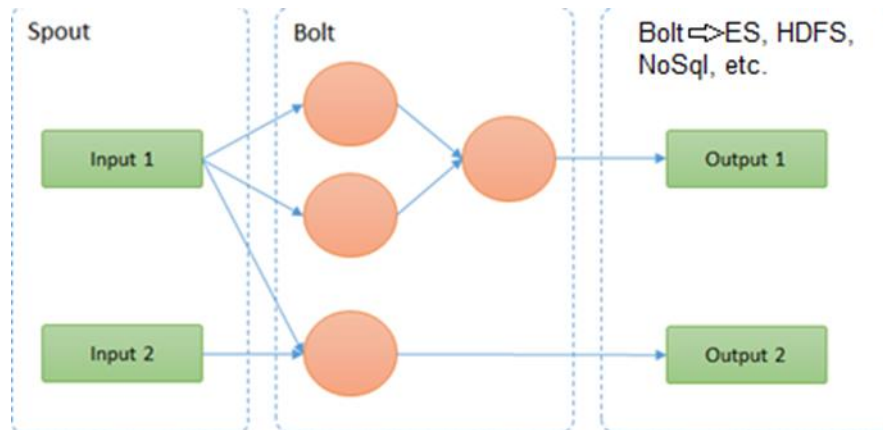


Figure 21 – Apache Storm: Topology Nodes.

When defining a topology, it's possible to associate a different degree of parallelism to each node so as to best adjust available resources based on the type of processing planned.

Data exchanged within a topology are called "tuples". The lifecycle of a tuple begins with a spout, the source of a tuple.

Whenever Storm requests a tuple from Spout:

- the tuple remains in the queue but assumes the "pending" state thus blocking elaboration by other processes;
- the tuple is assigned an "id" allowing the tracking of all messages generated within the topology during processing;
- Storm sends the tuple to all bolts specified in the topology and keeps track of all generated messages; based on the processing results, Storm sends to the spout that generated the tuple a message of type:
 - ack: so that the tuple will ultimately be eliminated from the queue,
 - fail: so that the tuple will be re-inserted into the queue to undergo re-processing.

Esper is a Complex Event Processing component able to perform Event Stream Processing. This feature allows real-time or quasi-real-time processing. The Esper engine operates in a different manner compared to a database management system. Instead of storing data and performing queries on the data stored, it allows applications to store their own queries and directly launch them on the data. The processing mode is continuous and a reply is in real-time whenever the conditions contained within the query are met.

Esper provides two principal methods for processing events and they are:

- Event pattern,
- Event stream query.

The first method is based on a language allowing specification of expression-based patterns for event matching. This event processing method, at the base of which is the implementation of a state machine, expects to receive event sequences or a combination of event sequences and allows their correlation based on time factors. On the contrary, the second method offers the possibility to define queries allowing filtering, aggregation and correlation (through join operators) as well as to analyze event streams. These queries follow the EPL (Event Processing Language) syntax. EPL is a

declarative language similar to SQL and by which it is possible to define CEP statements for the correct processing of an event stream.

EPL differs from SQL in that it uses the view concept as opposed to the table concept. These views represent the various operations required to structure and extract data from event streams. Note: once the queries are expressed (in EPL), they must be registered (at runtime) in the engine so that Esper can verify the results of the queries applied to the incoming events and possibly forward them in real-time to one or more subscribed listeners. This mode of operation allows, if necessary, a simple and swift modification to the already registered queries and offers the possibility re-process the events.

4.3.2.2. *Machine Learning Module*

The Machine Learning module will be implemented as a distributed module of the platform able to receive signals for the computation of new rules. Historical data retrieved from the past will be used as first baseline for the elaboration of the statistical procedure. Then this repository will be increased as telecommunication data will be collected.

The operation principle of the machine learning module is based on a periodical schedule of a statistical procedure which will analyse information included within data packets. Statistical data analysis will take into account trends and data distribution in order to compute metrics for the following rule-based evaluation. These metrics will also be related to the outputs of the descriptive statistics. Descriptive statistics provides great insight into the data in terms of count, mean, standard deviation, minimum value, percentiles. The usage of descriptive statistics as well as the correlation analysis of the data attributes will highlight interesting patterns to be translated in rules. To this end, one can give an example for the RESISTO project purposes. The machine learning module will consider information related to data traffic and will evaluate specific parameters which will be linked to well-known cyber attacks. For example, considering a scenario related to a Denial of Service (DoS) attack, metrics computed on the TCP protocol or number of requests sent to elements of the telecommunication infrastructure defined for validation test purposes, it will be modeled as parameters for the rules specified by the Rule-based correlator.

The statistical procedure will be implemented as software modules developed in R or Python languages. The core of the elaboration will be included within a generic software application and the mechanisms for the metric computation will be triggered by taking into account one of these approaches:

- a batch operation periodically scheduled by the application, and configured by means of a configuration file,
- a REST web service which will receive invocations from outside in order to schedule the computation on demand.

The statistical procedure will define metrics which will be used as parametric values for rule definition. Once the procedure will be completed, then the rules will be updated according the new computed values. The communication among the machine learning and rule based correlators will be implemented by means of either synchronous communication (e.g. web service interaction) or asynchronous communication based on publish-subscribe paradigm. This way, the Correlator Engine will receive new definitions and will update the internal rule list belonging to the complex event processor engine without stopping the current stream processing.

4.3.2.3. Interconnection Layer

The correlation module requires scalable and flexible interconnection instruments for the exchange of data between data sources and the modules collecting the results of the analyzed data. The technology used to implement this layer is Apache Kafka (see section 4.3.1) and, where necessary, Apache Flume as a supporting component.

The use of Apache Flume originates from the need to optimize data acquisition management and centralization. The architecture is based on the Agent concept representing an implementation of the producer/consumer model (Figure 22).

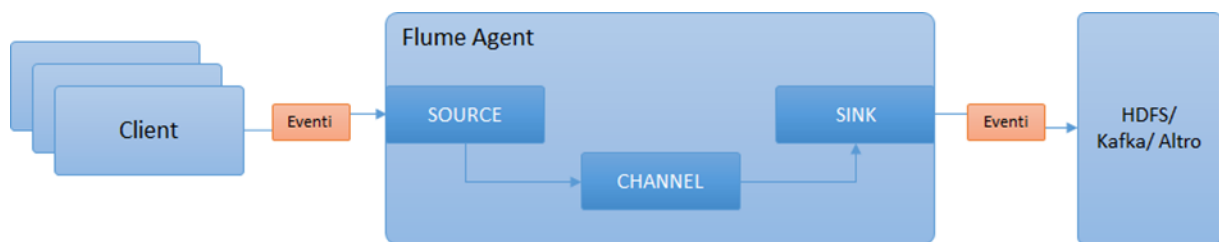


Figure 22 – Apache Flume architecture.

There are three fundamental concepts underlying this technology:

- **Events:** an event is data produced by an external source and forwarded to a final destination. From a network point of view, it is a byte-array composed of a header and a payload. The “external data” is inserted in the payload and is ignored by Flume. All information required for management and processing of events is accumulated in the header under the form of a key-value couple.
- **Client:** a Client is an entity that collects data, encapsulates it in events and sends it to one or more Agents. It could be a Flume client that collects data produced by external applications according to standard formats (for example, Apache, Log4j) or, if no Flume client is available, it could be developed ad hoc.
- **Agent:** Agent is the central Flume component: it receives events generated by an external client, deposits them in an internal queue and sends them to one or more final destinations.

Like Kafka, Apache Flume is based on the producer-consumer model and possesses a similar architecture.

4.3.3. Risk (Impact) Predictor – Dynamic Risk Analysis Tool

This section describes the Risk Predictor architecture, commonly adopted in Critical Infrastructure Protection (CIP) projects, in order to provide a new type of Dynamic Risk Assessment tool. Such a tool is based on CISI Apro (Critical Infrastructure Simulation by Interdependent Agents) software engine, which is able to calculate complex cascading effects, taking into account (inter)dependencies and faults propagation among the involved complex systems.

As mentioned in section 5.2 of deliverable D2.4, modelling complex interdependent systems, using the Mixed-Holistic-Reductionist (MHR) approach, is a prerequisite to produce an effective Risk

Predictor tool. Once modelled the involved scenario, with CISIAPro [Ref3] it is possible to implement the MHR methodology.

CISIAPro is an Agent-Base simulation software and it is mainly composed by two modules. The first one is the off-line tool known as CISIAPro in which is possible design and implements complex and highly interdependent scenarios. While the second one is the on-line tool called CISIARun which represents the real engine at the base of the Risk Predictor module. This engine will be integrated in the RESISTO architecture as in Figure 23.

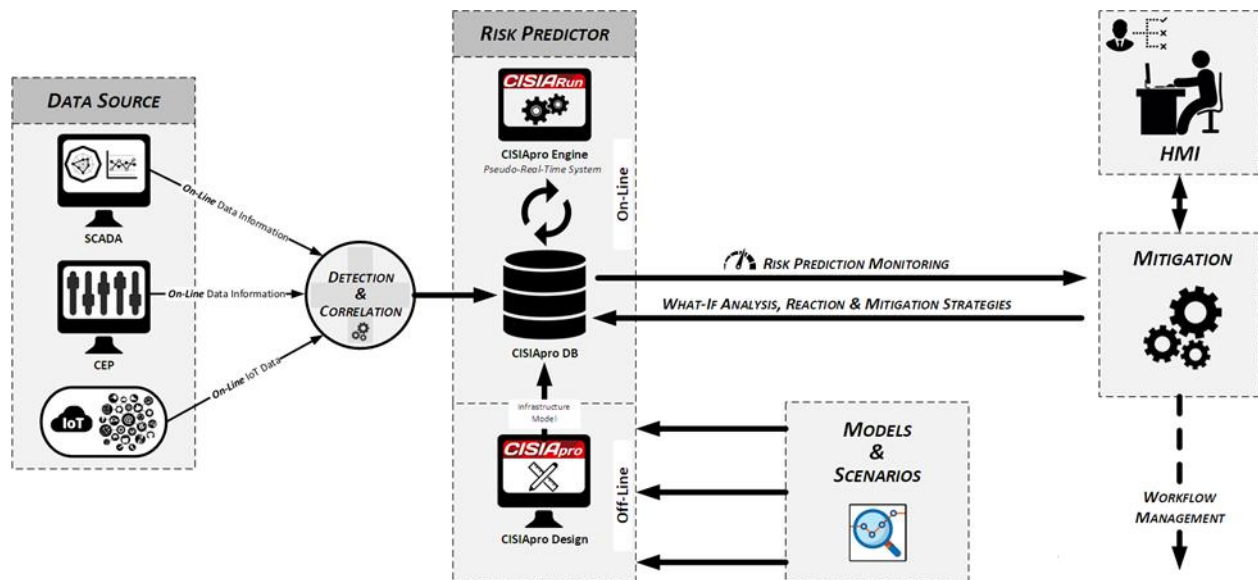


Figure 23 – Risk Predictor Architecture

CISIAPro is a software platform based on a database-centric architecture in which the database plays a crucial role. This means a centralized asynchronous design that allows a good modularity and scalability where each element of the informatics infrastructure interfaces, independently, with the centralized database (DB) in order to get the last actualized data from the field (e.g. SCADA Systems), Complex Event Processing and generic IoT (Internet of Things) data systems.

From this point of view, CISIAPro engine does not only analyze actual situation and calculate the risk projected in the possible near future but, first, it plays the important role of Hybrid Risk Evaluation Tool. Hybrid because it is able to get information of different natures (sensor and data acquisition and complex event processing systems) and translating them in operational levels of resources, faults or services for the entities introduced in the critical infrastructure model.

With the proposed architecture, through CISIAPro modelling software, it is possible to dynamically change the interdependencies model and plugin other modules in order to have a pseudo-real-time scalable and flexible system, which can be changed at any time.

Figure 24 shows the database structure. The DB stores the information needed for the representation of several Critical Infrastructures, such as:

- Each entity is a specific instance of an entity type;
- Each entity has a status made of variables with values;
- Each entity has ports for exchanging resources;

- Each resource is associated with a MHR layer/net;
- Each layer has proper interdependencies;
- Each interconnection is made of a couple of ports, associated to two entities.

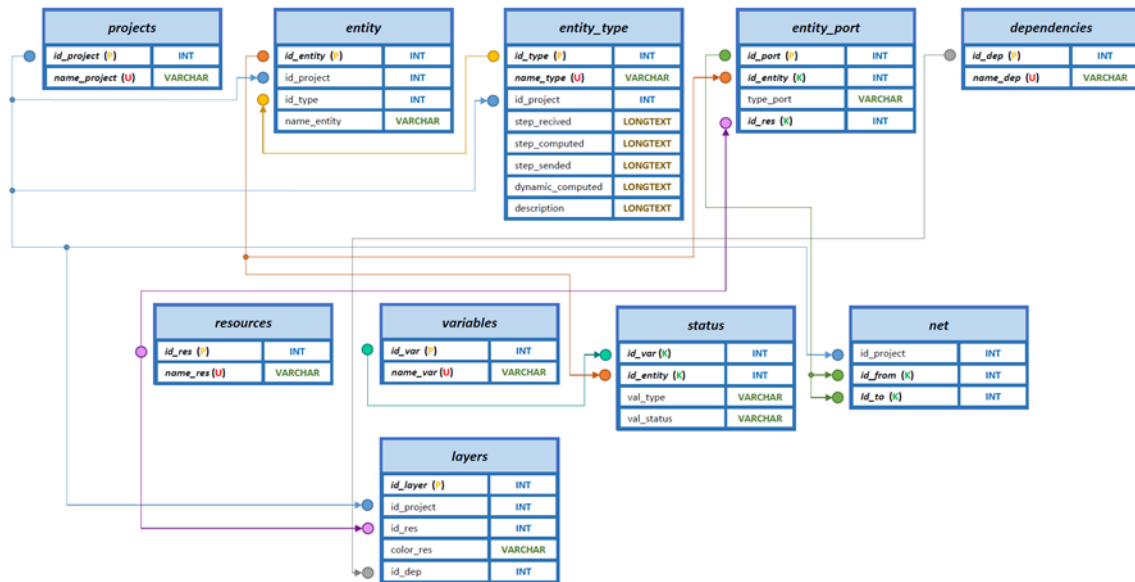


Figure 24 – CISIApro SQL input data structure.

Outputs of CISIApro are stored in a different database with specific features, see Figure 25, such as the record time-stamp in terms of date, time and milliseconds. In this way, any downstream module can retrieve data regarding the latest actualized critical situation in the modelled scenario. Adjacency matrices which represent interdependencies existing between entities are generated during the design phase. During the simulation, the matrices are represented as queue data structures to speed up computations.

