Distributed Intelligence for Enhancing Security and Privacy of Decentralised and Distributed Systems (Di4SPDS)

Topic: Chist-era 2022 — Security and Privacy in Decentralised and Distributed Systems (SPiDDS)

Deliverable D.4.1 Evidence based Risk Management and sustainable security

Work Package	*WP4.1: Evidence based Dynamic Risk Management* *WP4.2: Sustainable Security*
Delivery Date	*30º December 2025*
Responsible Partner	*University of Castilla-La Mancha / Firat University*
Authors	*Antonio Santos-Olmo (UCLM)* *Luis Enrique Sánchez (UCLM)* *David G. Rosado (UCLM)* *Carlos Blanco (UCLM)*
Contributors	*José Luis Ruiz-Catalán (UCLM)* *Natalia Jiménez (UCLM)* *Eugenio Romero (UCLM)* *Joaquín Sierra (UCLM)*
Version	*2.1*
Reviewer Name	*Eduardo Fernández-Medina (UCLM)*

Version History

Version	Date	Comments, Changes, Status	Authors, contributors, reviewers
0.1	*01.06.2025*	*Table of Contents*	*Antonio Santos-Olmo (UCLM)*
1.0	*22.09.2025*	*MARISMA-DS*	*José Luis Ruiz-Catalán (UCLM)* *Luis Enrique Sánchez (UCLM)*
1.1	*05.11.2025*	*VulnQ*	*José Luis Ruiz-Catalán (UCLM)* *Joaquín Sierra (UCLM)*
1.2	*22.11.2025*	*Risk Management Middleware*	*Natalia Jiménez (UCLM)*
1.3	*22.11.2025*	*Architecture Diagram & Architecture Overview*	*Eugenio Romero (UCLM)*
2.0	*22.12.2025*	*Sustainable Security*	*David G. Rosado (UCLM)* *Carlos Blanco (UCLM)*
2.1	*30.12.2025*	*Final review*	*Antonio Santos-Olmo (UCLM)* *Eduardo Fernández-Medina (UCLM)*

List of Abbreviations and Acronyms

Abbreviation / Acronym	*Meaning*
API	*Application Programming Interface*
B2B	*Business-to-Business*
CAPEC	*Common Attack Pattern Enumeration and Classification*
CASB	*Cloud Access Security Broker*
CPE	*Common Platform Enumeration*
CPS	*Cyber-Phycal Systems*
CVE	*Common Vulnerabilities and Exposures*
CVSS	*Common Vulnerabilities Scoring System*
CWE	*Common Weakness Enumeration*
DDoS	*Distributed Denial of Service*
Di4SPDS	*Distributed Intelligence for Enhancing Security and Privacy of Decentralised and Distributed Systems*
DS	*Distributed Systems*
ENISA	*European Union Agency for Cybersecurity*
EPSS	*Exploit Prediction Scoring System*
ETL	*Extract, Transform and Load*
GDPR	*General Data Protection Regulation*
HTTP	*Hypertext Transfer Protocol*
ICS	*Industrial Control Systems*
IDS/IPS	*Intrusion Detection System / Intrusion Prevention System*
IoT	*Internet of Things*
ISMS	*Information Security Management*
ISO/IEC	*International Organization for Standardization / International Electrotechnical Commission*

KRI	*Key Risk Indicators*
MARISMA	*Information Systems Risk Analysis and Management Framework*
ML	*Machine Learning*
NIST	*National Institute of Standards and Technology*
OWASP	*Open Web Application Security Project*
PII	*Personally Identifiable Information*
RAM	*Risk Assessment and Management*
REST	*Representational State Transfer*
RoHS	*Restriction of Hazardous Substances*
SaaS	*Software as a Service*
SAML	*Security Assertion Markup Language*
SCADA	*Supervisory Control and Data Acquisition*
SHA	*Secure Hash Algorithm*
SIEM	*Security Information and Event Management*
TLS	*Transport Layer Security*
VPR	*Vulnerability Priority Rating*
WAF	*Web Application Firewall*

Summary

This deliverable reports the results achieved in **WP4 (Risk Management, Communication and Sharing)**, addressing two complementary objectives: **evidence-based dynamic risk management** for distributed and decentralised systems, and **sustainable security** as a first-class concern. It introduces **MARISMA-DS**, a specialised and reusable risk management pattern derived from the MARISMA meta-pattern and aligned with recognised standards and technical guidance (e.g., ISO/IEC 27001/27002/27005, NIST and ENISA recommendations), explicitly modelling assets, threats, vulnerabilities, controls, analysis dimensions, and their interdependencies to better capture risk propagation in highly interconnected environments.

To operationalise continuous and data-driven decision-making, the deliverable further presents **VulnQ**, a web-based vulnerability and asset management system that integrates standard security taxonomies (CVE/CWE/CAPEC/CPE), automated ETL pipelines, prioritisation and alerting capabilities, and a predictive module that estimates the monthly evolution of asset risk using Key Risk Indicators and regression-based modelling. In addition, a **Risk Management Middleware** is described to automate incident and risk workflows over an external platform lacking a documented API, using HTTP traffic inspection and a REST API implementation (Python/FastAPI) to orchestrate the required call sequences with traceability and observability. Finally, the deliverable extends the analysis beyond security-only criteria through **EcoSec** (and related ontology work), enabling sustainability-aware reasoning and labelling (A–G) of security mechanisms and devices across operational and lifecycle dimensions.

Content

Version History 2
List of Abbreviations and Acronyms 3
Summary 5
1 Introduction 8
1.1 Purpose and Scope 8
1.2 Goals of the Deliverable 8
1.3 Expected Impact 9
1.4 Structure of the document 9
2 MARISMA-DS – A Pattern for Dynamic Evidence Based Risk Management for DDS based on MARISMA 11
2.1 Introduction 11
2.2 Related Work 13
2.2.1 International safety and risk management standards used 14
2.2.2 Comparison of traditional approaches with the MARISMA-DS Pattern 16
2.3 MARISMA Framework 18
2.4 Defining the pattern for IoT infrastructures in MARISMA-DS 21
2.4.1 Security Domains and Objectives 23
2.4.2 Inventory and Asset Classification 23
2.4.3 Security Dimensions for Risk Assesment 25
2.4.4 Types of Threats 26
2.4.5 Domain-Objective-Threat Matrix in DS Infrastructures 27
2.4.6 Asset-Threat-Dimensions Matrix in DS Infrastructures 29
2.5 Conclusion and Future Work 31
3 Predictive Evidence-Based Security Threat Management System Leveraging CWEs, CVEs, And CAPECs 33
3.1 Introduction 33
3.2 Related work 33
3.2.1 Existing tools 34
3.3 Development 34
3.3.1 System base 35
3.3.2 Database and search engine 36
3.3.3 Prioritization and alerts 37

3.3.4 Risk estimation (ML) 38
3.3.5 Training and prediction 39
3.3.6 Model integration and testing 39
3.3.7 Visualization 39
3.4 Conclusions 40
4 Risk Management Middleware 42
4.1 Analysis of external system through HTTP traffic inspection 42
4.1.1 Capture of the authentication flow 42
4.1.2 Identification of the complete functional flow 42
4.1.3 Summary of identified endpoints 43
4.2 Middleware design 44
4.2.1 Architecture summary 44
4.2.2 Architectural Overview 44
4.2.3 API endpoints 49
4.2.4 Middleware analysis (CustomMiddleware) 50
5 Sustainable Security 51
5.1 EcoSec Ontology 52
5.1.1 Sustainability Labels 54
5.1.2 Sustainability Requirements 55
5.1.3 Sustainability of Security Mechanisms 56
5.1.4 Sustainability of Devices 63
6 Conclusions 71
7 References 72

1 Introduction

1.1 Purpose and Scope

This deliverable document the work carried out under WP4 (Risk Management, Communication and Sharing), with a specific focus on evidence-based dynamic risk management and sustainable security for distributed and decentralised systems. It addresses the growing cybersecurity challenges posed by highly interconnected, heterogeneous and resource-constrained distributed systems, where traditional, static and largely manual risk assessment approaches struggle to capture evolving threat contexts and dependency-driven risk propagation.

Within this scope, the deliverable presents a set of complementary contributions aimed at enabling **dynamic, operational and reusable risk management** in distributed environments. First, it introduces **MARISMA-DS**, a specialised risk management pattern derived from the MARISMA meta-pattern and grounded in recognised standards and technical guidance (e.g., ISO/IEC 27001/27002/27005, relevant NIST publications and ENISA recommendations), explicitly modelling assets, threats, controls and security dimensions alongside their interdependencies. Second, it describes **VulnQ**, a web-based vulnerability and asset management system that leverages security taxonomies (e.g., CVE/CWE/CAPEC) and integrates predictive analytics to support proactive risk monitoring and prioritisation.

In addition, the deliverable covers the development of a Risk Management Middleware layer designed to automate risk/incident workflows and system interactions where direct integration interfaces are unavailable, supporting end-to-end operationalisation of risk management processes. Finally, it extends the scope beyond “security-only” considerations by incorporating sustainable security, introducing an ontology-driven approach (EcoSec) intended to reason about and evaluate the sustainability footprint of cybersecurity solutions so that security objectives can be achieved with improved resource efficiency and environmental awareness.

1.2 Goals of the Deliverable

The primary goal of this deliverable is to provide an **evidence-based and dynamic approach to risk management** for distributed systems, addressing the limitations of traditional, largely static RAM methodologies in environments characterised by heterogeneity, scale, interconnectivity and rapidly evolving threats. In this context, the deliverable aims to define and validate **MARISMA-DS** as a reusable, modular risk management pattern grounded in recognised standards and guidance (e.g., ISO/IEC 27001/27002/27005, NIST, ENISA), with explicit modelling of **assets, threats, controls and security dimensions**, as well as their dependency relationships to better represent risk propagation and support continuous risk management.

In addition, the deliverable aims to operationalise these concepts through supporting technical enablers: (i) **VulnQ**, a web-based vulnerability and asset management system that leverages established taxonomies (e.g., CVE/CWE/CAPEC) and introduces **predictive analytics** to anticipate the evolution of cybersecurity risk over time; (ii) a **Risk Management Middleware** layer to automate and orchestrate risk/incident workflows and facilitate integration where direct interfaces are limited; and (iii) a **sustainable security** contribution, introducing ontology-driven foundations to reason about and label the sustainability footprint of security mechanisms and devices across operational and lifecycle dimensions, enabling more resource-efficient security decision-making without compromising protection objectives.