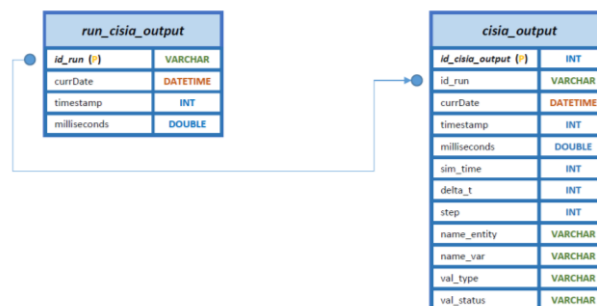


Figure 25 – CISIApro SQL output data structure.

It should be noted that CISIApro has introduced efficient ways to model, execute and debug simulations and cascading effects. In particular, an intuitive Graphical User Interface (Figure 26) is provided to create entities and connect them in easy way.



Figure 26 – CISIApro user interface.

Below are presented some brief descriptions regarding the main modules provided by CISIApro design tool to exploit the potential of proposed modelling techniques.

4.3.3.1. Layers & Resources Module

Thanks to the *Layers & Resources* module (Figure 27) it is possible to instantiate all the required layers in a critical infrastructures scenario model. It is the first step for the simulation implementation. Assign a specific resource to a corresponding dimension also gives us a deeper awareness with respect to the nature of the managed information.

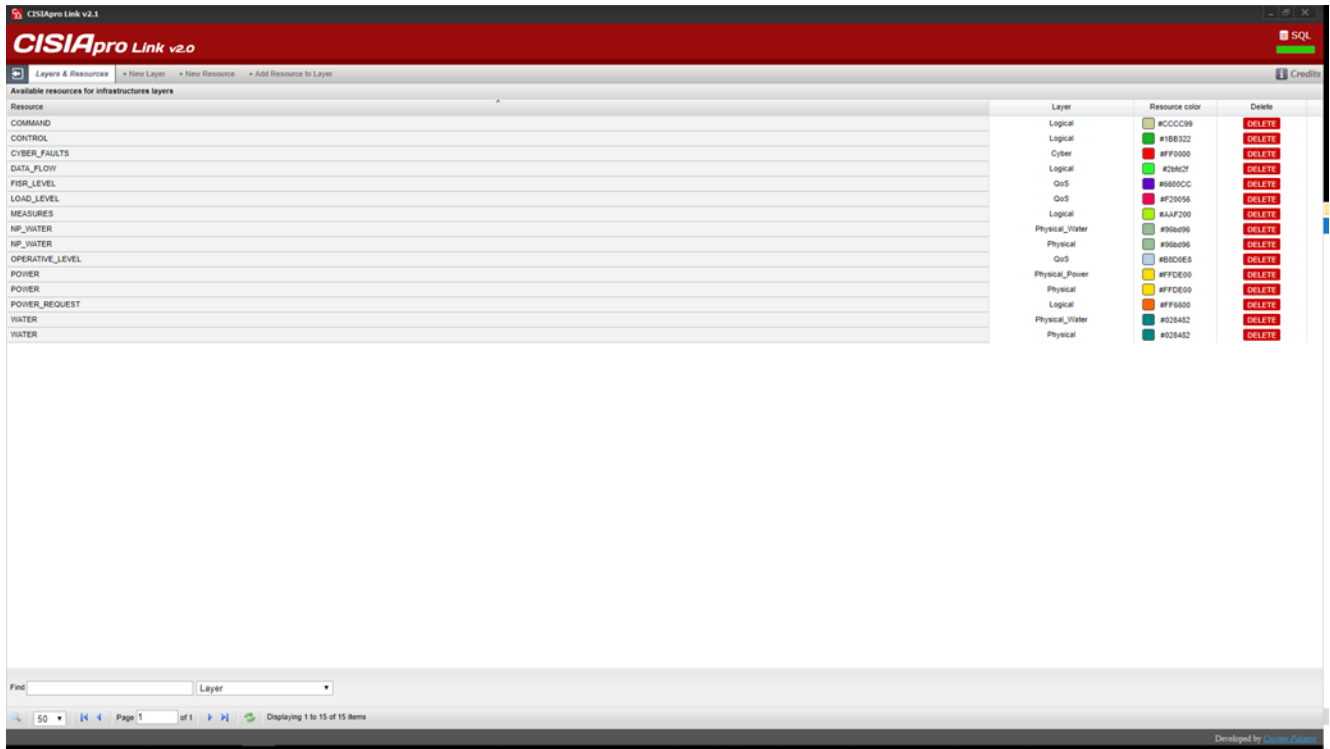


Figure 27 – CISIApro Module: Layers & Resources.

4.3.3.2. Entity Maker Module

In the *Entity Maker* module (Figure 28), using the integrated PHP code editor, it is possible to instantiate and programming behaviors for each considered entity class. Once completed this step, the introduction and duplication in the design phase will be allowed. Each entity class is composed of four modules that are executed, several times, during the simulation run:

- 1) RECEIVED, which evaluates the received resources and faults;
- 2) DYNAMIC COMPUTED, which implements dynamic evolution;
- 3) INSTANT COMPUTED, which implements instantaneous evolution;
- 4) SENT, which evaluates the resources that are sent to the downstream entities.

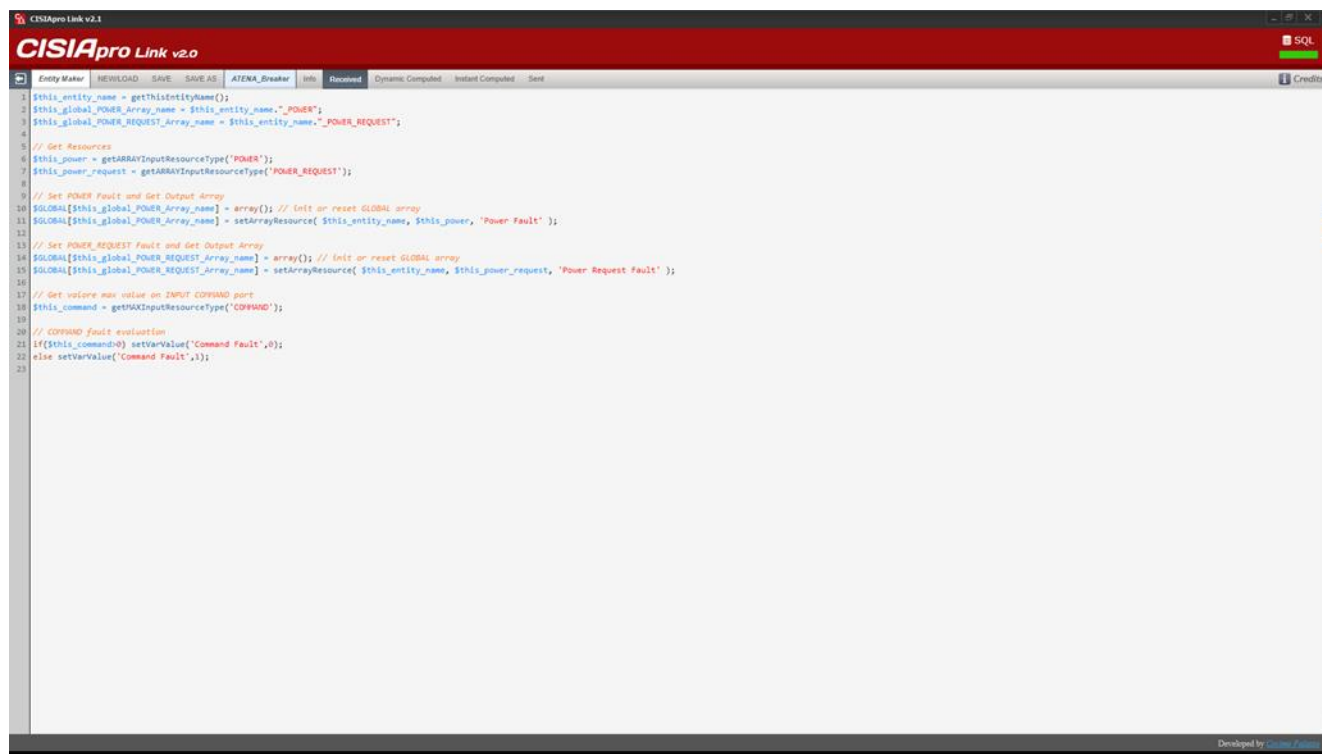


Figure 28 – CISIApro Module: Entity Maker.

4.3.3.3. Modeler Module

The *Modeler* module (Figure 29) is designed in order to improve the productivity in modelling Critical Infrastructures scenarios. Thanks to a usable graphic interface and a Drag & Drop system, it is an easier operation to introduce entities and create (inter)dependencies.

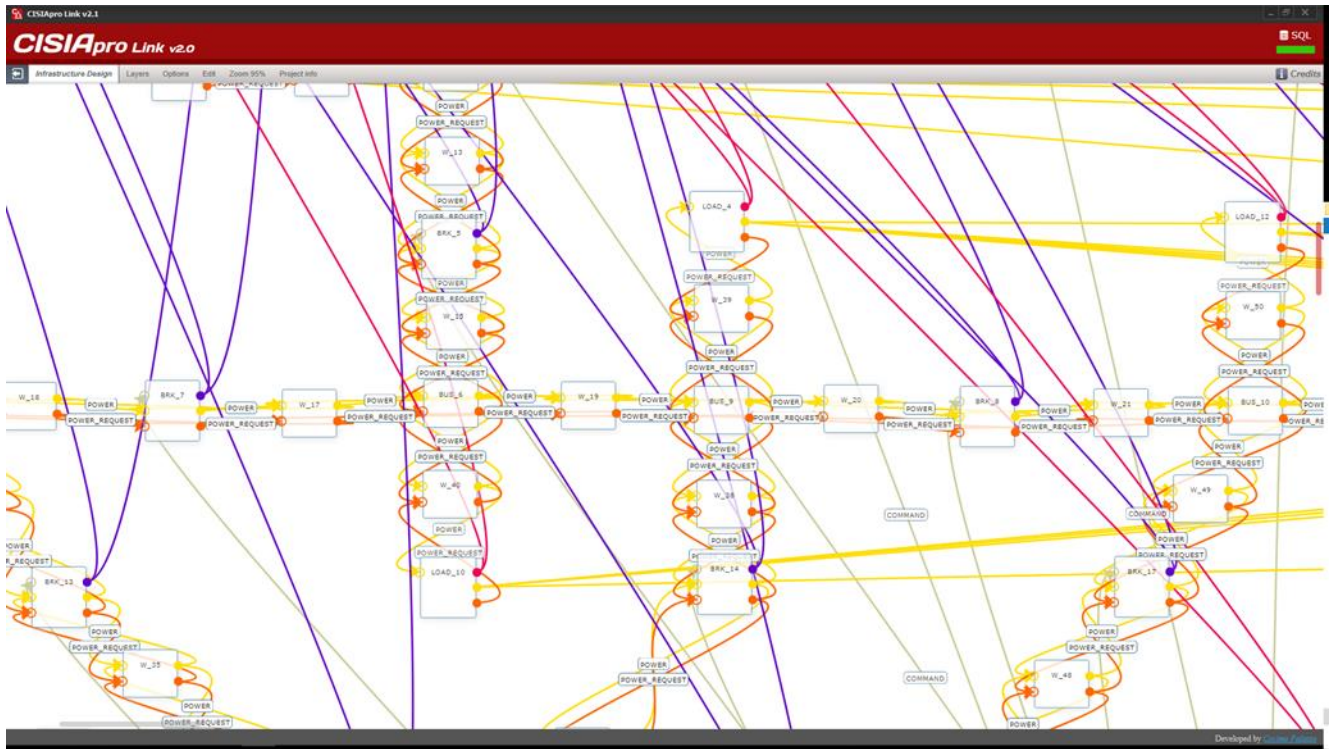


Figure 29 – CISIApro Module: Modeler.

4.3.3.4. State Variables Module

As previously mentioned, with CISIApro simulation, defining Input and Output is not required. This is possible because they are calculated, instant by instant, during the simulation time, with respect to entity **state variables** and especially evaluating **operational levels** related to each modelled element. In *State Variables* module (Figure 30) indeed it is possible to set the initial state for every variable that is part of the simulation.

Variables & entity faults	Variables	Data type	Status	Change Status	Delete Status
Backup_Centre	Layer	OTHER	SCADA_GRID	CHANGE	DELETE
Backup_Centre	Send Control	NUMERIC	0	CHANGE	DELETE
Backup_Centre	MTM Attack	NUMERIC	0	CHANGE	DELETE
Backup_Centre	INFECTION Attack	NUMERIC	0	CHANGE	DELETE
Backup_Centre	SCAN Attack	NUMERIC	0	CHANGE	DELETE
Backup_Centre	DOS Attack	NUMERIC	0	CHANGE	DELETE
Backup_Centre	Cyber Fault	NUMERIC	0	CHANGE	DELETE
Backup_Centre	Mechanical Fault	NUMERIC	0	CHANGE	DELETE
Backup_Centre	Operative Level	NUMERIC	1	CHANGE	DELETE
Breaker_Performance	BRK Level	NUMERIC	0	CHANGE	DELETE
Breaker_Performance	Operative Value	NUMERIC	1	CHANGE	DELETE
BRK_1	Strategic Value	NUMERIC	1	CHANGE	DELETE
BRK_1	Power Request Fault	NUMERIC	1	CHANGE	DELETE
BRK_1	Layer	OTHER	GRID	CHANGE	DELETE
BRK_1	Switch ON_OFF	NUMERIC	1	CHANGE	DELETE
BRK_1	Power Fault	NUMERIC	1	CHANGE	DELETE
BRK_1	Command Fault	NUMERIC	1	CHANGE	DELETE
BRK_1	Mechanical Fault	NUMERIC	0	CHANGE	DELETE
BRK_10	Operative Level	NUMERIC	1	CHANGE	DELETE
BRK_10	Switch ON_OFF	NUMERIC	1	CHANGE	DELETE
BRK_10	Power Fault	NUMERIC	1	CHANGE	DELETE
BRK_10	Command Fault	NUMERIC	1	CHANGE	DELETE
BRK_10	Mechanical Fault	NUMERIC	0	CHANGE	DELETE
BRK_10	Power Request Fault	NUMERIC	1	CHANGE	DELETE
BRK_10	Layer	OTHER	GRID	CHANGE	DELETE
BRK_10	Operative Level	NUMERIC	1	CHANGE	DELETE
BRK_10	Strategic Value	NUMERIC	1	CHANGE	DELETE
BRK_11	Command Fault	NUMERIC	1	CHANGE	DELETE
BRK_11	Power Fault	NUMERIC	1	CHANGE	DELETE
BRK_11	Strategic Value	NUMERIC	1	CHANGE	DELETE
BRK_11	Mechanical Fault	NUMERIC	0	CHANGE	DELETE
BRK_11	Power Request Fault	NUMERIC	1	CHANGE	DELETE
BRK_11	Layer	OTHER	GRID	CHANGE	DELETE
BRK_11	Operative Level	NUMERIC	1	CHANGE	DELETE
BRK_11	Switch ON_OFF	NUMERIC	1	CHANGE	DELETE
BRK_12	Power Request Fault	NUMERIC	1	CHANGE	DELETE
BRK_12	Switch ON_OFF	NUMERIC	1	CHANGE	DELETE

Figure 30 – CISIApro Module: State Variables.

4.3.3.5. Link States Module

Link States is the most recent module tool introduced in CISIApro software for RESISTO scenario model needs (Figure 31). Such module was designed in order to be compliant to model in which dynamic links are required. A dynamic link is defined as a link that connects two entities to each other, which could change its state during different simulation. For instance, it allows to model scenarios, like transportation infrastructures or telecommunication networks, where an entity may represent an element of the system and links represent multiple available paths (e.g. SDN dynamic routing). Through this mechanism it will be possible to instantiate all the multiple connection, among the involved entities, activating only one of them at a time.

Entity From	Entity To	Resource	State	On/Off
Backup_Centre	MCPT_2	CYBER_FAULTS	1	On
Backup_Centre	MCPT_1	CYBER_FAULTS	1	On
Backup_Centre	MCPT_1	OPERATIVE_LEVEL	1	On
Backup_Centre	MCPT_2	CONTROL	1	On
Backup_Centre	MCPT_1	CONTROL	1	On
Backup_Centre	MCPT_2	OPERATIVE_LEVEL	1	On
BRK_1	FISIR_Performance	FISIR_LEVEL	1	On
BRK_1	W_2	POWER_REQUEST	1	On
BRK_1	Breaker_Performance	FISIR_LEVEL	1	On
BRK_1	W_2	POWER	1	On
BRK_1	W_1	POWER	1	On
BRK_1	W_1	POWER_REQUEST	1	On
BRK_10	Breaker_Performance	FISIR_LEVEL	1	On
BRK_10	W_23	POWER_REQUEST	1	On
BRK_10	W_23	POWER	1	On
BRK_10	W_24	POWER	1	On
BRK_10	W_24	POWER_REQUEST	1	On
BRK_10	FISIR_Performance	FISIR_LEVEL	1	On
BRK_11	W_26	POWER	1	On
BRK_11	W_25	POWER	1	On
BRK_11	Breaker_Performance	FISIR_LEVEL	1	On
BRK_11	W_26	POWER_REQUEST	1	On
BRK_11	W_25	POWER_REQUEST	1	On
BRK_11	FISIR_Performance	FISIR_LEVEL	1	On
BRK_12	W_26	POWER	1	On
BRK_12	Breaker_Performance	FISIR_LEVEL	1	On
BRK_12	W_27	POWER	1	On
BRK_12	W_27	POWER_REQUEST	1	On
BRK_12	FISIR_Performance	FISIR_LEVEL	1	On

Figure 31 – CISIApro Module: Link State.

4.3.3.6. Simulation Module

Thanks to the *Simulation* module (Figure 32) it is possible to debug a simulated scenario, validating its results and performances before proceeding to the *Risk Predictor* production phase.

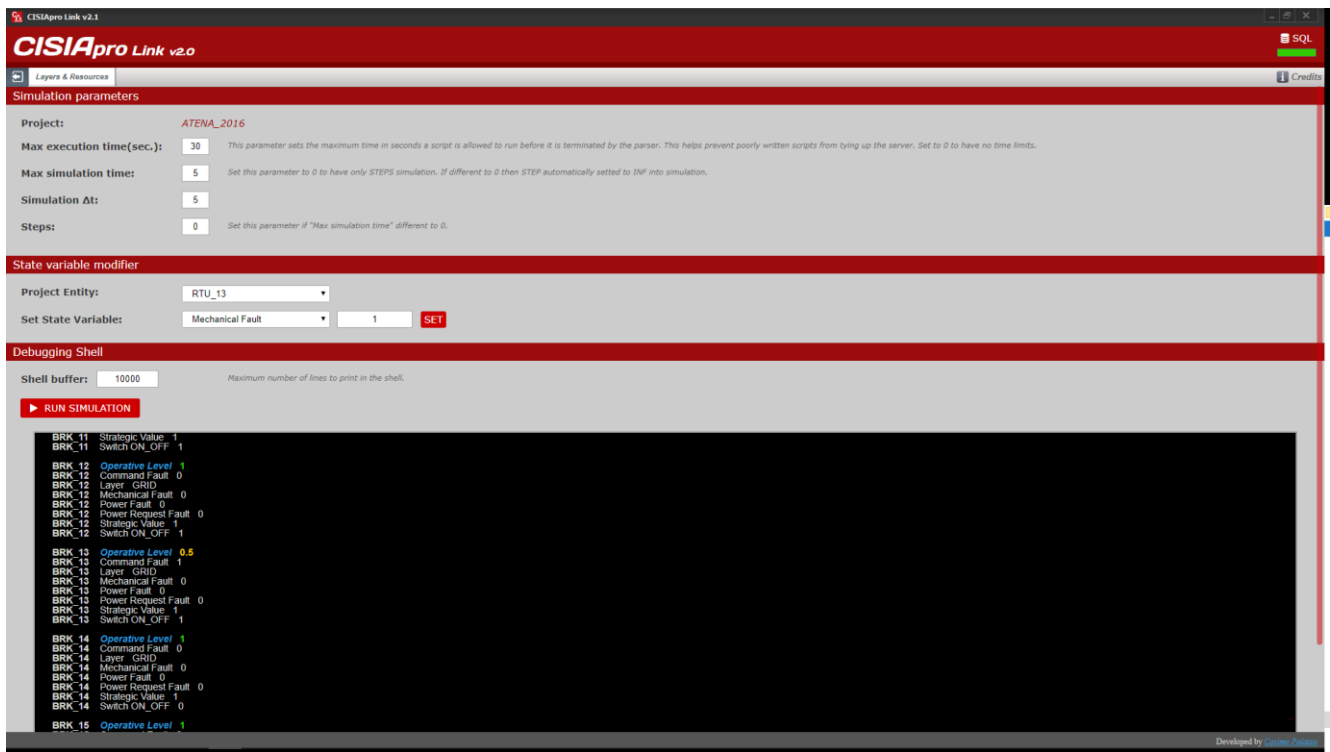


Figure 32 – CISIApro Module: Simulation.

4.3.3.7. CISIArun – the CISIApro on-line engine.

Usually Risk Analysis and Business Continuity are mainly focused on prevention and consequences mitigation. In RESISTO case study a pseudo-real-time performances monitoring is also considered in order to have an ongoing decision support system.

In RESISTO project CISIApro engine is part of the **Short Term Control Loop**. The **Short Term Control Loop** continuously monitors the CI system and its cyber-physical state, in order to **detect** the presence of failures/attacks in a real-time environment. To defend against such attacks, the Short Term Control Loop shall compute domain specific mitigation actions and propose them to the RESISTO operator, in order to enforce them into the underlying plant and preventively mitigate consequences on the service provision. The main aim of the on-line loop is to increase the awareness of the operator, displaying possible impact/consequences and exposure to risk with respect to the actual adverse events on the physical infrastructure and on the telecommunication network.

It is important to understand the primary role played by **Risk Predictor** (CISIApro On-Line engine) together with the **Mitigation Module**. First of all CISIApro On-Line engine provides an impact evaluation of detected anomaly. In order to mitigate the effects the Decision Maker, also supported by a Workflow Manager, can choose among different sequences of possible reaction strategies to send to the Risk Predictor module. **Risk Predictor**, in turn, starting from actualized scenarios and QoS (Quality of Service) levels of involved devices, simulates **What-If** scenarios to provide useful information for the Decision Maker with respect to 'forthcoming' critical situations.

Thanks to **CISIApro on-line engine** is also possible to create, as already demonstrated in URANIUM [Ref4] and H2020 ATENA [Ref5] European projects, web-based synoptic platforms. Such projects

made evident that it is important to provide an intuitive user experience in order to simplify and speed up decision processes in emergency situations. For instance, in URANIUM project (Figure 33) it was developed a GIS (Geographic Information System) demonstrating how it is possible to carry out a **What-If analysis** taking into account different emergencies procedures and considering available civil protection resources at a specific point in time.



Figure 33 – URANIUM platform civil protection panel.

Another important example is represented by ATENA H2020 synoptic platform (Figure 34). In this case, a What-If analysis procedure was exploited in order to test possible consequences of adopted Electrical Grid reconfiguration with respect to all interconnected critical infrastructure (Water distribution, Gas network and involved SCADA systems).

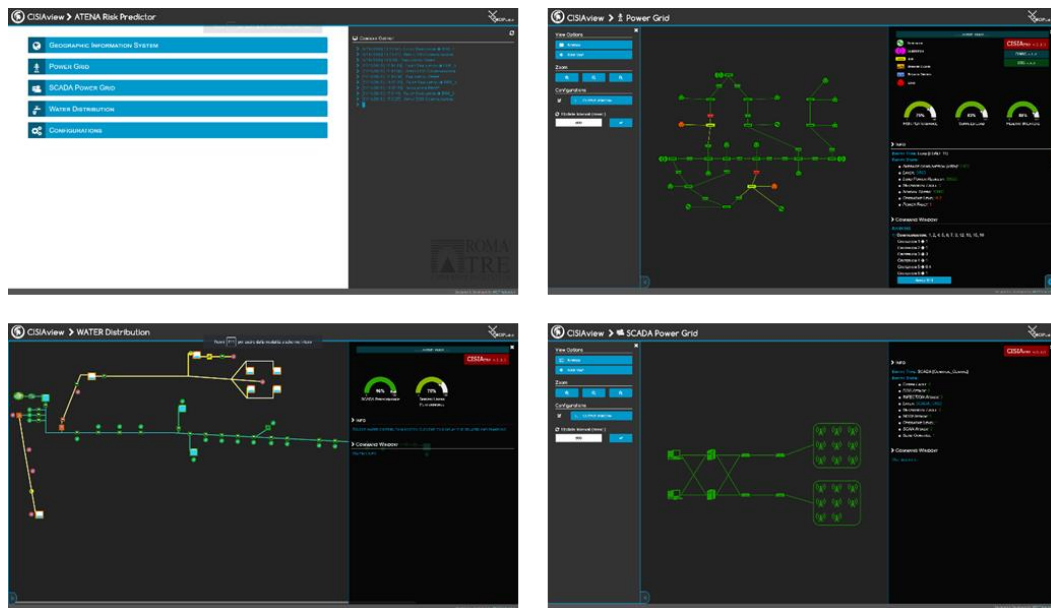


Figure 34 – ATENA Platform.

4.3.3.8. List of planned interfaces for the Risk Predictor Module in RESISTO Architecture

No.	Acronym	From	To	Short Description
1	adp-rp	Correlator	Risk Predictor	The Detector/Correlator exchange the normalized data source to the Risk Predictor. The RP exploits the 'current' status of the cyber-physical infrastructure for assessing the impact/consequences and possible exposure to risk on other involved infrastructures and services.
2	adp-mm	Risk Predictor	Mitigation Module	The Mitigation module exchange data with the Risk Predictor module. The Mitigation module will also able to provides, as output, the What-If Analysis, Reaction and possible Mitigation strategies and Workflow Management.

Table 4 – Planned interfaces for the Risk Predictor Module.

4.3.4. Mitigation Module

The Mitigation Module of the Short Term Control Loop deals with response and mitigation, combining a software module (the Workflow engine) and a decision support system, as depicted in Figure 35. In particular, on the basis of target value and quantitative analysis performed by the Risk Predictor, the Decision Maker selects the best reaction/mitigation strategy to mitigate the risks or to recover from a damaged situation. After the operator's choice, the Workflow engine defines the specific actions to be enforced.



Figure 35 – Mitigation Module of the Short Term Control Loop.

The Workflow engine included in the RESISTO platform is an extremely effective tool for managing critical infrastructure security. It is a Business Process Model (BPM) engine for the configuration and execution of automatic or semi-automatic processes, consisting of sequences of actions and reactions, which can be triggered by a defined event. Given a certain alarm/event, it permits selecting and executing the most appropriate workflow, i.e. a conditional sequence of tasks. Some of the tasks may foresee the direct execution by the security operator, in other words may not be executed automatically. A possible list of activities is described below:

- carry out a procedural action (e.g., alert a reaction team with an emergency message),
- drive a physical actuator (e.g., lock a physical gate),
- perform a complex action on the Communication Network (e.g., activate a Virtual Network Security Function, isolate a faulty or attacked component, reconfigure a part of the network, disable a 5G slice, etc.).

A graphical user interface allows the introduction of Security Operational Procedures (SOPs) that implement the security plan for a specific infrastructure. This way, the system can ensure that the actions carried out in response to an alarm event are always connected to a coded process related to that specific event.

Some key features are:

- graphic design of the workflow,
- basic rule definition (Boolean operators, time, location, priority, etc.),
- enhanced rules (model-based relationship).

The workflow execution is carried out via Activiti, an open-source workflow engine written in Java that can execute business processes described in standard BPMN (Business Process Model and Notation) 2.0. The workflow is represented by an xml file, which is managed by the Activiti engine through its deployment in the H2 database. After that, the workflow can be executed (Figure 36).



Figure 36 – Activiti Engine.

Activiti, in accordance to BPMN standard, performs the workflow execution on the basis of two concepts:

- task,
- process.

A BPMN task is an atomic activity which represents work that is not broken down. On the contrary, process represents work that is broken down to a finer level of detail (Figure 37).



Figure 37 – BPMN task and process.

You draw a task when the work in the process cannot be broken down to a finer level of detail. For instance, sending an email or requesting the operator to insert some data may be tasks.

You use a process when you want to model the internal details of work in a lower-level process diagram. The process defines the timing for task execution. It may contain a block of parallel tasks, alternative blocks to be executed depending on session variables and so on. An example of process and task is depicted in Figure 38.