1.3 Expected Impact

The work reported in this deliverable is expected to strengthen the **operational cyber-resilience** of distributed and decentralised systems by enabling risk decisions that are **faster, more consistent, and better aligned with DS-specific regulatory and best-practice guidance**. In particular, the reuse-oriented and modular nature of the MARISMA-DS pattern is intended to reduce the effort of repeatedly starting assessments from scratch, improving the consistency of analyses across scenarios and supporting more scalable adoption from domestic environments to critical infrastructures. In parallel, the explicit modelling of dependency relationships is expected to improve the identification of **risk propagation effects**, helping organisations prioritise controls and mitigation actions more effectively as systems evolve.

A further impact is the promotion of a more resource-aware and environmentally responsible security posture. By introducing sustainability as a first-class concern through ontology-driven constructs (e.g., EcoSec sustainability labels and lifecycle/operational criteria), the deliverable supports more transparent comparison of security options and encourages configurations that maintain protection while improving efficiency. This contributes to decision-making that balances security strength with longer-term operational viability, particularly relevant for cyber–physical and IoT-heavy deployments.

1.4 Structure of the document

The reminder of the document is structured as follows: Section 2 presents MARISMA-DS, describing the proposed pattern for evidence-based dynamic risk management in distributed and decentralised systems, including its conceptual foundations, key components, and alignment with relevant standards and guidance. Section 3 introduces the predictive, evidence-based security threat and vulnerability management system (VulnQ), outlining its objectives, architecture, and how it leverages established taxonomies (e.g., CWE/CVE/CAPEC) to support monitoring and prediction. Section 4 describes the Risk Management Middleware, detailing how

the automation/orchestration layer is designed to support incident and risk workflows when direct integration interfaces are limited. Section 5 addresses Sustainable Security, presenting the ontology-driven approach and the role of EcoSec in supporting sustainability-aware cybersecurity decision-making. Finally, Section 6 concludes the deliverable by summarising the main outcomes and their contribution to WP4 objectives.

2 MARISMA-DS – A Pattern for Dynamic Evidence Based Risk Management for DDS based on MARISMA

2.1 Introduction

The Distributed Systems (DS) represents an emerging technology with high transformative potential, whose current and future applications promise substantial improvements in user comfort, process automation, and productivity across multiple industrial, commercial, and domestic sectors [2.11, 2.44]. Indeed, DS is among the most relevant and debated topics in contemporary research, being regarded by numerous experts as a foundational technological pillar upon which the future of a connected society will be built [2.6]. Since its inception, significant research and development efforts have been directed toward this field; however, multiple vulnerabilities have also been identified, jeopardizing its secure and widespread adoption [2.10, 2.44].

The increasing penetration of DS in critical contexts [2.4], such as industry, healthcare [2.18, 2.43], energy [2.30], and transportation [2.54], has led to increasingly complex architectures, composed of heterogeneous and highly interconnected devices. While this interconnectivity enables the monitoring and intelligent control of distributed systems [2.4], it proportionally expands the attack surface, thereby posing substantial cybersecurity challenges [2.7, 2.33].

Within this scenario, ensuring the secure and sustainable adoption of DS necessitates the integration of robust security measures, advanced privacy-preserving mechanisms [2.33], and resilient authentication schemes capable of operating under resource-constrained conditions [2.49]. Such constraints—including limited processing capacity, scarce memory resources, low power consumption, bandwidth limitations, and protocol diversity—hinder the application of conventional security solutions, compromising fundamental information properties such as confidentiality, integrity, availability, and verifiability [2.44].

The proliferation of DS devices reliant on continuous connectivity to open networks such as the Internet has triggered a steady increase in the frequency, sophistication, and scope of cyberattacks. These attacks, often automated and persistent, can result in unauthorized access, exfiltration of sensitive data, malicious data manipulation, or even the denial of critical services, directly impacting the operational stability of affected systems [2.33]. Consequently, DS environments in real-world scenarios increasingly regard security and privacy as central pillars in the design and operation of their technological infrastructures [2.29].

The accelerated evolution of the DS paradigm, catalyzed by global digital transformation, has turned DS into a highly complex distributed cyber-physical system whose technical and societal implications demand a multidisciplinary analytical approach. Despite its numerous benefits, DS introduces inherent risks associated with data exposure, information traceability, device

authenticity, and distributed identity management. In this context, the development of comprehensive risk management strategies and adaptive security frameworks that consider the entire lifecycle of devices and associated architectures becomes imperative [2.29].

In this regard, risk assessment emerges as an essential component not only from a technical perspective but also from a regulatory standpoint, constituting one of the cornerstones of the European Union's legal frameworks in cybersecurity, privacy, and data protection. This regulatory emphasis highlights the importance of identifying, analyzing, and mitigating risks associated with technological systems, thereby fostering the development of risk assessment methodologies and tools specifically tailored to DS environments [2.8].

In summary, within an environment characterized by the continuous expansion of connected devices, DS security constitutes a top-priority challenge in modern systems engineering. The integration of emerging technologies such as blockchain [2.49], intrusion detection systems [2.34], digital twins, and fog computing represents a strategic avenue for mitigating vulnerabilities and strengthening trust in the large-scale deployment of DS solutions [2.44].

Therefore, the rapid advancement and massive adoption of DS highlight the critical need to implement comprehensive and efficient risk assessment methodologies specifically oriented to this complex ecosystem. The state of the art reveals significant limitations in conventional approaches, particularly regarding the estimation of atomic attack probability and the quantitative and qualitative valuation of assets involved. Furthermore, these methods frequently overlook the analysis of interdependencies and associative relationships among nodes, aspects that directly influence the accurate evaluation of individual host risk within DS networks. This omission undermines the ability to faithfully represent threat propagation dynamics and real exposure to risk, ultimately compromising the effectiveness of mitigation and management strategies. As a result, it is imperative to develop advanced models that integrate both individualized node assessment and the contextual analysis of their interrelations, with the aim of optimizing risk identification and prioritization in DS environments [2.55].

Based on these premises, this work proposes the definition of a reference risk pattern specifically designed for DS environments, grounded in the MARISMA framework. The objective is to provide a model intrinsically compatible with the distinctive characteristics of these environments, such as scalability, complexity, technological heterogeneity, and component dynamism. This pattern aims to ensure comprehensive coverage of assets, threats, controls, and security dimensions relevant to DS systems, as well as the dependencies and interactions among these elements. Consequently, it enables a holistic and contextualized representation of risk in such environments, ensuring a systematic and adaptable approach to risk management in contexts characterized by high variability and operational complexity.

2.2 Related Work

The accelerated adoption of DS-based systems has significantly increased the need to develop and implement specialized risk management frameworks capable of addressing the cybersecurity challenges inherent to highly interconnected and heterogeneous environments. In response to this need, multiple international standards, regulations, and guidelines have been established to enhance the security, resilience, and reliability of critical infrastructures and DS devices, providing practical guidance and best practices for their effective protection. The specialized literature acknowledges several widely used proposals in the field of information security, which serve as a basis for DS environments. Among these, MAGERIT (2012) [2.36] , OCTAVE (2007) [2.3], MONARC (2024) [2.48], CORAS (2011) [2.35], MEHARI (2010) [2.9], as well as international standards from the ISO/IEC family (27005:2022 [2.22]; 21827:2008 [2.21]; 27400:2022 [2.23]; 31000:2018 [2.20]; etc.), and reference frameworks such as COBIT (2019) [2.26] or NIST standards [2.31, 2.32, 2.37], stand out. Additionally, various ENISA recommendations exist [2.13–2.17].

Within the context of DS environments, risk assessment and management (RAM) assumes critical importance due to the growing complexity, heterogeneity, and level of interconnection of these ecosystems. Nevertheless, there remains a lack of mature and standardized methodological frameworks that comprehensively address the risks associated with DS.

This section provides a detailed analysis of the main conceptual frameworks, methodologies, recommendations, regulations, and standards related to Risk Assessment and Management (RAM), as well as various proposals specifically aimed at DS ecosystems. Furthermore, a critical review of the most relevant regulatory standards and frameworks considered as a basis for the design and construction of the approach proposed in this work is presented.

MAGERIT [2.36] incorporates risk management within the organizational governance framework, offering a structured approach for identification, analysis, and mitigation. OCTAVE [2.3], MEHARI [2.9], and CORAS [2.35] provide strategic, methodological, and graphical models, respectively, enabling the consideration of both vulnerabilities and threat scenarios in DS infrastructures.

MONARC [2.48], developed by GOVCERT.LU, integrates predefined scenarios, control effectiveness and maturity metrics, and automatic calculation of inherent and residual risk, adapting to DS architectures through an open-source collaborative platform. Similarly, ENISA guidelines for DS security [2.13-2.17] offer practical recommendations specifically oriented toward critical sectors.

From a regulatory perspective, ISO/IEC 27005 [2.22], ISO/IEC 21827 [2.21], ISO/IEC 15443 [2.27, 2.28], and ISO/IEC 27400 [2.23] address risk management, security maturity, and specific

requirements for DS environments. Complementarily, ISO 31000 [2.20] and BSI 7799-3 [2.5] provide systematic principles and approaches for risk evaluation and treatment.

NIST provides widely adopted frameworks, such as SP 800-30 Rev. 1 [2.31] for risk analysis, RMF (SP 800-37 Rev. 2) [2.32] for comprehensive risk management, and the SP 800-160 series [2.38, 2.42] and NISTIR 8228/8259 [2.39-2.41], which cover topics from secure systems engineering to minimum cybersecurity capabilities for DS devices.

BSI IT-Grundschutz [2.19] offers a modular approach, based on protection levels and safeguard catalogs adaptable to DS environments. It integrates the PDCA cycle and predefined templates that facilitate traceability and documentation, critical aspects in heterogeneous environments. EBIOS Risk Manager 2.0 [2.2] introduces a scenario-oriented approach, allowing control prioritization and maintaining traceability in strategic and operational decisions.

Regarding specialized tools, RESISTO [2.1] stands out by integrating dynamic risk analysis, automated response, and artificial intelligence techniques under a software-defined security (SDSec) approach, which is particularly relevant for cyber-physical infrastructures such as DS.