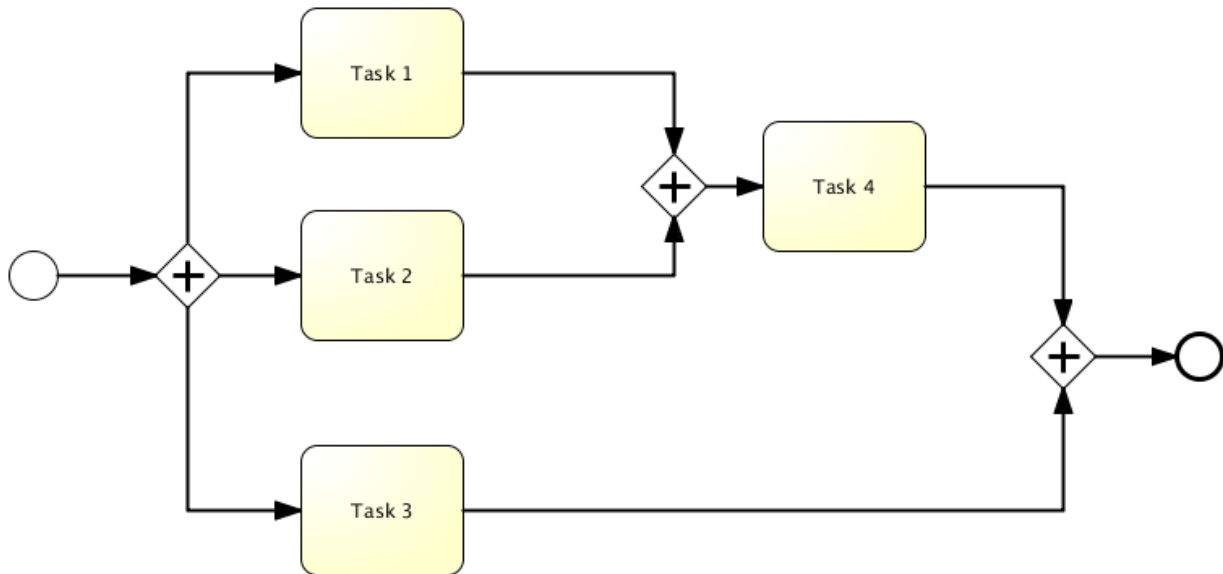


Figure 38 – Example of process and task.

At run time, the Workflow Manager, which is a Mule application, receives the alarm trigger for activating a specific sequence of tasks on a certain ActiveMQ queue. The workflow is executed by the Activiti engine. The alarm lifecycle (New, In progress, Closed) is instead managed through the publication of its state updates on the corresponding ActiveMQ topic.

4.3.5. Orchestration Controller

In this section, the main features of the Orchestration Controller are reported. As described in the Description of Action - part B [Ref1], the Orchestration Module is the subsystem that implements complex security functions and services related to the Communication Network, as requested by the security policies and workflows activated by the Mitigation Module. Among them we cite: activation of a Virtual Network Security Function, isolation of a faulty or attacked component, reconfiguration of a part of the network, disabling a 5G slice. This may require:

- scaling up or out individual functions;
- restructuring a service graph;
- modifying the placement of functions inside an actual network;
- rerouting traffic to or between different instances of such functions,
- adjudicating resources between competing services;
- lifecycle management of individual functions.

The Orchestration Controller is built around the concept of Software Defined Security (SDS) taking advantage of the Network Function Virtualization (NFV) and Software Defined Networking (SDN) paradigms of the underlying communication network.

At this aim it basically manages the cyber physical resources needed to apply the selected countermeasures, and more in general, the selected security policies. Its role is to build complex

security functions and services from less complex/primitive security mechanisms/functions acting on physical resources (i.e. network physical equipment) as well as Virtual Network Functions in a NFV/5G perspective. In this process, the orchestrator shall consider service specific requirements, in terms of Authenticity, Integrity, Confidentiality, etc. This is done through the entire lifecycle of a function/service, i.e. deployment, operation, monitoring and termination.

Also, it can be considered an umbrella function for e2e (end-to-end) service/network slice management when cross-tenant orchestration (Inter-slice Orchestration, Umbrella and Service Management) are needed. In case of multi-domain orchestration, it can be used for the automated management of services and resources in multi-technology environments and multi-operator environments which include operation across legal operational boundaries.

The Orchestration Controller supports:

- a) TLC infrastructure not equipped with a security mitigation and reaction system as well as
- b) TLC infrastructures that have their own mitigation and network orchestration system.

In order to support a large variety of TLC infrastructures with their own mitigation and network orchestration system the Orchestration Controller includes an adaptation layer specifically designed for allowing the interaction between the Controller and the other RESISTO modules and the specific TLC infrastructure.

The Orchestrator is based on the following modules:

- 1) Orchestrator Central Unit (OCU),
- 2) Network Controller,
- 3) Orchestrator Operator Client,
- 4) Orchestrator Adaptation Layer.

In case a) only the modules 1, 2, and 3 are deployed while in case b) only modules 1 and 4 are in use.

The Orchestration Controller design is aimed at:

- 1) allowing the system to have the desired behavior according to the current security level and security policies and
- 2) exploiting the potential of an SDN-based infrastructure.

The policies are implemented through an ad-hoc designed high-level firewall to filter command towards the network sub-systems or adapters. In more details, during an attack, the Orchestrator Central Unit can force a network configuration that allows delivery or drop-off of messages to subsystems depending on the reputation and trustiness of the message source and of the vulnerabilities of destination node. The reputation and trustiness can be set a priori (according to the internal policies of the TLC system) or can be modified during the time (according to the node-system behavior, or to the information received by the security probes about the possible corruption of the network/source). For example, during an attack, the operator may be the only authorized source of commands for the operating system.

A simplified implementation of this framework consists in exploiting two lists: WHITE LIST and BLACK LIST of components.

In this case, a typical operation sequence can be the following:

The OCU verifies the source/destination (src/dst) of messages received leading to an action to be taken. Different behaviors are possible:

- 1) If src/dst is in BLACK LIST the message is blocked.
- 2) If src/dst is in WHITE LIST the message is delivered.
- 3) If src/dst is not in WHITE LIST:
 - a. if system is in normal alert-state the message is delivered,
 - b. if system is in critical state the message is blocked.
 - c. An alert is sent to the user that may choose to change the status of src/dst to BLACK LIST or WHITE LIST.

As SDN manager, the Orchestrator Controller handles the SDN Network configuration and reaction to threats. In particular:

- the OCU receives from Asset Manager the devices list and the links among them (both physical and logical),
- the list is forwarded to the network controller that will provide to generate the flow rules to allow traffic between the authorized devices,
- the OCU verifies the correspondence with the device and link in the operating system,
- the OCU reconfigures the network in order to react to link failure or to exclude a device that has been inserted in the BLACK LIST.

From an implementation point of view, the Orchestrator Central Unit communicates with other RESISTO tools through a Message Broker.

4.3.6. Emergency Warning Communication

An Emergency Warning Communication Function shall be designed to send targeted alerts and/or informational instant messages to specific categories of users such as rescue teams or security officers that will be physically present on specific target areas (e.g. group of mobile cells, geographical perimeter, etc.). The Decision Maker policy implementation will resort to the Emergency Warning Communication Function service for this purpose when events like natural disasters, physical or cyber-attacks occur.

The app will deliver the information to the rescue teams in the location where events are actually happening and will be able to collect location information also from the terminal itself.

Specific categories of users (such as rescue teams, security officers, etc.) based on specific target areas will be defined as users of the RESISTO platform and will receive relevant information when events like natural disasters, physical or cyber-attacks occur. The function can be integrated in 5G networks as well as in existing telecommunications networks and it can be made available “as a Service” also to specific public safety agencies.

The implementation will include a server and an Android application. The server will expose an interface towards the other modules of RESISTO framework that need to communicate information concerning a physical-cyber attack to the intervention team that operates where the telecom infrastructure is located. The rescue team will leverage on the application information, both textual and visual. In particular, the position of points of interests or of the other team members will be collected and visualized. The app will be available on Android smartphones, or any other Android device.

Communication with populations in the event of an attack to a critical infrastructure integrating with use of a version of the app for general use where location information available from the devices (eg. GPS) are used.

The Emergency Warning Communication Function (EWCF) server will be connected with the RESISTO platform Mitigation module (operator and/or Workflow Manager) in order to receive messages that need to be dispatched to different teams or neighboring population.

Coordinates collected by the smartphone device will be exchanged with the server that will relay them to the other members of the team, in order for any of the team members to be aware of the position of the other members in case of coordinated actions which for instance are typical of natural disaster events. The application can be easily extended to collect and share with the team information regarding specific sensors other than the app user locations.

EWCF server collects GPS coordinates but in principle can be extended to collect other data from the terminal if available through sensors connected to it such as IoT collection platforms.

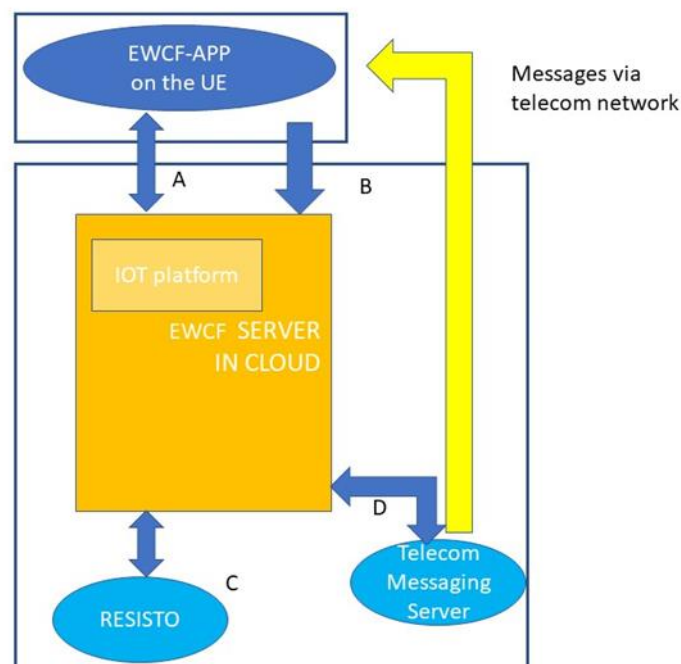


Figure 39 – EWC function architecture.

Figure 39 depicts the EWCF architecture and its interfaces:

- C interface will receive messages from RESISTO that need to be delivered to the team regarding targeted alerts and/or informational instant messages to the users of the EWCF-APP. C interface provides a service based REST API (Application Program Interface) to RESISTO to make requests to the EWC service.
- A and B interfaces shall be used to collect location information and other information from sensors in field.
- D interface (as an alternative to A) can be used to send messages via standard protocols using available telecom messaging capabilities (e.g. SMS, IMS) to target groups.

4.4. Cockpit

The RESISTO platform has not been conceived as an old-style monolithic all-in-one system but rather as a more modern set of interoperating tools. Each tool is equipped with its own HMI (Human Machine Interface), mainly web-based. Some tools, such as the Risk and Resilience assessment analysis (section 4.1) and the CISIApro modeler (section 4.3.3), operate off line. Each one provides its own HMI application and interoperates with other RESISTO tools through stored data.

The Cockpit represents the set of HMI applications that RESISTO operators can use to command & control run-time operations, in other words the Short Term Control Loop run-time HMI. As depicted in Figure 40, through the RESISTO Cockpit users can access the following main functionalities:

- **run-time situation awareness** of potential threats and alarms and related data, along with indicators to assess the current status of the infrastructure and the (communication) service performance indicators. Situation awareness is provided in different forms and points of view: dashboards, video display, geo-referenced data representation on maps;
- **risk impact evaluation**;
- **what-if analysis** of possible mitigation actions;
- **mitigation workflow** selection and progress;
- **network orchestration controller** to drive the reaction on the communication network.

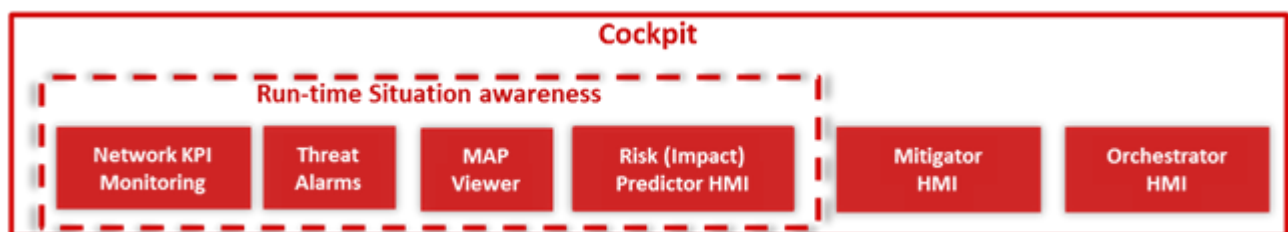


Figure 40 – Functionalities of RESISTO Cockpit.

The Cockpit is composed of a set of interoperable web-based applications for CI protection adapted to communication infrastructure. Run-time situation awareness is provided by the Leonardo SC2 WebViewer adapted for RESISTO purposes (Figure 41).



Figure 41 – Leonardo SC2 WebViewer.

The WebViewer module includes:

- situation viewer,
- map viewer, and
- event management.

The situation viewer allows:

- the configuration of the monitor layout according to the user's preferences to organize different tiles each containing dashboards or video displays;
- video streams display and management;
- dashboards display to provide monitoring of network KPIs.

The map viewer displays the geographical map systems which are geo-referenced on, the devices, potential threats, detected anomalies handled by the platform. It is possible to define different cartographic layers, each of them dedicated to a specific group of entities (cartographical elements or devices). The operator has the possibility to navigate the map and to define and switch on/off the layers. For each element of the system shown on the map, it is possible to see its essential properties in a tooltip activated with a simple mouse-over, or interact directly with the element according to its specific type. The cartographic map function allows the display of the cartographic map through the use of a GIS (Geographic Information System) showing the characteristic topographic elements of the site. As said, it is possible to define different layers, each of them gathering of all the elements having a common theme, i.e. a given characteristic (e.g. roads, pavilions, parking, etc.) and the user can choose the layers to be displayed. It is possible to have a georeferenced view of all the relevant entities in order to obtain a so-called CROP (Common Relevant Operational Picture), i.e. a view of the position of the assets monitored by the platform and of the eventual alarms/anomalies arisen. Like the normal cartography, the display of CROP entities is handled by means of thematic layers, superimposed on the cartography (Figure 42).

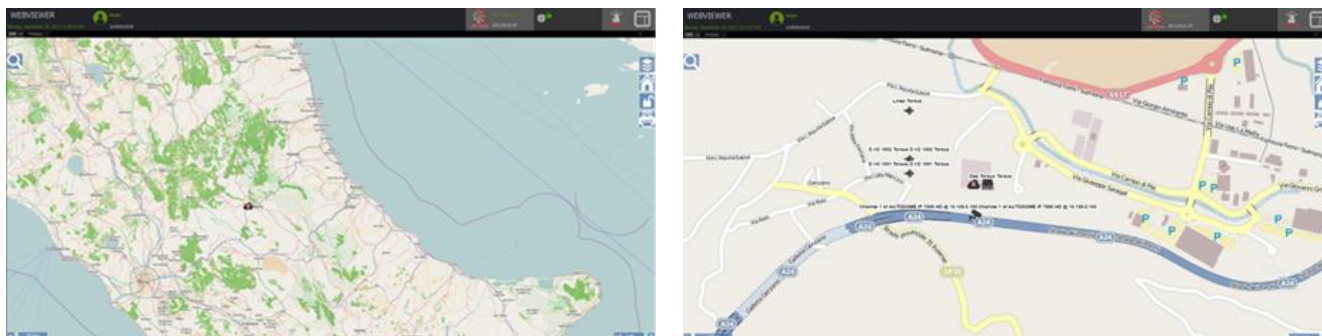


Figure 42 – Leonardo SC2 Map Viewer.