Finally, among emerging initiatives, the ETSI EN 303 645 standard [2.12] defines minimum security requirements for consumer DS devices, and the DS Security Foundation best practices [2.25] are oriented toward secure development and lifecycle management of these devices.

Despite these frameworks, DS environments still present significant gaps in specific RAM proposals tailored to their dynamic, distributed, and heterogeneous characteristics. While preliminary initiatives exist for DS, industrial systems, SCADA, or smart grids, many lack maturity, standardization, and supporting tools to facilitate practical implementation. This is often because significant adaptation or customization is required due to the particularities of DS devices, networks, and applications (limited resources, large scale, interconnectivity, emerging threats). Interoperability among different DS devices and platforms, as well as full lifecycle management (from manufacturing to decommissioning), are critical aspects that often complicate the application of existing security frameworks.

To address this need, the MARISMA framework [2.53] is proposed, designed to flexibly address risks through the use of patterns as a mechanism for extension and adaptation to dynamic scenarios. This allows for comprehensive, scalable management aligned with international standards, demonstrating its utility and effectiveness in multiple previously developed patterns, such as MARISMA-CPS [2.46], MARISMA-BiDa [2.45], MARISMA-BP [2.47], among others.

2.2.1 International safety and risk management standards used

Among the most relevant regulatory frameworks for managing security in technological environments is ISO/IEC 27001:2023 [2.24], which defines the requirements for establishing, maintaining, and improving an Information Security Management System (ISMS), adopting a risk-

based approach to protect the confidentiality, integrity, and availability of assets. Its applicability extends to DS infrastructures, enabling the establishment of controls consistent with the device lifecycle and threats arising from ubiquitous connectivity.

Complementarily, ISO/IEC 27005:2022 [2.22] provides a methodological guide to identify, analyze, and treat associated risks without imposing prescriptive requirements. Its flexibility allows risk analysis to be adapted to DS environments characterized by dynamism, heterogeneity, and interdependence. Likewise, ISO/IEC 27002:2022 [2.51] offers technical guidelines for implementing the Annex A controls of 27001, structured across organizational, physical, technological, and human domains. This version introduces an attribute-oriented approach including authenticity, traceability, and non-repudiation, integrating key aspects such as secure configuration, data protection, and lifecycle management of digital assets, which are particularly relevant in DS ecosystems.

These standards form the methodological foundation for a technical pattern for risk analysis and management in DS infrastructures. ISO/IEC 27001 [2.24] establishes the organizational structure, ISO/IEC 27005 [2.22] operationalizes risk management through systematic processes, and ISO/IEC 27002 [2.51] translates those processes into applicable controls.

In parallel, NIST provides a set of complementary frameworks. SP 8259 [2.40] proposes fundamental activities for DS device manufacturers, aimed at integrating minimum cybersecurity capabilities from the design phase. Its incorporation into a risk management pattern allows for the assessment of inherent exposure starting at the manufacturing stage.

NIST SP 800-82 Rev. 2 [2.52] provides guidelines for the security of industrial control systems (ICS), including SCADA and BMS, which are often integrated with DS devices. It is essential for modeling risks in hybrid architectures, enabling the identification of attack vectors and mitigating controls. In parallel, SP 800-161 Rev. 1 [2.50] addresses supply chain risk management, considering factors such as component provenance, firmware validation, and supplier trust. Its application in environments such as Smart Buildings is critical to ensure interoperability and reduce exogenous risks.

Collectively, these NIST standards provide a robust regulatory basis for designing technical patterns for risk analysis and management tailored to the specificities of DS, ensuring a structured approach for identifying, assessing, and mitigating risks in distributed and complex environments.

Additionally, specific guidelines such as those from ENISA complement this regulatory framework. The document Baseline Security Recommendations for DS in the Context of Critical Information Infrastructures [2.17] establishes technical and organizational measures to mitigate risks in systems such as HVAC, IP surveillance, and environmental sensors, facilitating the

definition of proportional controls based on device criticality. Likewise, the report Threat Landscape and Good Practice Guide for Smart Home and Converged Media [2.16] identifies characteristic threats and attack vectors in smart home environments, such as device hijacking, DDoS, or data interception, and proposes mitigation practices.

These guidelines provide a contextualized and operational view of cybersecurity in DS, enabling the enrichment of a methodological risk analysis pattern with controls adapted to different functional domains (industrial, residential, multimedia), thereby enhancing the maturity of the management model against emerging threats.

2.2.2 Comparison of traditional approaches with the MARISMA-DS Pattern

In conclusion, various methodological approaches have historically been employed for risk assessment and management in information systems, including widely recognized frameworks such as MAGERIT, OCTAVE, and CORAS. While these methodologies are robust in traditional IT contexts, they exhibit significant limitations when applied to more dynamic and distributed technological environments, such as those characterizing smart infrastructures based on DS, as illustrated in Figure 2.1. The main identified limitations include:

- Low adaptability to heterogeneous contexts: These methodologies are typically oriented toward static systems with well-defined organizational boundaries. In contrast, DS infrastructures involve devices and services from multiple vendors, short lifecycle durations, and high operational variability, which require flexible and context-aware approaches.
- Absence of native automation: Traditional methods rely on manual or semi-automated risk analysis and assessment processes, rendering them ineffective in contexts where real-time response to dynamic threats is necessary, as occurs in DS ecosystems connected to critical or residential infrastructures.
- Lack of alignment with DS-specific standards: Although extensible, these methodologies were not originally conceived considering recent specialized DS security frameworks, limiting their capacity to cover all exposure vectors and current regulatory compliance requirements.

In contrast, the pattern proposed in this work, based on the MARISMA Framework [2.53] and grounded on recognized standards such as NIST SP 800-82 [2.52], SP 800-161 [2.50], ISO/IEC 27001:2023 [2.24], ISO/IEC 27002:2022 [2.51], ISO/IEC 27005:2022 [2.22], and ENISA guidelines [2.16, 2.17], offers a solution better suited to the intrinsic characteristics of DS environments. Its technical advantages include:

- Structured integration with specific regulatory frameworks: The pattern leverages international standards explicitly designed for DS environments and critical

infrastructures, enabling direct alignment with recommended security practices, regulatory requirements, and applicable technical controls.

- Adaptable and reusable risk model: The pattern's architecture allows for knowledge reuse through catalogs of threats, assets, controls, and previously modeled dependency relationships, adaptable to different scenarios.
- Continuous, real-time risk management: Unlike static methods, the pattern incorporates mechanisms for automating the risk management cycle, including tools for continuous threat identification and monitoring.
- Scalability and modularity: The modular approach of the pattern enables its application from domestic environments to critical infrastructures.
- Incorporation of cyberresilience as a key objective: The pattern not only focuses on threat prevention but also ensures the system's ability to respond and recover from incidents affecting operational continuity.

Approach	IoT Adaptability	Analysis Automation	Continuous Risk Management	IoT Threat Modeling	Connected Asset Management	Real-Time Assessment	Supply Chain Coverage	Updated Regulatory Integration	Advanced Control Management Automation and Effectiveness	Advanced Quantitative Risk Analysis Based on Historical and Statistical Data
MAGERIT (2012)	P	P	N	N	N	N	N	N	N	N
MONARC (2024)	Y	Y	N	Y	Y	N	N	N	N	N
OCTAVE (2007)	P	N	N	N	N	N	N	N	N	N
EBIOS (2020)	P	N	P	N	N	N	N	P	N	N
CORAS (2011)	P	P	P	N	N	N	N	N	N	N
MEHARI 2 (2010)	P	N	N	N	P	N	P	N	N	N
ENISA (2020)	Y	N	N	Y	Y	N	Y	Y	N	N
RESISTO (2018)	N	N	N	N	N	N	N	N	N	N
COBIT (2020)	N	N	N	N	N	N	N	P	N	N
IT-Grundschutz (2023)	P	Y	Y	N	Y	N	Y	P	Y	N
ISO/IEC 27005:2022	P	N	P	N	Y	N	P	Y	N	P
ISO/IEC 21827:2008	N	N	N	N	N	N	N	N	N	N
ISO/IEC 27400:2022	Y	N	N	Y	Y	N	Y	Y	N	N
ISO 31000:2018	P	N	Y	N	P	N	P	Y	N	N
BSI 7799-3 (2017)	N	N	N	N	N	N	N	N	N	N
BSI Germany Std 200-3 (2017)	N	N	N	N	N	N	N	N	N	N
NIST SP 800-30 (2017)	P	P	N	N	Y	N	P	Y	N	Y
NIST SP 800-37 (2018)	P	Y	Y	N	Y	Y	Y	Y	N	P

Capability	Technical Description
IoT Adaptability	Ability of the approach to adjust to IoT ecosystem characteristics (heterogeneous devices, ubiquitous connectivity, dynamic operations).
Analysis Automation	Ability to integrate or support automated tools for data collection, risk analysis, and mitigation.
Continuous Risk Management	Support for iterative or real-time processes, enabling continuous risk reassessment.
IoT Threat Modeling	Capability to represent and analyze threats specific to the IoT context (firmware, M2M networks, sensors, etc.).
Connected Asset Management	Support for identification, classification, and control of distributed IoT assets with limited capabilities.
Real-Time Assessment	Capability to perform risk assessments in dynamic operational environments, integrating telemetry or security events.
Supply Chain Coverage	Explicit inclusion of risk associated with third parties, integrators, and IoT hardware/software suppliers.
Updated Regulatory Integration	Alignment with updated standards and frameworks relevant for IoT (e.g., NIST 8259, ISO 27400, etc.).
Advanced Control Management Automation and Effectiveness	Ability of the approach to automate the application, monitoring, and continuous evaluation of security controls, systematically measuring their effectiveness.
Advanced Quantitative Risk Analysis Based on Historical and Statistical Data	Ability to employ sophisticated quantitative analyses, supported by historical data and statistical models, to accurately estimate and predict risks.

Figure 2.1: Coverage of key capabilities by traditional approaches in risk analysis and management for DS environments

In summary, the proposed approach represents not merely a methodological evolution over traditional frameworks but a structural transformation of risk analysis and management, specifically oriented to the technical, operational, and regulatory requirements of modern DS infrastructures. Its design supports both international regulatory compliance and the efficient technical implementation of protective measures in dynamic, distributed environments with highly specialized security requirements.

2.3 MARISMA Framework

MARISMA [2.53] is a risk analysis and management (RAM) framework adaptable to diverse technological environments, including those characterized by the heterogeneity and complexity of DS systems. This framework introduces the concept of a meta-pattern, a conceptual model that integrates security controls from the initial phases of risk analysis, enabling a proactive and structured approach to security management. The meta-pattern also facilitates the reuse of artifacts and the definition of specific patterns tailored to particular domains (see Figure 2.2), thereby increasing efficiency and consistency in the application of the framework.

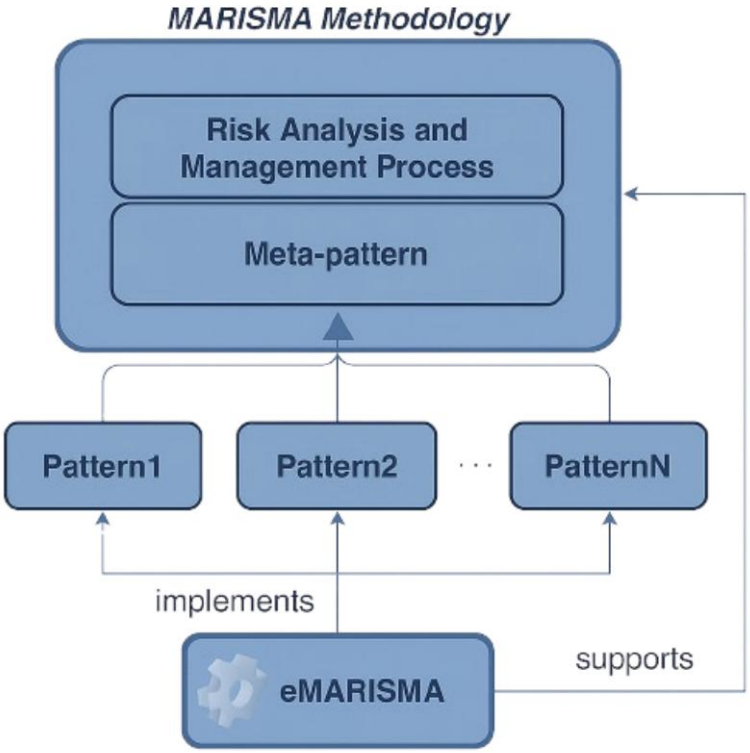

Figure 2.2: General schema of MARISMA framework

MARISMA's flexibility to adapt to different contexts is primarily supported by its ability to derive specialized patterns from the meta-pattern (see Figure 2.3). These patterns inherit a base structure that incorporates the fundamental elements of any RAM process, such as the taxonomy of assets, threats, and analysis dimensions, and are subsequently extended or modified to fit the specific characteristics of a given environment, such as DS.

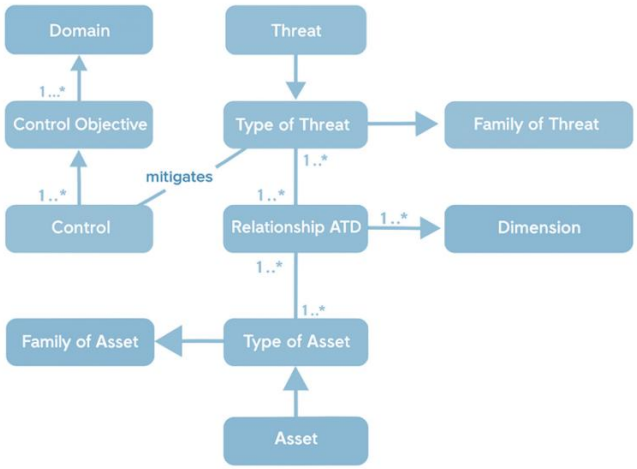

Figure 2.3: Meta-pattern of MARISMA

The development of a specific pattern for DS environments begins with a systematic review of scientific and technical literature, accompanied by an analysis of standards, recommendations, and best practices in security risk management. This process aims to identify relevant regulatory frameworks and applicable controls for these environments, as well as to establish a representative taxonomy of pertinent assets and threats.

The construction of the pattern has been carried out with the support of cybersecurity and risk analysis experts, whose knowledge has been essential to ensure the validity and applicability of the model in real-world environments. Specifically, five specialists from cybersecurity consulting firms participated, including senior security consultants, physical security experts, and junior consultants.

Once the essential elements of the pattern (assets, controls, threats, and dimensions) were defined, semantic and operational relationships among them were established, such as association matrices between assets, threats, and dimensions, or linkages between functional domains, threats, and security controls. These relationships allow for an accurate modeling of

the DS environment's exposure surface and facilitate the identification and prioritization of controls based on residual risk.

Following the pattern's construction, the next methodological step involves its practical application in a real-world setting, which entails instantiating the pattern in a specific use case. In this study, a smart home has been selected as the application scenario, with the goal of validating the framework's effectiveness, adaptability, and relevance within a domestic DS environment characterized by increasing automation, ubiquitous connectivity, and functional diversity of devices.

Applying the pattern requires, in a first stage, the detailed identification and characterization of the specific assets comprising the smart home's technological and operational infrastructure. Simultaneously, a comprehensive analysis of potential threats that could compromise both the logical and physical security of the environment is necessary. This adaptation process allows the pattern to map structural components such as assets, threats, evaluation dimensions, controls, and mitigation measures to the operational and risk-exposure characteristics specific to this type of infrastructure.

Pattern instantiation in a concrete environment must be performed by professionals with specialized knowledge in risk management and DS security, preferably members of the technical team responsible for implementing and maintaining the smart home infrastructure. These specialists are responsible for properly parameterizing the pattern elements, continuously managing the supporting tool, and adapting control measures in response to technological evolution, the emergence of new vulnerabilities, or operational configuration changes in devices and services.

This approach ensures dynamic, contextualized, and proactive risk management, enabling the system to adapt to emerging threats and maintain adequate levels of protection, operational resilience, and security for users and their digital and physical assets in the domestic environment. This claim is not only theoretical, as it has been validated in multiple previously defined patterns, such as MARISMA-CPS [2.46], which incorporates a set of reusable and adaptable elements to facilitate risk management and control in Cyber-Physical Systems; MARISMA-BiDa [2.45], aimed at addressing risks associated with the inherent characteristics of Big Data environments; and MARISMA-BP [2.47], designed to enable flexible, holistic, and executable security risk assessment and management in business process models, providing these models with complete, adaptable, and executable mechanisms to address risk comprehensively.

The proposed MARISMA-DS pattern enables adaptive, flexible, and scalable risk management specifically designed for DS environments. These environments, characterized by high heterogeneity, structural complexity, and scalability, pose significant challenges to existing

regulatory frameworks, recommendations, and guidelines, highlighting the need for specialized approaches that effectively address the particularities of this technological domain.

Finally, the MARISMA framework includes a cloud-based tool called eMARISMA, designed to optimize risk analysis and management. It facilitates dynamic management of evolving risks in a simple and efficient manner, significantly reducing analysis times. The tool uses a relational structure among the different elements involved in risk analysis and the necessary controls to manage security. These relationships are established based on knowledge acquired from previous implementations, enabling information reuse and cost reduction. Additionally, eMARISMA allows the establishment of associative and hierarchical dependencies among assets, so that if a shared asset is impacted, the risk for all associated assets, companies, projects, and standards is automatically recalculated. Therefore, eMARISMA serves as a comprehensive risk management tool that supports more secure and efficient decision-making.

2.4 Defining the pattern for IoT infrastructures in MARISMA-DS

The design of a methodological pattern for risk management in DS-based infrastructures requires the construction of a robust, systematic, and coherent conceptual architecture. This architecture must be capable of comprehensively addressing the identification, assessment, and mitigation of specific threats that arise in environments characterized by high interconnectivity, technological diversity, functional heterogeneity of devices, and an expanded attack surface that significantly increases opportunities for malicious exploitation.

In this work, the MARISMA-DS pattern has been developed, a specialization of the general meta-pattern for risk management, whose main purpose is to provide an adaptable, modular, and scalable methodological framework, specifically oriented to risk assessment in DS ecosystems. This pattern has been designed to ensure its transversal applicability across multiple contexts, including corporate infrastructures, smart residential environments, commercial facilities, and more. The meta-pattern constitutes a high-level abstraction that incorporates the generic elements necessary to represent any risk assessment and management process (RAM, Risk Assessment Methodology), regardless of the type of system or domain in which it is applied. The fundamental components of the meta-pattern—assets, threats, vulnerabilities, dimensions, controls, and mitigation measures—serve as base structures that must be instantiated and adapted to the specific context under analysis to be operational.

In the case of MARISMA-DS, the elements of the meta-pattern have been selected, adjusted, and implemented taking into account the technical, functional, and operational particularities of DS environments. This contextual adaptation has been guided by a detailed analysis of the most relevant international standards and technical recommendations, including:

- ISO/IEC 27001:2023 [2.24], for the establishment of an Information Security Management System (ISMS).
- ISO/IEC 27002:2022 [2.51], which provides guidelines and best practices for implementing information security controls within an ISMS framework.
- ISO/IEC 27005:2022 [2.22], for the specific management of information security risks.
- NIST SP 8259 [2.40, 2.41], focused on cybersecurity for DS devices.
- NIST SP 800-82 [2.52], addressing industrial control system security.
- NIST SP 800-161 [2.50], concerning risk management in the information and communications technology (ICT) supply chain.
- ENISA Baseline Security Recommendations for DS [2.17] and the ENISA Threat Landscape report [2.16], which provide a comprehensive compendium of threats, recommendations, and best practices specific to DS environments within the European Union.

Figure 2.4 illustrates the overall architecture of the MARISMA-DS pattern, which includes key components such as security domains, critical assets, analysis dimensions, and threat typologies. The following sections provide a detailed description of its main components.