Each item is represented in the CROP using a special symbol in order to make easier to the user to distinguish the type of anomalies and assets, along with the representation of their status of operation (diagnostics). By clicking on a specific symbol, the user can have some interaction related to the selected element (Figure 43).

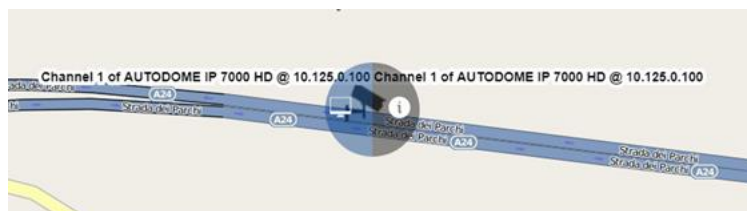


Figure 43 – Item representation in the CROP (Common Relevant Operational Picture).

The system (Correlator) generates alarms/anomalies by associating one or more critical events acquired using correlation criteria defined in the configuration of the system. Events, Alarms/Anomalies can be managed by the operators by means of the Event Management view. When an alarm/anomaly is detected by the system, the operator can evaluate the impact on the Infrastructure by the CISIAview (**Risk Predictor HMI**), the web interface of CISIApro Engine.

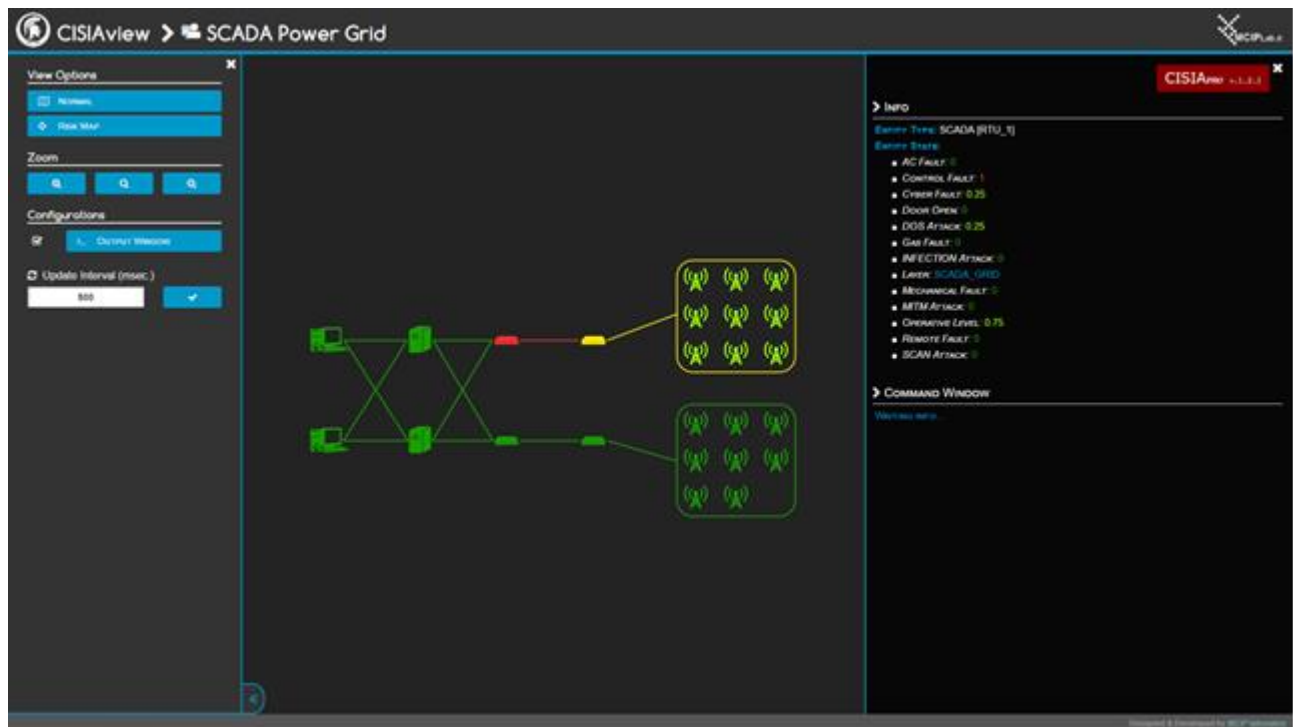


Figure 44 – CISIAview (Risk Predictor HMI).

CISIAview provides a synoptic view of the infrastructure highlighting the nominal or degraded status of each component facing an anomaly and evaluating its effect propagation. Moreover, by means of CISIAview the operator can evaluate the effect of different mitigation actions to face the anomaly in order to select the best one. To react to the anomaly, the operator can:

- Activate the relevant workflow(s).
- The workflow can be executed step by step. It is graphically presented in the ad hoc area, as a text (in order to read the actions to be performed and those already done) or in a graphical form with the indication of the activities.
- Drive the reconfiguration actions on the Communication Network acting on the Orchestrator Controller HMI.
- Coordinate reaction teams by means of Emergency Warning Functions.

4.5. Physical Resources Monitoring

4.5.1. PSIM

In the following, each TLC Operator describes its own PSIM, according to the available test beds and defined use cases.

4.5.1.1. Orange Romania PSIM

Orange Romania is currently testing a Smart Site Management System. The pilot involves 20 technical sites.

The Smart Site Management System (Figure 45) is a distributed hardware and software system that integrates a range of electronic access control devices, locks and input/output systems with management tools to help manage, control and supervise the site network operation, who has access to the sites, facilities and operations, to collect and aggregate data about site alarms, mains and generated energy, about DC (Direct Current) rectifier systems and about environment.

The benefits of using such a system in regards to improved network reliability are:

- real-time cell site health monitoring during mass power outages, weather events, and other crisis;
- improved Mean Time to Repair (MTTR) through detailed monitoring of cell site alarm conditions;
- extended site life during crisis due to intelligent management of site energy usage;
- improved network uptime due to improved physical site security.

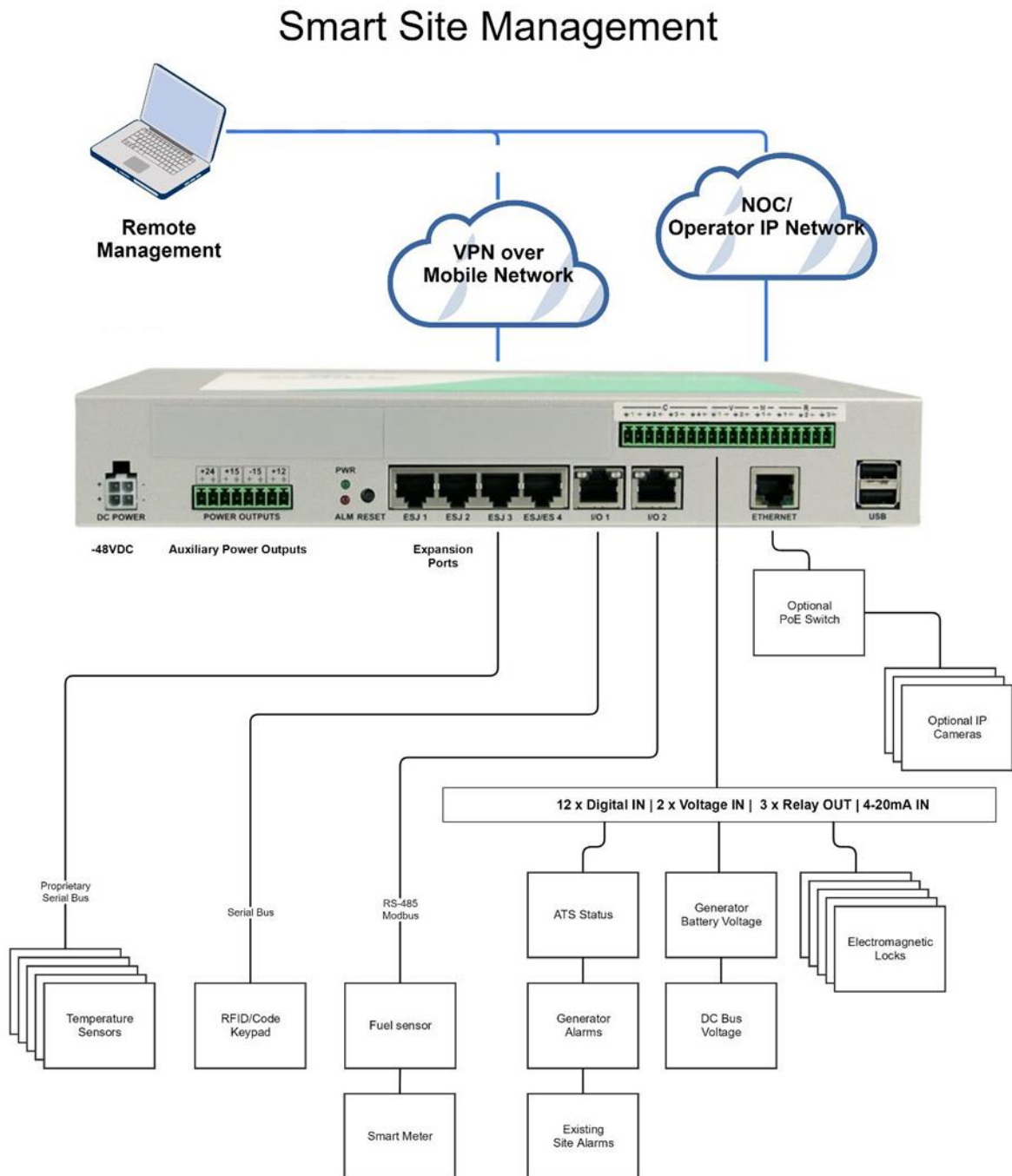


Figure 45 – Orange Romania Smart Site Management System.

The system is composed by site controllers connected to site hardware sensors and locks and a central management software application that runs on a virtual server. The server runs in the same networks as the site controllers.

Each site controller is connected to the network via the built-in Ethernet port and/or via the internal wireless modem. The wireless modem uses normal SIM cards (with a data plan) configured with a private APN (Access Point Name). The unit connects to the server using VPN connection.

Each site controller have the following main functions:

- collects power, energy, usage information and faults from the grid and from the generator (if present) using an AC (Alternating Current) smart meter;
- collects and transmits data about the diesel fuel levels, consumption and abnormal variations;
- controls remotely the generator;
- collects data and alarms (voltage, faults) from the DC rectifier system using analogue and digital inputs; additionally it collects data from solar power sites;
- collects alarms and controls the site security systems – door sensors, PIR (Post Incident Report) sensors, IP cameras and access control system – RFID (Radio Frequency Identification) reader, keypads, locks;
- collects data regarding the environment – temperatures (optional humidity) from one or more enclosures or shelters on site;
- controls remotely fans, ventilators, heaters or other HVAC (Heating, Ventilation, & Air Conditioning) systems on site;
- collects and transmits existing alarm inputs.

The site controller operates online and offline, ensuring functionality even is the connection is cut-off. Emergency site access is provided using special RFID card access independent of the working status of the site controller. The site locks used to secure the site rooms, shelters or cabinets are fail-safe design, which means that access is enabled if the mains or generator power on site is down.

The server application software system manages the remote sites both from the operational perspective – displaying events such as alarms, notifications based on changing input, measured values conditions and thresholds and also from the site access control perspective by synchronizing with each site controller the latest list of correct credentials. The application maintains a database of sites, users, access schedules, access policies and rights.

4.5.1.2. OTE PSIM

The description will be added after completing the definition of the use cases for operational validation. The correspondent deliverable D2.8 “Table-top read teaming results of RESISTO architecture, scenarios and use cases” will be delivered at month 15.

4.5.1.3. TIM PSIM

The description will be added after completing the definition of the use cases for operational validation. The correspondent deliverable D2.8 “Table-top read teaming results of RESISTO architecture, scenarios and use cases” will be delivered at month 15.

4.5.1.4. *BTC PSIM*

The description will be added after completing the definition of the use cases for operational validation. The correspondent deliverable D2.8 “Table-top read teaming results of RESISTO architecture, scenarios and use cases” will be delivered at month 15.

4.5.1.5. *RTV PSIM*

The description will be added after completing the definition of the use cases for operational validation. The correspondent deliverable D2.8 “Table-top read teaming results of RESISTO architecture, scenarios and use cases” will be delivered at month 15.

4.5.1.6. *ALB PSIM*

The description will be added after completing the definition of the use cases for operational validation. The correspondent deliverable D2.8 “Table-top read teaming results of RESISTO architecture, scenarios and use cases” will be delivered at month 15.

4.5.2. *Airborne Threats Detector*

Considering the emerging use of unmanned devices, UAVs (Unmanned Aerial Vehicles) or drones are nowadays more and more regarded as potential human-driven, physical threats. Within RESISTO, mixed techniques involving low cost radars combined with acoustic sensors are implemented to detect small airborne objects and moving targets. The relevant airborne threats detection system is a lab prototype offered by ICCS and can act complementary to other relevant, more sophisticated radar sensors provided within the project.

The specific airborne threat detection system is a set of tools designed and developed to detect the presence of small UAVs and drones as airborne threats and provide alarm signals. The system can be also deployed in small unprotected areas such as antenna telecom parks providing additional situational awareness and perimeter defence against low-flying threat aircrafts.

The system consists of active and passive sensors namely radar and acoustic sensors respectively.

The radar sensor is of a Doppler type, able to detect and track fast moving targets providing good visibility in harsh conditions (dusk, rain or snow). In contrast to optical or infrared systems, Doppler radars detect more accurately small sized objects in the (optimal) 3 GHz - 30 GHz frequency band with a sensing range of several kilometers. ICCS exploited and tested its existing Doppler CW (Continuous Wave) radar lab prototypes at microwave frequencies (X-band / 10-12GHz, K-band / around 24GHz, 2.5GHz and 875MHz) with the 24GHz band yielding optimal results. However, radar's detection capability depends on the target's RCS (Radar Cross Section) and its ability to distinguish between small objects and clutter that the sensor will also pick up especially for objects with low RCS as the drones are. This is tackled with advanced signal processing and the combined use of the acoustic sensors.

The acoustic sensors used in this system are a set (array) of high sensitivity dynamic microphones for a low cost, low power combination. Acoustic sensors have many advantages that include non-line-of-sight, omni-directionality, passiveness, low-cost and low-power, playing a potential key role in situational awareness; since they do not depend on the target's size, but rather on its acoustic signature i.e. sound of the engine. The acoustic microphone arrays are used as a second sensor

modality to detect broadband acoustic emissions from approaching targets. By exploiting the target's strong emitted sound harmonics, moving targets can be detected by acoustic sensors, through tracking of the strong sound harmonic lines they emit (in the 20Hz–2kHz range).

Advanced signal processing and machine learning techniques are applied to the radar and acoustic data, both in the time-domain and the frequency-domain to achieve detection and to estimate the target's angle of arrival. The above system's tools and components (radars and acoustic sensors) can be either used separately or in combination, through a multiplexing console and a windows-based computer (laptop or desktop) that performs the sensing-data processing and is physically connected to the sensors. Target detection visualization is enabled through the Labview software environment.

In the framework of RESISTO, the technical assessment through the relevant use cases and scenarios will be held with commercial drone platforms (i.e. DJI Phantom 3 Advanced Drone) along with the UAV platforms of ADI (ADITESS). To comply with newest technology trends, implementation of mixed techniques will be pursued; potentially in conjunction with visual methods (i.e. cameras), depending on the ADI platforms payloads. More detailed description of the ICCS airborne threats detection system and tools is already given in the relevant deliverable D4.1 "Active and Passive Sensor Definition".

Concerning the integration of the above system within the overall RESISTO architecture, it should be noted that the sensing tools may act as plug-in modules providing alerts to the PSIM part, affecting mainly the short-term control loop. The overall detection system is a standalone one and thus an interface to the overall RESISTO platform with Data Integration Layer, based on Mule Enterprise Service Bus, will be defined in order to make easy data exchange between the two systems.

Concerning the physical connection to the RESISTO platform, the detection system can be either connected to the same local network (LAN) with the RESISTO platform or, at least, can be IP visible from the RESISTO platform, depending on the final configuration that will be defined towards the project evolution and integration related aspects. The RESISTO correlator will provide a pub-sub Message Queue on which the potential threat detectors, will be able to publish potential threat events.

Consequently, and since the aim of the airborne threats detection system is to extract and provide potential intrusion events corresponding to the presence of potential moving airborne threats (UAVs and drones), a threat event with relevant attributes will be provided. Based on the above, as a preliminary at this point data model, the following can be defined:

The data exchange between the ICCS detection system and the RESISTO platform will use JSON-formatted data objects corresponding to events. In its preliminary form a JSON-formatted event will include the values listed in Table 5.

The specific event formats and related interfaces will be further examined according to the overall architecture and integration principles and will be defined and presented in the next version of the present Deliverable.

Id	Description
"eventID": Number	A unique identifier for an event, it doesn't uniquely identify the source of the event.
"eventTYPE": String	At least two types of events will be identified "ApproachingUAV" and "UnknownPotentialThreat".
"eventTIMESTAMP": String	Estimated time of the detection.
"eventDIRECTIONOFARRIVAL": Number	Estimated direction of arrival in degrees, can be NULL.
"eventDETAILS": {"THRESHOLD": Number, "DISTANCEfromTHRESHOLD": Number}	Algorithmic details that maybe useful to the RESISTO platform.
"event.LatLng": {"LATITUDE": Number, "LONGITUDE": Number}	Latitude and longitude coordinates of the detected threat.

Table 5 – Data exchange between the ICCS detection system and the RESISTO platform.

4.5.3. Audio/Video Analysis

Video and Audio sensors are widely used in surveillance operations and protection of critical infrastructures. Intelligence algorithms are applied in audio and video streams for the real-time detection of events for the early identification of illicit activity. Pattern recognition and machine learning techniques are used to extract acoustic events (i.e. gunshot, screaming, glass breaking) or to classify persons, vehicles and other objects that are moved within the controlled by the infrastructure area. Both the audio and video analytics modules form an intelligence surveillance system where the security operator is notified with an alert about the suspicious activity accompanied with important information such as location (source) and type of the event, detected objects, etc. This intelligent process reduces the effort of the operator by monitoring in a 24/7 base a huge number of sensors.

The Intelligent Audio Analytics Component (AAC), allows the detection of abnormal behaviour regardless of the field of view, while also allowing the triggering of the system with the occurrence of pre-defined keywords. The solution implements well established methods from the fields of audio coding, machine learning and speech recognition and allows efficient operation on low cost power limited devices (or embedded systems) for the detection of screaming, glass breaking and gunshots within the environment. The Video Analytics Component (VAC) provides the necessary functionalities for visual surveillance analytics, aiming to identify and provide methods that abstract the information of interest contained in video surveillance streams.

The design of Intelligence Surveillance System including both the AAC and VAC is based on two different processing levels, one with reduce processing capability (infrastructure based on embedded system, i.e. Raspberry PI 3) and one with enhanced processing capability (i.e. GPU enabled computers or servers). Based on the application, audio or video, several components will be deployed on each level of processing. AAC is composed from two main components: the audio feature extractor and the classifier which are deployed on the first level and second level respectively (Figure 46). Similar, VAC component is divided in several sub-components according the functionalities. Lightweight components such as motion detection are deployed on embedded systems (first level) while more demanding components such classification and tracking are run on the second processing level (Figure 47).

Additionally, within RESISTO project, audio and video analytics modules will be enhanced with some other components in order to support the smooth operation, logging and correlation of the events. In particular, a local database will be used to store the generated alarms, fusion component in order to correlate alarms that are generated from more than one sensor, cross-cue component which is responsible to send a PTZ order to a camera in case of and acoustic event. Furthermore, Mini-UAV System components that support the deployment of Mini-UAV platforms as part of the surveillance operation (aerial image/video) will be deployed on the second level resources.

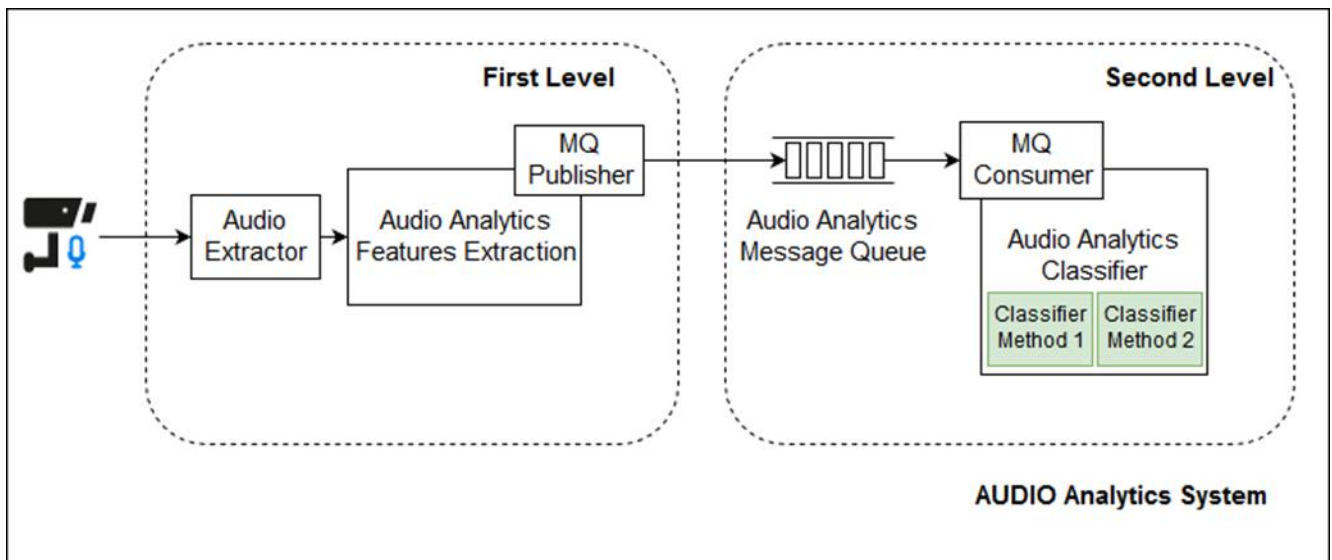


Figure 46 – Audio Analytics System.

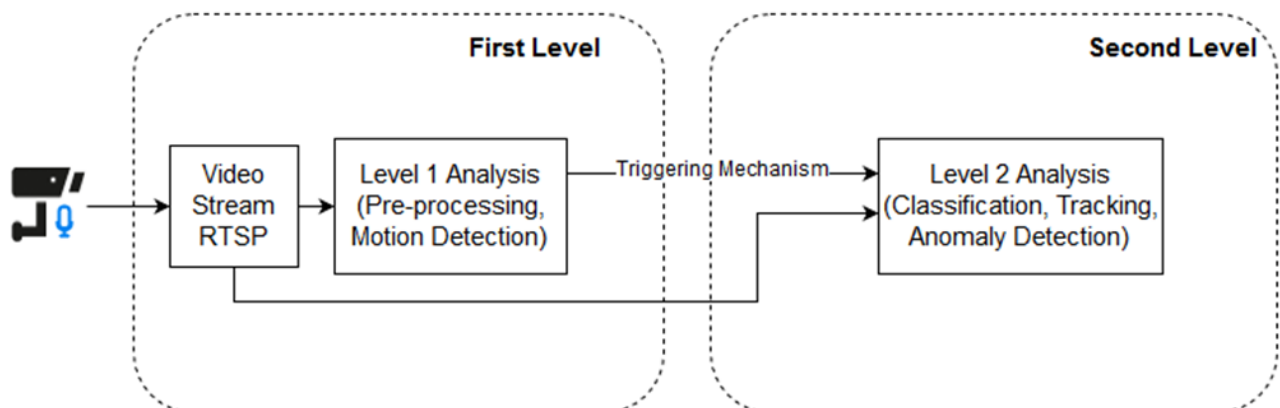


Figure 47 – Video Analytics System.

Next Figure 48 presents a higher level of the intelligence surveillance system.

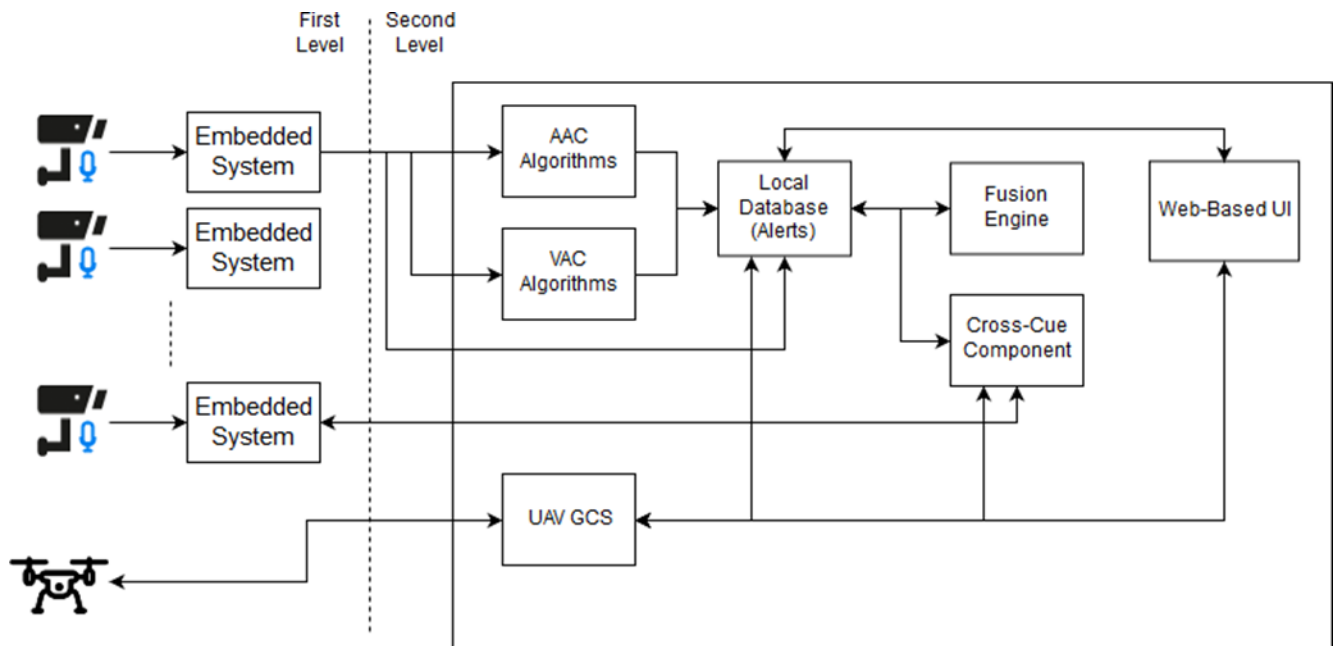


Figure 48 – Intelligence Audio/Video Surveillance System.

The interfaces and internal communication of these components are based on well-known protocols such as REST API, Message Buses (ActiveMQ), RTSP, GStreamer, etc. Next Table 6 summarizes the input and output interfaces of each component.

Component	Input	Output
Audio Extractor	Audio Stream (from sound card, gstreamer)	Audio Chunks (predefined window)
Feature Extractor	Audio Chunks (predefined window)	JSON with features
Classifier	Features (JSON)	Event (JSON) with analysis information (i.e confidence)
Video Stream (Video Proxy)	RTSP (or other) from camera	RTSP
Level 1 VAC	RTSP	Event (JSON), Trigger to VAC 2
Level 2 VAC	RTSP, Trigger (from VAC 1)	Event (JSON)
Local Database	REST API	
Fusion Engine	Message Bus Consumer (Events)	Message Bus Consumer (Events)
Cross-cue	Event (Message Bus or REST API), JSON	Order to PTZ or UAV (JSON)
UAV GCS	MAV Link, Video Stream	Events, Metadata (JSON)

Table 6 – Input and output interfaces for the audio/video analytics component.

4.5.4. Smart Spectrum Surveillance

Integrasys' **SSS (Smart Spectrum Surveillance) System** is a set of tools designed to detect unauthorized presence be it a person/equipment in a restricted-access area or a rogue base station in a mobile radio access network. The tools which make up the system are the following:

- Smart Spectrum Surveillance for Restricted-Access Environments. This tool detects persons or equipment (rogue Wi-Fi APs) unauthorized presence in a restricted area. It is based on a Monitoring WSN (Wireless Sensor Network) and a Central Processing Node.
 - Each of the sensors include Bluetooth and Wi-Fi transceivers which can capture and analyze the packets sent by any device using these technologies (Wi-Fi AP/client and Bluetooth devices) to check out the devices' MAC address.
 - All the information from the sensors is received at the Central Processing Node which has access to a whitelist (this list could be managed locally or centrally in the RESISTO Short Term Control Loop. To be decided) with the authorized devices MACs. This way, any person using a device which is not whitelisted would prompt an alarm in the system. The same goes for any rogue Wi-Fi AP.
 - On the top of that SDRs (Software Defined Radios) will be used as spectrum analyzers in the overall system.
 - Additionally the system might add a location feature that would approximately compute the position of the unauthorized person/AP (this is to be decided).