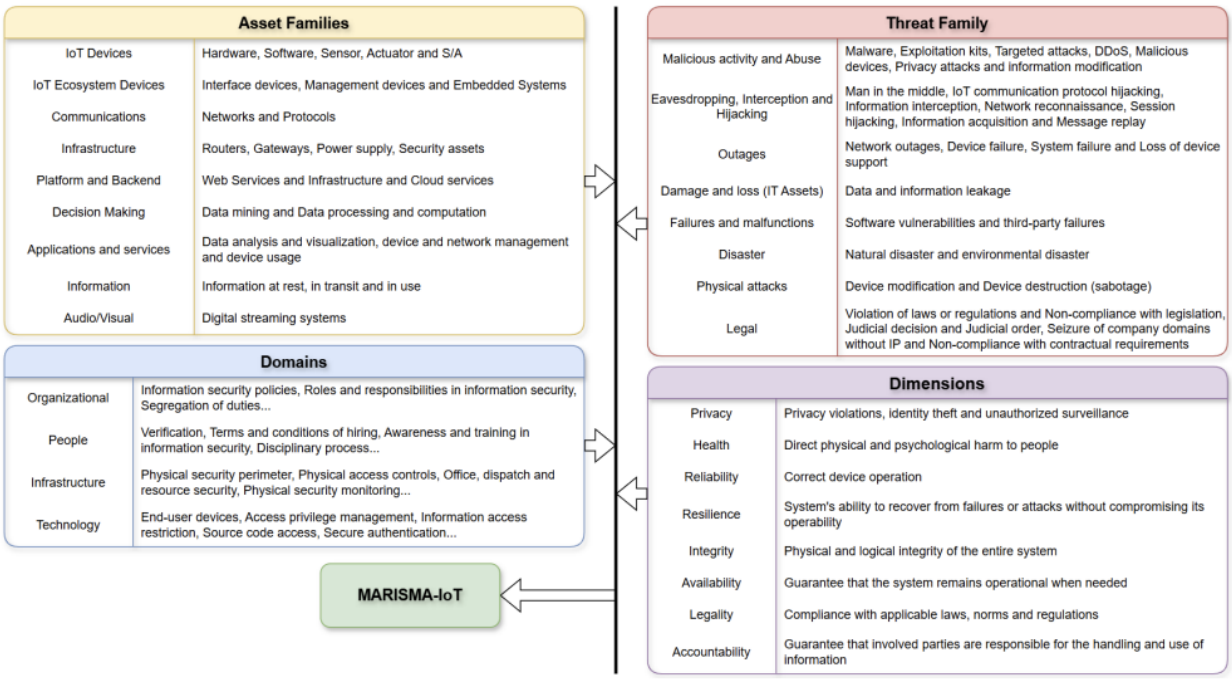

Figure 2.4: Components of the MARISMA-DS pattern

2.4.1 Security Domains and Objectives

The first structural layer of the MARISMA-DS pattern focuses on defining functional domains and security objectives, which serve as the backbone of the risk analysis. Its purpose is to provide a logical and coherent segmentation of the DS ecosystem, aligned with the different phases of the lifecycle of devices, services, and information flows in smart environments.

The selection and structuring of these domains are based on an analytical and contextual reinterpretation of established regulatory frameworks, emphasizing ISO/IEC 27001:2023 [2.24], which sets requirements for information security management systems (ISMS), and ISO/IEC 27005:2022 [2.22], which provides specific guidelines for information security risk management. Both standards have been adapted to the DS paradigm through an operational reading that considers the functional, topological, and technological particularities of these environments. In this context, ISO/IEC 27002:2022 [2.51] plays an instrumental role by offering detailed technical guidance for implementing security controls aligned with the objectives defined in ISO/IEC 27001.

Each domain identifies a relevant functional characteristic of the system (Organizational, Personnel, Infrastructure, and Technology) and is associated with specific control objectives, defined based on the inherent risks affecting that aspect. These objectives enable the establishment of proportional and contextualized safeguards, in line with the criticality of the involved assets and the available technical and organizational capabilities.

These objectives constitute concrete security goals, whose effective implementation ensures compliance with the fundamental properties of information and DS services: confidentiality, integrity, availability, authenticity, and traceability.

This modular and hierarchical organization enhances both the adaptability of the pattern to different application contexts (smart homes, offices, corporate buildings, domotic installations, etc.) and its horizontal and vertical scalability, enabling systematic, dynamic, and verifiable security management in heterogeneous DS infrastructures.

This conceptual layer is represented integrally with the other components of the pattern in Figure 2.4, which illustrates the overall architecture of the model and its methodological articulation.

Finally, the eMARISMA tool incorporates the definition and integration of all domains, objectives, and security controls in a structured manner, enabling centralized management within subsequent risk analyses.

2.4.2 Inventory and Asset Classification

The asset inventory constitutes the fundamental starting point for any risk analysis, as risk manifests only in relation to assets exposed to threats and vulnerabilities. In the context of a Smart Home, the digital ecosystem is composed of a wide variety of interconnected assets, whose classification is critical for defining the scope of evaluation and protection.

The characterization of each asset is carried out based on technical and operational criteria, including its relative value to the business or end user, operational criticality, degree of exposure to attack vectors, and the level of functional dependency within the overall system. This structured classification not only allows prioritizing protection efforts and resource allocation but also facilitates the identification of dependency relationships between assets. Such dependencies are essential for modeling chained failure scenarios or lateral movement attacks, which can compromise the integrity and availability of the entire ecosystem.

For the definition and classification of assets, the main references adopted are the ENISA Baseline Security Recommendations for DS [2.17] and the ENISA Threat Landscape and Good Practice Guide for Smart Home and Converged Media [2.16]. These guides provide an exhaustive and up-to-date taxonomy covering physical devices and embedded systems, as well as cloud services and network components, while also incorporating security and privacy considerations specific to intelligent residential environments. Integrating these standards ensures that the asset inventory aligns with international best practices, facilitates the interoperability of risk analysis models, and enhances coverage against emerging threats inherent to the DS domain.

The MARISMA-DS pattern proposes a multi-class asset taxonomy, as show in Figure 2.5.

Asset Group	Asset	Description
IoT Devices	Hardware	Physical components of the IoT device (microcontrollers, processors, physical ports, motherboards, etc.).
	Software	Includes the operating system, firmware, and installed or running applications.
	Sensors	Subsystems for detecting or measuring environmental events (temperature, motion, etc.).
	Actuators	Output units that execute decisions based on processed information.
	Sensors/Actuators	IoT devices that integrate sensing and actuation capabilities.
Other IoT Ecosystem Devices	Object Interface	Devices that act as an interface or aggregator for other IoT devices.
	Object Management	Devices designed to manage IoT devices and networks.
	Embedded Systems	Devices with processing capability, integrated sensors/actuators, and direct cloud connectivity.
Communications	Networks	Enable data exchange between nodes in the IoT ecosystem (LAN, PAN, WAN, etc.).
	Protocols	Communication rules between IoT devices (ZigBee, MQTT, CoAP, BLE, etc.).
Infrastructure	Routers	Route data packets between networks in the IoT ecosystem.
	Gateways	Nodes that enable interoperability between networks with different protocols.
	Power Supply	Provide electrical energy, either wired or via integrated battery.
	Security Assets	Protection components such as firewalls, WAF, CASB, IDS/IPS, authentication/authorization systems.
Platform and Backend	Web Services	Web-based services for H2M and M2M interaction.
	Cloud Infrastructure and Services	Collection, storage, processing, and remote management of data and devices.
Decision Making	Data Mining	Big data algorithms and services to transform data into actionable structures.
	Processing and Computing	Data processing for automation, machine learning, business rules, etc.
Applications and Services	Analytics and Visualization	Analysis and visualization of processed data to improve efficiency and detect patterns.
	Device and Network Management	Updating, monitoring, logging, and diagnosing devices and networks.
	Device Usage	Contextualization of status, usage patterns, and performance of the IoT ecosystem.
Information	At Rest	Information stored locally or in the cloud.
	In Transit	Information transmitted between IoT devices.
	In Use	Information actively processed by applications or services.
Audio/Visual	Multimedia Output Systems	Devices for playback, management, and streaming of digital content such as smart displays and sound systems.

Figure 2.5: Classification and Description of Assets in DS Environments

As with the Security Domains and Objectives, this asset taxonomy is also incorporated into the eMARISMA tool, serving as a basis for the characterization of assets under analysis during risk assessment.

2.4.3 Security Dimensions for Risk Assesment

Risk assessment in DS environments requires a multidimensional approach that enables comprehensive coverage of the various threat vectors and technological particularities associated with smart infrastructures. In this context, a set of security dimensions has been defined, inspired by the guidelines of ISO/IEC 27001:2023 [2.24] and the NIST frameworks 8259 [2.40, 2.41], 800-82 [2.52], and 800-161 [2.50], whose integration provides a robust analytical framework aligned with international best practices.

These dimensions structure risk assessment systematically and effectively, facilitating the identification, prioritization, and mitigation of potential impacts on the DS assets comprising the smart building. The defined dimensions in the MARISMA-DS pattern are detailed below:

- **Privacy**: DS devices collect, process, and transmit significant volumes of data, much of which may be associated with personally identifiable information (PII), such as images captured by surveillance systems, occupant behavior patterns, access credentials, or usage preferences. Improper exposure or handling of this data through interception, insecure storage, or unauthorized transfer can lead to severe privacy violations.
- **Health**: Beyond conventional cyber risks, certain DS environments introduce direct physical security threats arising from compromised devices that interact with the physical environment, potentially leading to hazardous situations and psychological impacts on occupants.
- **Operational Reliability**: Reliability depends on the functional and technical availability of multiple DS components supporting critical services. Systematic failures can directly affect service continuity.
- **System Resilience**: Resilience refers to the system's ability to withstand, absorb, adapt to, and recover from failures, attacks, or adverse conditions without compromising operability.
- **Integrity (Physical and Logical)**:
    - Physical integrity focuses on protection against physical tampering or damage to devices.
    - Logical integrity covers the accuracy, consistency, and reliability of processed and transmitted data.
- **Availability**: Refers the capability of DS systems and devices to remain operational when required.
- **Legality**: Concerns compliance with applicable legal and regulatory frameworks governing DS environments.
- **Accountability**: Refers to the obligation of organizations and operators to be answerable for their security, privacy, and operational management practices.

2.4.4 Types of Threats

The final structural layer of the pattern focuses on the identification and categorization of threats, an essential component for anticipating risk scenarios and prioritizing mitigation actions. Threats may arise from multiple vectors, both internal and external, and can originate intentionally (e.g., deliberate malicious actions by actors with economic, ideological, or destructive motivations) or accidentally (e.g., misconfigurations, human errors, or device malfunctions).

For systematic management, threats have been organized into conceptual families aligned with internationally recognized taxonomies in the field of critical infrastructure cybersecurity. In particular, an adaptation of the models proposed by ENISA has been adopted, contextualized to the smart home environment, which is characterized by high technological heterogeneity, an expanded exposure surface, and lower professionalization in security management.