- **Smart Spectrum Surveillance for Mobile Radio Access Network.** This tool detects equipment (rogue base stations) unauthorized presence. It is based on a Monitoring mobile modem, a SDR Spectrum Analyzer and a Central Processing node.
 - A multi-band multi-RAN modem able to detect and identify the surrounding cells in a specific area.
 - A SDR spectrum analyzer able to monitor the radio spectrum.
 - The Central Processing Node will have access to a whitelist (this comes from a database that could be managed locally or centrally in the RESISTO Short Term Control Loop. To be decided) with the cells that are expected/authorized to be in a specific area (based on a central location a certain coverage radius), always having in mind that there is an unavoidable level of uncertainty related to this information. This way, any rogue base station which is not whitelisted would prompt an alarm in the system.

The inputs/outputs from the tools would be the following:

- Smart Spectrum Surveillance for Restricted-Access Environments.
 - **INPUTS**
 - ✓ Configuration parameters: Monitored technologies, monitored bands, update rate.
 - ✓ MAC addresses whitelist (this list could be managed locally or centrally in the RESISTO Short Term Control Loop. To be decided)
 - **OUTPUTS**
 - ✓ JSON file with a list of seen and current Wi-Fi clients and AP including, among others, the following data: MAC Address, BSSID (AP only), SSID (AP only), Associated BSSID (Client only), Associated SSID (Client only), First seen, Last seen, Average RSSI, Traffic type (Client only) and Coarse Location (To be decided).
 - ✓ JSON file with a list of seen and current Bluetooth clients including, among others, the following data: MAC Address, Count, Name, First seen, Last seen, Average RSSI and Coarse Location (To be decided).
- Smart Spectrum Surveillance for Mobile Radio Access Network.
 - **INPUTS**
 - ✓ Configuration parameters: Monitored technologies, monitored bands, update rate, Center coordinates and Radius.
 - ✓ Cell ID database (this database could be managed locally or centrally in the RESISTO Short Term Control Loop. To be decided).
 - **OUTPUTS**
 - ✓ JSON file with a list of seen and current cells including, among others, the following data when available: cell identifier (ECGI and/or PCI for LTE), Location area identifier (TAC for LTE), Received power (RSRP for LTE), Band, MCC and MNC.

The transport protocols to be used in both tools are yet to be agreed with involved partners but initially TCP/IP sockets are the preferred options.

4.5.5. IoT Based Sensors

Since sensor networks may gather sensitive data or used by a malicious adversary to conduct attacks, security is a key concern for such networks and for that reason particular attention is paid to

secure the sensor networks. Two aspects that will be taken into account are the **security of each sensor** and the **security in the communication between sensors**. For this activity we will take a reference a generic wireless sensor with enough computing capability to execute cryptography functions, with Bluetooth Low Energy as the main communication means and with a complementary Sub-GHz link for out of band security related transactions.

Regarding the **security of each sensor**, security threats to hardware of the sensors are a well-known concern and care is needed to reduce the opportunity for hardware to incorporate flaws or to be a vector for malicious attack. For that, embedded systems do provide a level of security in that firmware is less prone to compromise. However, testing is recommended prior to implementation.

In order to ensure the physical security of the sensors and to provide reliability of data processed or stored in the Wireless Access Networks (Bluetooth), the firmware of all the monitoring sensors will be signed by the firmware developer and every other party in software supply chain by using KSI Signature (anchored in Guardtime's KSI Blockchain), supported by a server inside the premises. The integrity of the firmware in any sensor will be checked against the signature.

Secondly, regarding **security in the communication between sensors**, it will use the wireless standard Bluetooth Low Energy (BLE), since it offers several features to ensure communication between devices. It is important to understand the specific security threats that BLE faces and how BLE security features help mitigate them. Some of the security issues are the following:

- Unsecured devices (or not signed by KSI) could listen passively to the data that are exchanged between the two paired devices and therefore signed. The way to solve this is by encrypting the data being transferred using cryptography but the key exchange protocols that BLE uses can introduce some serious security vulnerabilities that would allow an attacker to decrypt the data.
- Malicious device, pretending the other two legitimate devices, in order to fool these devices into connecting to it. This configuration not only allows the malicious device to intercept all the data that is sent, but also allows it to inject false data in the communication or delete data before it reaches its recipient.
- Identity tracking is where a malicious entity is able to associate the address of a BLE device with a specific user and then physically track that user based upon the presence of the BLE device.

To fight against these problems and try to mitigate them, the sensors of the WSN could have own pairing mechanisms, involving transferring the data once two devices are connected so that a secure link can be established. To achieve that, regardless of the version of BLE sensor to be used (the latest versions provide more secure mechanisms), it will follow three phases:

- Pairing feature exchange. The two sensors exchange authentication and link requirements. Basically, the two sensors exchange their capabilities and determining how they are going to configure a secure connection. It is also important to regard that all data exchanged during this phase is not encrypted.
- Key generation method selection and authentication. This phase is a step prior to the generation of the key used for the encryption of the link and will take the following steps:
 - Starting with authentication, each device generates its own public-private key pair. The public-private key pair contains a private key and public key. Both sensors get their public key through their private key and some parameters that are initially agreed.
 - The devices will only exchange the public key created and some random values during authentication process using one of the pairing methods. We propose to use the out of band as pairing method.

- After a series of procedures and application of functions to the public key exchanged, the two sides arrive at a resulting key that must be the same. If after checking both keys, are not the same, the pairing process will be aborted.
- Key generation. The aim is for protection against attacks and generation of the keys which will be used to encrypt the connection link. When authentication is successful, the two sensors start to compute the key which will be used for link encryption. The key on both sides must be the same. This key is used to encrypt future links so that the pairing process does not have to be repeated.

To protect a sensor's private key, a sensor needs to implements some method to prevent an attacker from retrieving useful information about the sensor's private key using invalid public keys. The inputs/outputs from this part would be the following:

- INPUTS
 - Identification of registered sensors.
 - To check the correct configuration of the sensors.
 - Data to be exchanged between sensors.
- OUTPUTS
 - Firmware/keys provisioning.
 - Protection of data (at-rest and in-transfer). The data collected by the sensors are trustable checked with the KSI signature.
 - Regular checks of firmware integrity.

Note: at first it is considered that the data itself is not important to send them to the central platform of the project. However, if it is specified that it is necessary to send them, it would be essential to sign the data sent by all the sensors of the project by using KSI Signature.

The protocols to be used are yet to be agreed with involved partners taking into account the Bluetooth protocol stack.

4.6. Network Resources Monitoring

4.6.1. SOC

In the following, each TLC Operator describes its own SOC, according to the available test beds and defined use cases.

4.6.1.1. Orange Romania SOC

ORO is in the process of designing its own Security Operations Centre as we are currently analysing three operational scenarios for the SOC, two of the scenarios involving outsourcing of some (or most) roles and activities to a third party. As we are currently in the "Think" stage of developing ORO-SOC, we'll describe the objectives, requirements, architecture, technology and the operational model(s) ORO is planning on using in this project. Regardless of the operational scenario that will be in place, the objective, architecture, requirements and operational model should be persistent and thus will be described in the following as-is.

4.6.1.1.1. Objectives

A SOC is related with the people, processes and technologies involved in providing situational awareness through the detection, containment, and remediation of IT threats. A SOC manages

incidents for the enterprise, ensuring they are properly identified, analyzed, communicated, actioned/defended, investigated and reported. The SOC also monitors applications to identify a possible cyber-attack or intrusion (event) and determines if it is a real, malicious threat (incident), and if it could have a business impact.

ORO is currently investigating building a SOC for both internal and external customers. The process will focus on the internal customers (ORO itself) and will extend to business customers in the future.

ORO is a MSSP (Managed Security Solutions Provider) thus has the capability and platforms required to offer SOC services to B2B (Business to Business) customers

ORO's approach to building a SOC is focused on sensitive data protection, compliance with industry-wide regulations and government regulations (such as GDPR, CESG-GPG 53, etc.).

4.6.1.1.2. Requirements

ORO's SOC must comply with and provide on the following requirements:

- Will use the "5 tick-boxes model" – ORO SOC will be made up of five distinct modules: event generators, event collectors, storage (messages) databases, analysis engines and orchestration and automation (reaction management) software.
- Module integration - The main problem encountered when building a SOC is the integration of all these modules, usually built as autonomous parts, while matching availability, integrity and security of data and their transmission channels. ORO SOC will be operational based on established policies, procedures and work instructions and will constantly monitor and improve it's performance by means of KPI definition and monitoring.
- Scalability – ORO SOC will be scalable by design, allowing operations in a MSSP regime for both internal and external customers.

4.6.1.1.3. Architecture

The SOC global architecture implements the different box types defined in the preceding sub-chapter. However, beside the pure technical aspects involved in such an implementation, it is necessary to consider the supervision of an IT infrastructure as a full operational project.

Data acquisition

Before setting up sensors and designing any correlation or analysis rule, it is necessary to evaluate the overall security level of the IT infrastructure to be supervised. This will make it possible to determine if an intrusion path may effectively lead to an intrusion on the target system and the criticality associated to such an intrusion attempt. Another point to be defined is the security policy, mostly in terms of access rights, permitted operations, etc.

- Technical and organizational inventory

Security level evaluation can be divided into two parts: vulnerability assessment and system criticality. This data should be stored in a specific module of the Knowledge Base: The Security Evaluation Module. The main data sources here are the automated vulnerability scanning tools' output and the internal audits such as Pentest reports. System criticality is defined according to the relative impact that an intrusion can have for each type of consequence, using standard ISO 27001 risk management concepts.

- Vulnerability database

The vulnerability database holds information about security breaches and insecure behaviour that would either impact the overall security level or that could be exploited by an attacker in order to perform an intrusion.

The database makes it possible to store the following types of vulnerabilities:

- structural vulnerabilities, i.e. vulnerabilities internal to a specific software such as a buffer overflow, format string, race conditions, etc.;
 - functional vulnerabilities, depending on configuration, operational behavior, users, etc.;
 - topology-based vulnerabilities, including networking impact on intrusions and their consequences.
- Security policy

The next step of the supervised system inventory is an organizational one and, more specifically, a review of security policy aspects that would impact either event generation and / or the reaction-reporting processes. The two major aspects of security policy that are reviewed are authorization and testing / audit procedures. Those two aspects will provide information concerning behaviour that sensors would detect. Events generated (administrator login, Port Scans, etc.) will then be marked as matching with security policy criteria. Others will be analysed as possible part of an intrusion attempt. Those pieces of information are stored in the Knowledge Base.

Event generation, collection and storage

Events are generated by the SIEM (Security Information and Event Management) in place that collects all relevant logs, samples and traces and stores those in a two-tier storage system: a short-term collector for immediate response and a long-term collector for forensics. The SIEM, in turn, collects logs, samples and traces from all network equipment, all endpoints (fixed and mobile) all virtualization platforms, databases, physical security management systems, data acquisitions systems (Figure 49). Incidents are further along generated, triaged and catalogued accordingly in the ticketing system by the SOC operators.

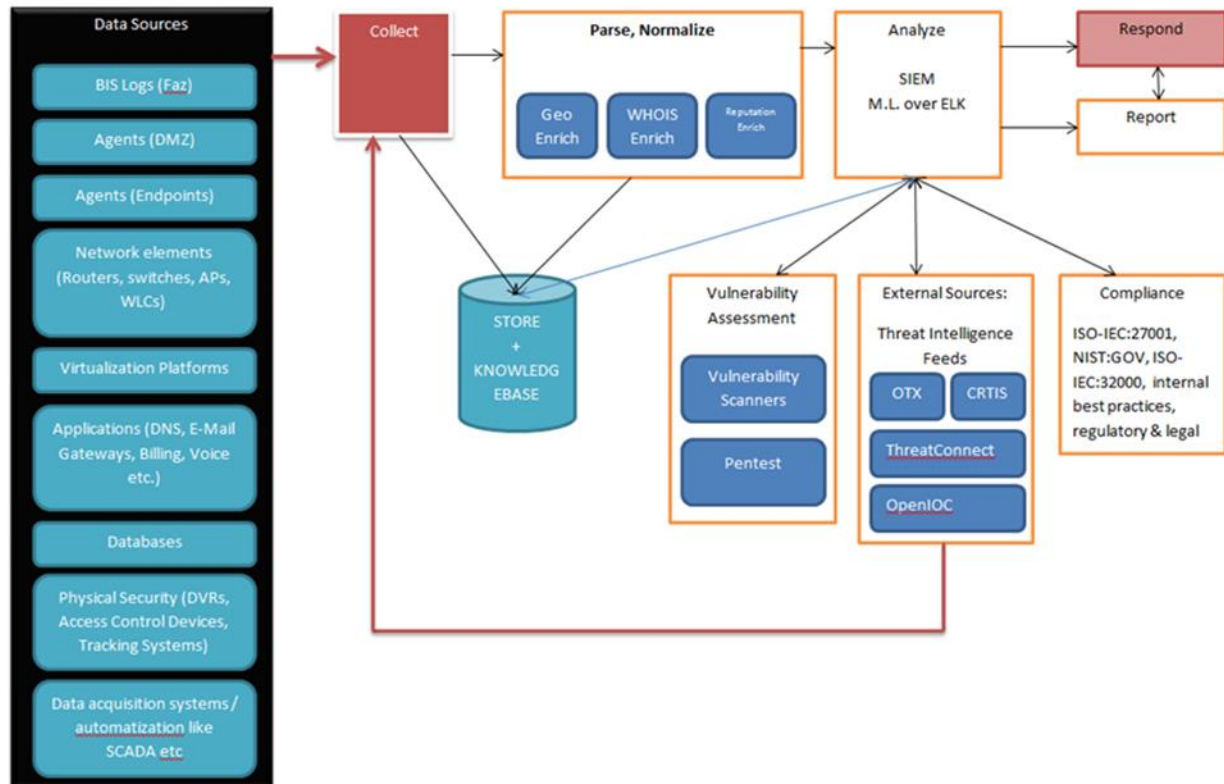


Figure 49 – ORO SOC: Architecture Overview.

Interfaces

Two interfaces are to be made available in the SOC, the SIEM Interface and the User Portal.