To define and classify these threats, the primary references used were the ENISA Baseline Security Recommendations for DS [2.17] and the ENISA Threat Landscape and Good Practice Guide for Smart Home and Converged Media [2.16], which provide an updated and DS-specific threat framework (see Figure 2.6), with clear orientation toward the residential environment but adaptable to all contexts associated with DS devices.

The inclusion of threats in this layer is not merely descriptive; it constitutes the core of risk management in these environments. The presence and dynamic evolution of these threats necessitate a proactive risk management strategy, capable of identifying and mitigating vulnerabilities preventively, before they can be exploited. This anticipatory approach strengthens the security posture, reduces the likelihood of incidents, and minimizes the impact of potential security compromises on the system's critical assets.

CATEGORY	THREAT	DESCRIPTION
Nefarious activity / Abuse	Malware	Software programs designed to carry out unwanted and unauthorised actions on a system without the consent of the user, resulting in damage, corruption or information theft. Its impact can be high.
	Exploit Kits	Code designed to take advantage of a vulnerability in order to gain access to a system. This threat is difficult to detect and in IoT environments its impact ranges from high to crucial, depending on the assets affected.
	Targeted attacks	Attacks designed for a specific target, launched over a long period of time, and carried out in multiple stages. The main objective is to remain hidden and to obtain as much sensitive data/information or control as possible. While the impact of this threat is medium, detecting them is usually very difficult and takes a long time.
	DDoS	Multiple systems attack a single target in order to saturate it and make it crash. This can be done by making many connections, flooding a communication channel or replaying the same communications over and over.
	Counterfeit by malicious devices	This threat is difficult to discover, since a counterfeit device cannot be easily distinguished from the original. These devices usually have backdoors and can be used to conduct attacks on other ICT systems in the environment.
	Attacks on privacy	This threat affects both the privacy of the user and the exposure of network elements to unauthorised personnel.
	Modification of information	In this case, the objective is not to damage the devices, but to manipulate the information in order to cause chaos, or acquire monetary gains.
Eavesdropping / Interception / Hijacking	Man in the middle	Active eavesdropping attack, in which the attacker relays messages from one victim to another, in order to make them believe that they are talking directly to each other.
	IoT communication protocol hijacking	Taking control of an existing communication session between two elements of the network. The intruder is able to sniff sensible information, including passwords. The hijacking can use aggressive techniques like forcing disconnection or denial of service.
	Interception of information	Unauthorised interception (and sometimes modification) of a private communication, such as phone calls, instant messages, e-mail communications.
	Network reconnaissance	Passively obtain internal information about the network: devices connected, protocol used, open ports, services in use, etc.
	Session hijacking	Stealing the data connection by acting as a legitimate host in order to steal, modify or delete transmitted data.
	Information gathering	Passively obtain internal information about the network: devices connected, protocol used, etc.
	Replay of messages	This attack uses a valid data transmission maliciously by repeatedly sending it or delaying it, in order to manipulate or crash the targeted device.
Outages	Network Outage	Interruption or failure in the network supply, either intentional or accidental. Depending on the network segment affected, and on the time required to recover, the importance of this threat ranges from high to critical.
	Failures of devices	Threat of failure or malfunction of hardware devices.
	Failure of system	Threat of failure of software services or applications.
	Loss of support services	Unavailability of support services required for proper operation of the information system.
Damage / Loss (IT Assets)	Data / Sensitive information leakage	Sensitive data is revealed, intentionally or not, to unauthorised parties. The importance of this threat can vary greatly, depending on the kind of data leaked.
Failures / Malfunctions	Software vulnerabilities	The most common IoT devices are often vulnerable due to weak/default passwords, software bugs, and configuration errors, posing a risk to the network. This threat is usually connected to others, like exploit kits, and it is considered crucial.
	Third parties failures	Errors on an active element of the network caused by the misconfiguration of another element that has direct relation with it.
Disaster	Natural Disaster	These include events such as, floods, heavy winds, heavy snows, landslides, among others natural disaster, which could physically damage the devices.
	Environmental Disaster	Disasters in the deployment environments of IoT equipment and causing their imperability.
Physical attacks	Device modification	Tampering a device by for example taking advantage of bad configuration of ports, exploiting those left open.
	Device destruction (sabotage)	Incidents such devices theft, bomb attacks, vandalism or sabotage could damage devices.
Legal	Violation of laws or regulations / Breach of legislation	Non-compliance with applicable laws, regulations, or legal requirements that govern the operation, data processing, or security of IoT systems. This may result in legal penalties, fines, or operational restrictions.
	Judiciary decision / Court order	A legally binding directive issued by a court that requires specific actions to be taken by an organization, such as data disclosure, service shutdown, or compliance measures. Failure to comply may lead to legal consequences.
	No-IP company domains seizure	Legal action taken by companys or other authorities to seize or block domains managed by No-IP due to alleged abuse, fraud, or malicious activity. This can lead to service disruptions and loss of domain control for legitimate users.
	Failure to meet contractual requirements	Inability to comply with contractual obligations agreed upon with partners, customers, or service providers, leading to potential legal disputes, financial penalties, or termination of agreements.

Figure 2.6: Taxonomy of threats for DS environments

2.4.5 Domain-Objective-Threat Matrix in DS Infrastructures

Risk management in DS infrastructures requires a systematic approach that establishes explicit relationships between security domains, control objectives, and identified threats. To provide this structured perspective, a correlation matrix has been developed, enabling a clear visualization of how each domain and objective may impact various threats and security goals, as illustrated in Figure 2.7.

Domain	Objective	Threat Families							
		Malicious Activity	Eavesdropping / Interception	Outages	IT Damage / Loss	Failures / Malfunction	Disaster	Physical Attacks	Legal
Organizational	Information security policies	X	X	X	X	X	X	X	X
	Roles and responsibilities in information security	X	X	X	X	X	X	X	X
	Segregation of duties	X	X						X
People	Background checks	X							X
	Employment terms and conditions	X							X
	Awareness, education, and information security training	X	X		X	X			X
Infrastructure	Physical security perimeter				X			X	X
	Physical entry controls				X			X	X
	Office, workspace, and resource security				X			X	X
Technology	End-user devices	X	X		X				X
	Access privilege management	X			X	X			X
	Information access restriction	X	X		X	X			X

Figure 2.7: Segment of the Domains, objectives and Threat Families matrix

The domains and control objectives have been adapted from the ISO/IEC 27001:2023 [2.24] framework, ensuring alignment with international standards for information security management. This framework provides a solid foundation for establishing policies, procedures, and controls applicable to high-exposure technological environments, such as DS systems. Additionally, the threats considered in the matrix have been extracted from the reference sources previously specified in the earlier sections.

Given the extensive scope of the full matrix, this section presents an abbreviated version that includes a representative selection of correlations between threats and domains. For example, the “Infrastructure” domain, associated with the control objective “Physical access controls,” exhibits a strong correlation with threats such as Damage or Loss of IT Assets, Physical Attacks, and Legal Threats. This relationship is justified because the implementation of physical control mechanisms constitutes a direct preventive measure against intrusion, sabotage, or theft attempts. Moreover, physical access control also contributes to regulatory compliance regarding data protection and legal responsibilities, by ensuring that only authorized personnel can access critical infrastructure, thus reducing the risk of information leakage or unauthorized alterations to devices.

On the other hand, the “People” domain, linked to the control objective “Awareness, education, and training in information security,” maintains a significant correlation with threats such as Malicious Activity, Eavesdropping or Information Interception, Damage or Loss of IT Assets, Failures or Malfunctions, and Legal Aspects. This relationship is based on the fact that continuous reinforcement of personnel training in security enhances their ability to identify anomalous behavior, prevent operational errors, and act in accordance with the principles of confidentiality, integrity, and availability of information. Awareness of secure practices decreases the likelihood of negligent or inadvertent actions that could be exploited by malicious agents. Additionally,

periodic training ensures that employees are knowledgeable about and comply with current regulatory frameworks, thereby reducing the risk of legal noncompliance arising from ignorance or improper handling of information assets.

2.4.6 Asset-Threat-Dimensions Matrix in DS Infrastructures

In addition to the relationships established in the previous section, it is necessary to comprehensively analyze the relationship between system assets, the specific threats that may compromise them, and the security dimensions that define the various impacts on assets caused by these threats. To this end, a multidimensional correlation matrix has been developed that explicitly links asset types, relevant threat families, and the security dimensions affected in each case.

The construction of this matrix employed an expert knowledge-based approach, combining accumulated experience in applied cybersecurity, threat analysis, and risk management in complex infrastructures. This perspective allowed for a high level of granularity in characterizing the relationships between threat vectors, susceptible assets, and the expected impacts on each security dimension within the structural framework of previously defined domains.

Threat Families	Threat Types	Asset Families								
		IoT Devices	Other IoT Devices	Communications	Infrastructure	Platform and Backend	Decision Making	Applications and Services	Information	Audio/Visual
Malicious Activity	Malware	P, S, C, RL, I, D, L, RP	P, S, C, RL, I, D, L, RP			P, C, RL, I, D, L, RP				P, S, C, RL, I, D, L, RP
	Exploit Kits	P, S, C, RL, I, D, L, RP	P, S, C, RL, I, D, L, RP		C, RL, I, D, L, RP					P, S, C, RL, I, D, L, RP
	Targeted Attacks				P, C, RL, I, D, L, RP	P, C, RL, I, D, L, RP			P, S, C, RL, I, D, L, RP	
Eavesdropping / Interception	Man in the Middle	P, I, C, D, L, RP		P, I, C, D, L, RP					P, S, I, D, L, RP	P, S, I, C, D, L, RP
	IoT Communication Protocol Hijacking	P, S, I, C, RL, D, L, RP			P, I, C, RL, D, L, RP		P, S, I, C, RL, D, L, RP		P, S, I, D, L, RP	P, S, I, C, RL, D, L, RP
	Information Interception	P, I, C, RL, D, L, RP		P, I, C, RL, D, L					P, S, I, D, L, RP	P, S, I, C, RL, D, L, RP
Outages	Network Outage			C, RL, D, L	C, RL, D, L, RP					
	Device Failure	C, RL, D, L, RP								C, RL, D, L, RP
	Loss of Device Support	C, RL, D, L, RP	C, RL, D, L, RP	C, RL, D, L, RP	C, RL, D, L, RP	C, RL, D, L, RP	C, RL, D, L, RP	C, RL, D, L, RP	C, RL, D, L, RP	C, RL, D, L, RP

Figure 2.8: Segment of the matrix of Threat Families and Types, Asset Families and Dimensions

The Figure 2.8 specific connections between threat families and the most vulnerable dimensions of different types of DS assets. Although each threat can affect multiple assets, its impact is not homogeneous: some threats compromise only a single dimension, while others have broader

effects, simultaneously impacting reliability, traceability, legality, or even the operational resilience of the asset.

It is worth noting that, in the proposed model, every threat is conceptualized as an action aimed at reducing the value of the affected asset, where the value is understood as a weighted combination of its security dimensions. Therefore, the nature and extent of the impact depend on the inherent properties of the asset and the type of threat involved.

For instance, Figure 2.8 shows that the threat family “Outages,” specifically the threat type "Network outage," has a high impact on communication and infrastructure assets, directly affecting the following dimensions:

- Reliability (C): Network interruption can compromise the stability of interconnection devices (routers, gateways, switches), causing intermittent or sustained failures in data transmission.
- Resilience (RL): A prolonged outage may significantly hinder the system’s ability to recover or adapt to the incident, reducing operational continuity
- Availability (D): By preventing connectivity between IoT nodes or with backend servers, there is a direct loss of access to services dependent on that infrastructure.
- Legality (L): Service interruption may result in noncompliance with legal or regulatory requirements related to the availability of critical services, particularly in regulated sectors.
- Accountability (RP): Managing this threat requires proactive monitoring mechanisms, contingency plans, and escalation procedures; the absence of these could result in failures in accountability or contractual obligations.

Continuing within the “Outages” threat family, the threat type "Device failure" significantly impacts assets classified as IoT Devices and Audio/Visual components, compromising multiple security dimensions:

- Reliability (C): Functional failure of devices can cause erratic or non-deterministic behaviors in the IoT ecosystem, reducing confidence in the correct execution of critical tasks, such as the activation of sensors, actuators, or notification systems.
- Resilience (RL): A failure without proper redundancy or fault-tolerance mechanisms can generate sustained interruptions in the operation of affected devices, degrading the system’s ability to maintain functionality during internal disturbances and compromising recovery or adaptation capabilities.
- Availability (D): Partial or total unavailability of these devices leads to a direct loss of access to associated functionalities, such as monitoring physical environments

(temperature, motion, audio/video), especially when devices are part of a distributed control or surveillance subsystem.

- Legality (L): In certain contexts (e.g., healthcare, educational, or regulated surveillance environments), the lack of device availability may result in non-compliance with sectoral regulations that mandate minimum operational guarantees for electronic monitoring and protection systems.
- Accountability (RP): The absence of traceability in the device lifecycle management (maintenance, diagnostics, replacement) can lead to contractual deficiencies or difficulties in assigning responsibilities to third parties, particularly in IoT architectures involving multiple vendors and integration layers.

This matrix provides a robust analytical framework for identifying causal relationships between threats and functional degradation, facilitating the design of compensatory controls and preventive measures aimed at preserving the overall security of the DS ecosystem.

2.5 Conclusion and Future Work

There are well-established solutions that partially address this issue. Wazuh [3.26], an open-source platform, offers notable capabilities in Security Information and Event Management (SIEM), log management, and regulatory compliance. It also includes vulnerability detection based on the software inventory; however, it has limitations due to the absence of active scanning and advanced predictive prioritization mechanisms. In contrast, Nessus [3.23], in its commercial version, excels in active system scanning and risk prioritization using metrics such as CVSS, VPR (Vulnerability Priority Rating), and EPSS (Exploit Prediction Scoring System); however, it lacks integrated asset management and contextualized predictive modeling. Table 1 summarizes this comparison; unlike Wazuh/Nessus, VulnQ unifies asset-aware risk aggregation with forward-looking prediction in the same operational workflow.

In this work, a specialized methodological pattern for DS environments has been developed, based on the architecture of the MARISMA framework (Information Systems Risk Analysis and Management Framework), with the objective of providing a formal, adaptable, and extensible tool for risk management in distributed cyber-physical ecosystems.

This pattern, termed MARISMA-DS, is structured around four key taxonomic catalogs (Security Controls, System Assets, Potential Threats, and Affected Security Dimensions), which have been designed and organized in accordance with the principles established by the most relevant international standards in the field, including ISO/IEC 27001:2023, ISO/IEC 27002:2022, ISO/IEC 27005:2022, NIST 8259, NIST SP 800-30, and technical documents from ENISA.

A primary contribution of the pattern lies in the definition of dependency matrices among these elements, which enable the explicit modeling of relationships between assets, threats, controls,

and affected dimensions. These matrices not only provide a basis for formal risk analysis but also promote the systematic reuse of knowledge generated in different scenarios, facilitating adaptation to new contexts through inference, inheritance, and specialization processes.

The integration of the MARISMA-DS pattern into the MARISMA technological platform has been empirically validated through its application in practical use cases, demonstrating its technical feasibility and its capacity to accurately and operationally represent the risk factors present in interconnected intelligent infrastructures.

In the medium term, the evolution of the pattern toward specialized sectoral versions is envisioned, allowing the generation of derived sub-patterns adapted to specific DS domains, such as smart cities, hospitals, connected offices, or smart factories.

Thanks to the structural inheritance, modularity, and component reuse capabilities offered by the core MARISMA model, this adaptation can be carried out without a complete reconstruction of the pattern, maintaining methodological consistency while adjusting key elements (asset typology, attack vectors, applicable regulations, etc.).

This strategy significantly enhances efficiency in risk modeling and maximizes the optimization of both technical and human resources, as it enables the systematic reuse of previously validated configurations in new and diverse scenarios with a high degree of reliability and confidence. By reducing the need to start risk assessments from scratch, organizations can accelerate decision-making processes, improve consistency in their analyses, and ensure that accumulated knowledge is effectively capitalized upon. Consequently, the MARISMA-DS pattern not only serves as a robust and valuable instrument for conducting current risk analysis but also positions itself as an evolutionary and sustainable framework, capable of adapting to technological progress and addressing the dynamic and emerging security challenges inherent to the ever-expanding DS device ecosystem.

3 Predictive Evidence-Based Security Threat Management System Leveraging CWEs, CVEs, And CAPECs

3.1 Introduction

In recent years, the accelerated digital transformation across multiple sectors has significantly increased organizations' exposure to cyber threats [3.20]. Critical domains such as healthcare, energy, finance, transportation, and government services have experienced sustained growth in both the frequency and sophistication of cyberattacks [3.12, 3.13, 3.15]. Recent research highlights an exponential rise in the number of recorded vulnerabilities, accompanied by the proliferation of offensive campaigns [3.3, 3.7, 3.19]. These incidents not only compromise the confidentiality, integrity, and availability of information but also disrupt operational continuity, cause significant financial losses, and erode public trust.

The economic impact of cybercrime is projected to reach trillions of dollars annually, ranking it among the largest global economies when measured in financial terms [3.4, 3.9]. Simultaneously, the attack surface continues to expand due to the massive adoption of cloud services, Internet of Things (IoT) devices, and interconnected industrial control systems [3.2].

This increasing complexity poses serious challenges to traditional reactive defense mechanisms, underscoring the urgent need for proactive and predictive approaches capable of mitigating risks before they materialize [3.5, 3.16].

In this context, vulnerability management and risk prediction have become strategic priorities. International standards and taxonomies such as Common Vulnerabilities and Exposures (CVE), Common Weakness Enumeration (CWE), Common Attack Pattern Enumeration and Classification (CAPEC) and Common Vulnerability Scoring System (CVSS) provide structured frameworks for describing and analyzing weaknesses and threats. However, current tools mainly focus on detection and remediation, offering limited capabilities for forecasting the evolution of risk over time [3.21]. Overcoming this limitation requires the integration of machine learning models with vulnerability databases, enabling the generation of actionable intelligence, anticipation of potential threats, optimization of resource allocation, and strengthening of organizational cyber-resilience [3.6].

To address these limitations, this work proposes the development of a vulnerability and asset management system named VulnQ, implemented as a web application designed to facilitate the assessment and continuous monitoring of cybersecurity risks.

3.2 Related work

Cybersecurity risk management has become a continuous and complex process due to the expanding attack surface and the rapid increase in the publication of vulnerabilities. Although

the emergence of new standards and regulatory frameworks such as ISO/IEC 27001 [3.11], NIST [3.18] CSF [3.17], NIS2 [3.8], and GDPR [3.24] has provided organizations with structured responses to the growing threat landscape, most of these current standards remain focused on reactive incident management, while predictive approaches are still in an emerging phase [3.10, 3.22, 3.25].

The scientific and technical community relies on standardized taxonomies —CPE, CVE, CWE, CAPEC, and CVSS— that enable the interoperability of security data and the linkage of vulnerabilities with asset inventories [3.1, 3.27]. However, the use of static scoring systems such as CVSS limits the consideration of dynamic and contextual factors. In response, recent research has proposed alternative methods to mitigate these limitations [3.14].

3.2.1 Existing tools

There are well-established solutions that partially address this issue. Wazuh, an open-source platform, offers notable capabilities in Security Information and Event Management (SIEM), log management, and regulatory compliance. It also includes vulnerability detection based on the software inventory; however, it has limitations due to the absence of active scanning and advanced predictive prioritization mechanisms. In contrast, Nessus, in its commercial version, excels in active system scanning and risk prioritization using metrics such as CVSS, VPR (Vulnerability Priority Rating), and EPSS (Exploit Prediction Scoring System); however, it lacks integrated asset management and contextualized predictive modeling. Figure 3.1 summarizes this comparison; unlike Wazuh/Nessus, VulnQ unifies asset-aware risk aggregation with forward-looking prediction in the same operational workflow.

Feature	Wazuh	Nessus	VulnQ
Focus and functionality	SIEM, intrusion detection, vulnerability management	Vulnerability scanner	Unified management of assets, vulnerabilities, and risks
Detection technique	Software inventory and correlation with CVE	Active scanning of systems and services	Software inventory and correlation with CVE
Vulnerability classification	Rules and event integration	CVSS, VPR, and automated recommendations	Custom risk score based on CVSS and calculated KRI
Distribution model	Open source	Commercial, limited free version	Commercial, limited free version
Integrations	SIEM and external security sources	Tenable solutions and other platforms	REST API

Figure 3.1: Comparison betweenWazuh, Nessus, and VulnQ

3.3 Development

The following section presents the design and implementation process of the proposed system. This development encompasses the definition of its modular architecture, the integration of databases, prioritization algorithms, and machine learning-based prediction models.