The SIEM interface can present data in both raw, unformatted way and in a processed, normalized way. The SIEM interface will be made available in order to access all events reported by the SIEM, access sample data, packet captures, memory dumps, raw log files etc. The SIEM interface will be presented in its web-based graphical way, spanning on multiple large displays. Several dashboards can be configured and used in different scenarios. Each dashboard is a visual, graphical representation of data in the collector box (module) and of the correlation between the events generated by the SIEM and this data. The SIEM interface is to be made available on both individual SOC Operator Workstations and on the cockpit multi-monitor display in the SOC Room.

The end-user portal provides formatted data of activity. It is designed in order to provide multi-level monitoring, response orchestration and reporting for all assets. It is divided into three main parts:

- The permanent risk evaluation interface, gives information about the current security level of supervised systems configuration and software versions. It provides information on the overall security level, vulnerability characteristics and criticality, intrusion scenarios and patch or configuration details.
- The security activity, is a mid-term to long-term reporting, providing macro data about intrusion types, frequency, sources and consequences on the supervised system. At a lower level, it is

- to be used in order to determine trends and identify specific items such as a recurring attack sources or mostly targeted services to watch for.
- The system status, which is the “pseudo real-time” interface for end-user, allowing a close follow-up of open incidents, systems under attack and intrusion paths activated by intruders. This part provides the operators with access to mitigation tools or to predefined, macro-like orchestrated scenarios that can be activated and deployed immediately following a positive detection.

Triage and Response

A basic workflow for the triage and response procedures that will be in-place in ORO SOC is provided below, in Figure 50.

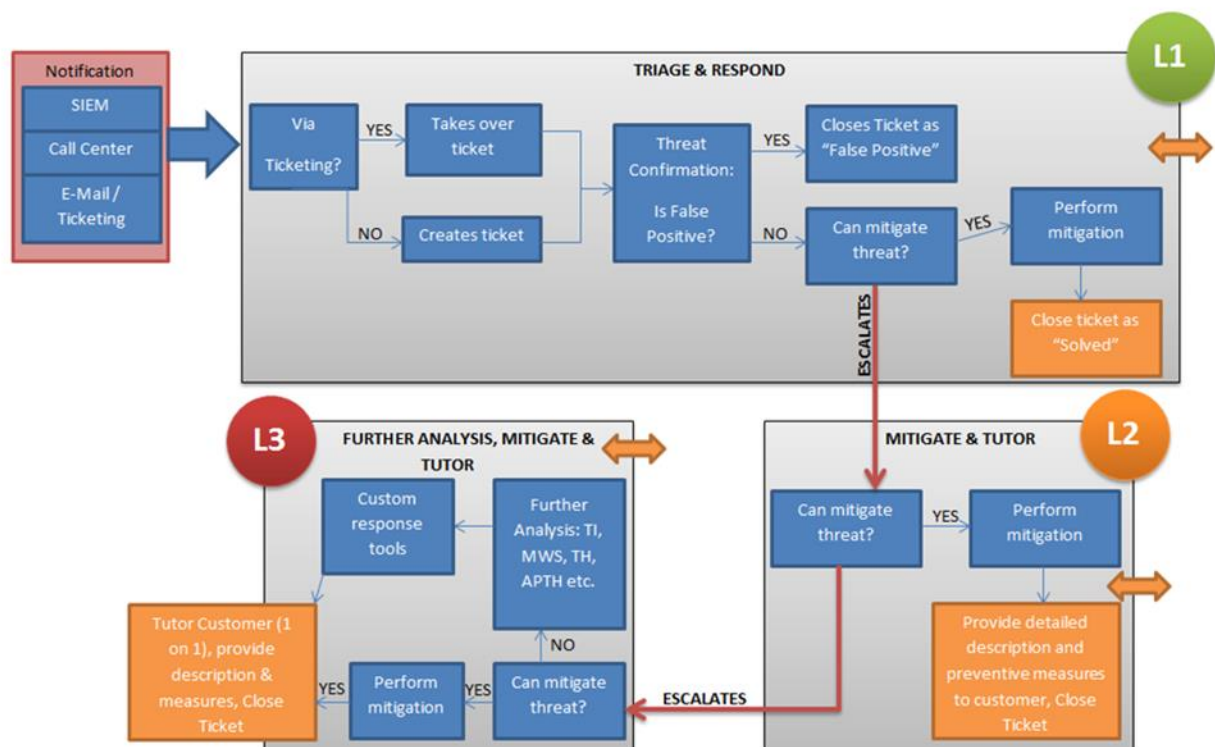


Figure 50 – ORO SOC: response workflow.

4.6.1.1.4. Technology

ORO SOC is to be built using both commercial and open-source technology, hardware and software. ORO currently deploys a SIEM solution that will interface the SOC providing the main data sources to the User Portal. ORO SOC will deploy technologies such as Kafka, Elastic Search, Logstash, Filebeat, Kibana and Grafana to build the End-User Portal. ORO SOC will use a Orchestration and Response Automation platform from a vendor or integrator to be later decided on. ORO SOC will use automated ('scripted') vulnerability assessment tools, both commercial and open-source (such as Nikto, OpenVAS, MBSA, or SecureCheq). ORO SOC will use incident analysis tools, both commercial and open-source (such as DFF, CIRTKit, OCFA, Sleuth Kit and Autopsy). All tools,

software or hardware used by ORO SOC must provide API interfaces for communicating both data and command and control sessions with the User Portal.

4.6.1.1.5. Operational Model

A basic operational model is provided below in Figure 51.

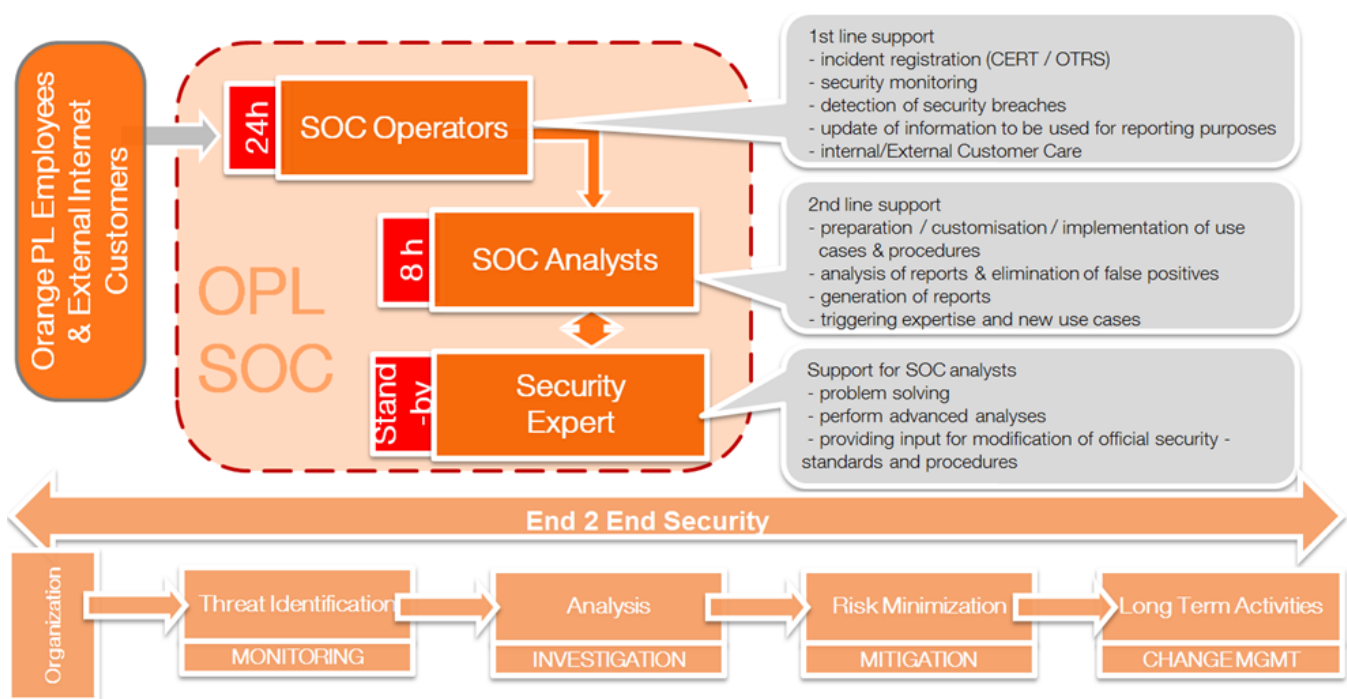


Figure 51 – ORO SOC: operational model.

ORO – SOC aims to provide End to End Security with monitoring, investigation, mitigation of incidents and change management for improvement measures. ORO – SOC is to be operated by a four-tier hierarchy of professional staff (Figure 52).

- 1) The first tier (L1) – Triage Specialist will operate in 8 hour shifts, in three rotations for 24/7 continuous monitoring of the SOC. Day to day activities includes call center operations, ticket management, basic incident triage and basic false positive detection, escalation of incidents and generating of reports from the SIEM interface.
- 2) The second tier (L2) – Incident Responders will work in 4 hour shifts by 2 rotations, on a 8/5 schedule including on-call rotations. Day to day activities include review of tickets opened by L1, review of alerts, IOCs, and events reported by L1, manual response in case of incident triggers, remediation and recovery where possible and escalation to L3 as needed.
- 3) The third tier (L3) – Threat Hunter will work in 8h shifts, 8/5 and on call for exceptional reasons. Will reviews vulnerability assessment data and performs further investigations (pentest) where required; uses latest Threat Intelligence data to identify stealthy threats such

as APTs; reviews sample data and searches for patterns associated with malware (malware sample analysis, process memory sample analysis, file sample etc. recommends optimization of monitoring tools (SIEM, log managers etc.).

- 4) The fourth tier (SOCMGR) – The SOC Manager will work 8h shifts in 8/5 and Supervises the activity of the SOC team. Recruits, hires, trains, and assesses the staff. Manages the escalation process and reviews incident reports. Develops and executes crisis communication plan to stakeholders. Runs compliance reports and supports the audit process. Measures SOC performance metrics and communicates the value of security operations to business leaders.

Level	1 – Triage Specialist (L1)	2 – Incident Responder (L2)	3 – Threat Hunter (L3)	4 – SOC Manager (SOCMGR)
	8h shifts by 3 rotations, 24/7	4h shifts by 2 rotations – 8/5 + on call outside 9-18	8h shifts, 8/5 + on-call for exceptional reasons	8h, 8/5
Day-to-day – SOC	<ul style="list-style-type: none"> -answers calls; -opens tickets; -perform basic triage of incidents; -recognize false positives and close tickets accordingly -escalation of incidents to higher competency levels -manages and configures SIEM reporting policies; -runs automated vulnerability assessments 	<ul style="list-style-type: none"> -reviews tickets open by L1; -reviews IoCs (Indicators of Compromise); -forensics analysis of compromised systems (collects data assets, samples process memory etc.); -determines and directs remediation and recovery; -escalates to L3, as needed. 	<ul style="list-style-type: none"> -reviews vulnerability assessment data and performs further investigations (pentest) where required; -uses latest Threat Intelligence data to identify stealthy threats such as APTs; -reviews sample data and searches for patterns associated with malware (malware sample analysis, process memory sample analysis, file sample etc.) -Recommends optimization of monitoring tools (SIEM, log managers etc.) 	Supervises the activity of the SOC team. Recruits, hires, trains, and assesses the staff. Manages the escalation process and reviews incident reports. Develops and executes crisis communication plan to stakeholders. Runs compliance reports and supports the audit process. Measures SOC performance metrics and communicates the value of security operations to business leaders.
Personal development, learning, certification, other activities	-Participates in general cyber security training conducted by SOCMGR + L3 – weekly, 3hrs - training should cover cyber-sec notions, technologies and tools and also work procedures	<ul style="list-style-type: none"> -Participates in professional certification courses (facilitated by ORO) such as CEH, LPT (ECCouncil); -Participates in training conducted by ORO HR, Learning & Development on matters like team work, presentation and communication skills etc. 	<ul style="list-style-type: none"> -Participates in professional certification courses (facilitated by ORO) such as CEH, LPT (ECCouncil); -Participates in training conducted by ORO HR, Learning & Development on matters like team work, presentation and communication skills etc. 	

Figure 52 – ORO SOC: four-tier hierarchy.

4.6.1.2. OTE SOC

The description will be added after completing the definition of the use cases for operational validation. The correspondent deliverable D2.8 “Table-top read teaming results of RESISTO architecture, scenarios and use cases” will be delivered at month 15.

4.6.1.3. TIM SOC

The description will be added after completing the definition of the use cases for operational validation. The correspondent deliverable D2.8 “Table-top read teaming results of RESISTO architecture, scenarios and use cases” will be delivered at month 15.

4.6.1.4. BTC SOC

The description will be added after completing the definition of the use cases for operational validation. The correspondent deliverable D2.8 “Table-top read teaming results of RESISTO architecture, scenarios and use cases” will be delivered at month 15.

4.6.1.5. RTV SOC

The description will be added after completing the definition of the use cases for operational validation. The correspondent deliverable D2.8 “Table-top read teaming results of RESISTO architecture, scenarios and use cases” will be delivered at month 15.

4.6.1.6. ALB SOC

The description will be added after completing the definition of the use cases for operational validation. The correspondent deliverable D2.8 “Table-top read teaming results of RESISTO architecture, scenarios and use cases” will be delivered at month 15.

4.6.2. OSINT and Machine Learning based Threat Detector

Some programmatic issues related to this task have arisen and are still to be solved. Therefore this section will be compiled later and presented in deliverable D2.7 “RESISTO platform and tools reference architecture - final” at month 15 (M15).

5. CONCEPT OF EXECUTION

This section will be provided with the final issue of the document, which is deliverable D2.7 “RESISTO platform and tools reference architecture - final” at M15. In fact, it needs further activities that are planned to be carried out after the issue of the present D2.6 “RESISTO platform and tools reference architecture - first”.

6. LOGICAL INTERFACE DESIGN

This section will be provided with the final issue of the document, which is deliverable D2.7 “RESISTO platform and tools reference architecture - final” at M15. In fact, it needs further activities that are planned to be carried out after the issue of the present D2.6 “RESISTO platform and tools reference architecture - first”.

7. REFERENCES

INDEX	REFERENCE
[Ref1]	RESISTO – Grant Agreement. Project Starting Date: May, 1 st 2018
[Ref2]	Häring I. et al., 2017. Towards a Generic Resilience Management, Quantification and Development Process: General Definitions, Requirements, Methods, Techniques and Measures, and Case Studies. <i>Resilience and Risk (NATO Science for Peace and Security Series C: Environmental Security)</i> ed. I. Linkov and J.M. Palma-Oliveira (Dordrecht: Springer Netherlands), pp. 21–80.
[Ref3]	CISIApro Webpage http://cisiapro.dia.uniroma3.it/
[Ref4]	URANIUM project http://uranium.ing.uniroma3.it/
[Ref5]	H2020 ATENA project https://www.atena-h2020.eu/