Additionally, it details the main functional components of the platform, as well as the technologies used for its construction and deployment.

3.3.1 System base

The proposed system is structured as a modular architecture comprising three primary layers: the application server, the user interface, and the prediction engine.

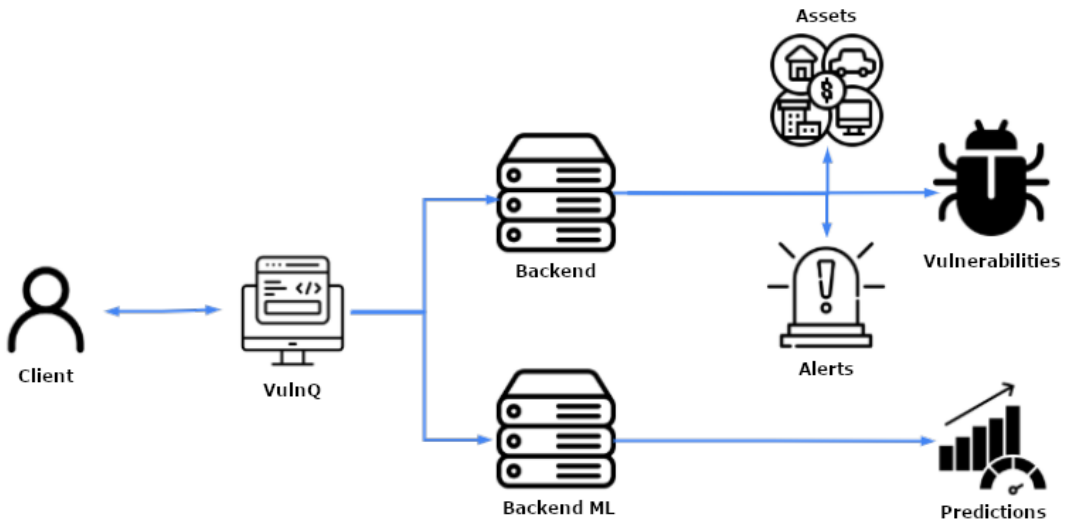

Figure 3.2: Architecture of the Proposed Solution

Figure 3.2 presents the architecture of the proposed solution, developed under the Software as a Service (SaaS) model. The design is initially oriented toward the Business-to-Business (B2B) sector, as the technical nature of the data requires users with specialized cybersecurity knowledge. Nevertheless, a progressive evolution toward non-specialized user profiles is envisioned through improvements in accessibility and interface simplification.

First, the application server acts as an intermediary between the client and the database, managing communication, requests, and responses (Figure 3.2, Backend component). Its main function is to coordinate the information flow, ensure data integrity, and enforce the business rules defined for asset and vulnerability management.

Second, the user interface is implemented as a web client (Figure 3.2, VulnQ component) that enables direct interaction with the system. This layer is designed to provide an intuitive and accessible experience through dynamic components that facilitate vulnerability queries, asset audits, and the visualization of risk metrics.

Finally, the prediction engine corresponds to the module responsible for processing both historical and current data to estimate the future evolution of risk (Figure 3.2, BackendML component). This component is based on machine learning techniques, allowing it to analyze vulnerability patterns and generate predictions regarding asset exposure levels over defined time

periods. The engine exposes specific services that can be consumed by the application server, thereby integrating its results into the overall system workflow.

This layered architecture ensures greater scalability, maintainability, and flexibility, allowing each component to evolve independently and adapt to new functional or technological requirements.

3.3.2 Database and search engine

The system's data architecture is based on a relational repository that integrates multiple standardized vulnerability catalogs. Specifically, dedicated schemas have been defined for the CVE, CWE, CAPEC, and CPE dictionaries. These schemas enable the consistent structuring of information, ensuring traceability and proper linkage between vulnerabilities, weaknesses, attack patterns, and the affected technological platforms.

To populate the database, an Extract, Transform, and Load (ETL) process has been developed to automate the collection of information from official sources. This process downloads the published files in structured formats, transforms them according to the defined model, and periodically imports them into the system. In this way, the repository remains up to date and accurately reflects the most recent state of the threat ecosystem.

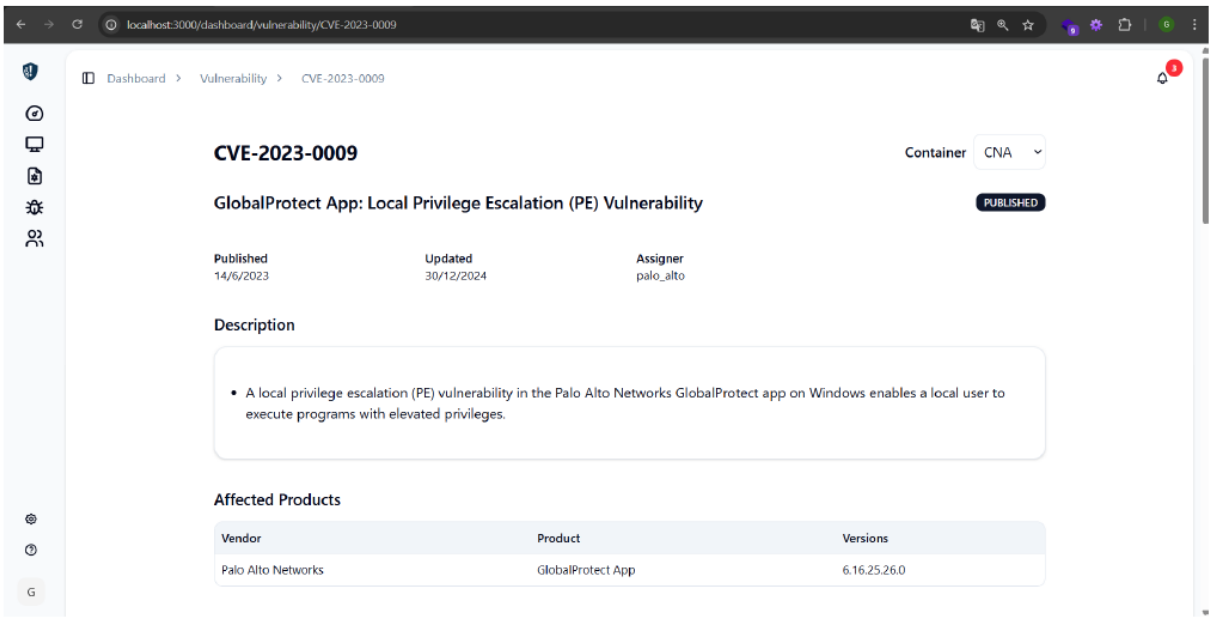

Figure 3.3: Visualization of Vulnerabilities in VulnQ

The vulnerability query module (Figure 3.3) enables searches based on standardized identifiers (e.g., CVE). This module facilitates direct user interaction with the database and forms the core of the management system, as it provides fast and structured access to critical security.

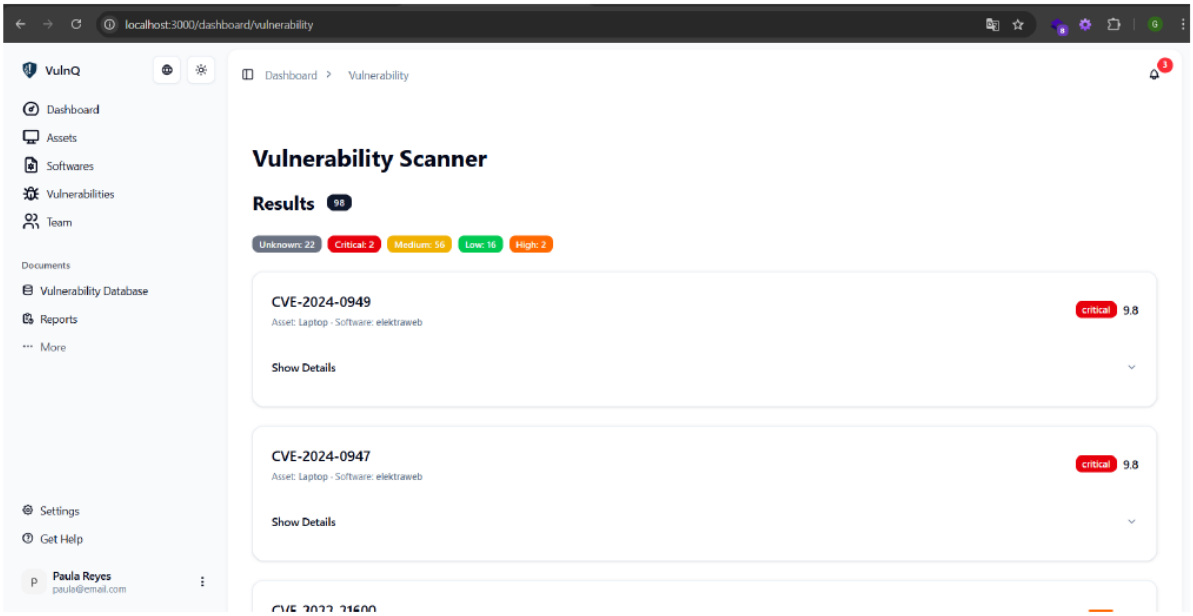

Figure 3.4: VulnQ Vulnerability Scanner

Additionally, the system integrates an audit interface that displays the results of analyzes performed on registered assets (Figure 3.4). This interface correlates detected vulnerabilities with inventoried devices and applications, offering a consolidated view of exposure levels and supporting the prioritization of mitigation measures.

3.3.3 Prioritization and alerts

The system incorporates a vulnerability prioritization mechanism based on standardized severity metrics. In particular, it adopts the reference scoring from vulnerability assessment systems (e.g., the CVSS base score) as the primary indicator, given its broad recognition and ability to unify evaluation criteria.

Building upon this metric, a prioritization algorithm is implemented to organize and filter the results obtained from analysis and audit processes. This algorithm applies auxiliary sorting and classification functions to structure the information according to risk level, thereby facilitating the identification of critical vulnerabilities that require immediate attention.

In parallel, an alert model has been defined with the purpose of automatically notifying the occurrence of relevant system events. Such events include both the detection of vulnerabilities associated with registered assets and incidents that may compromise the correct functioning of the platform.

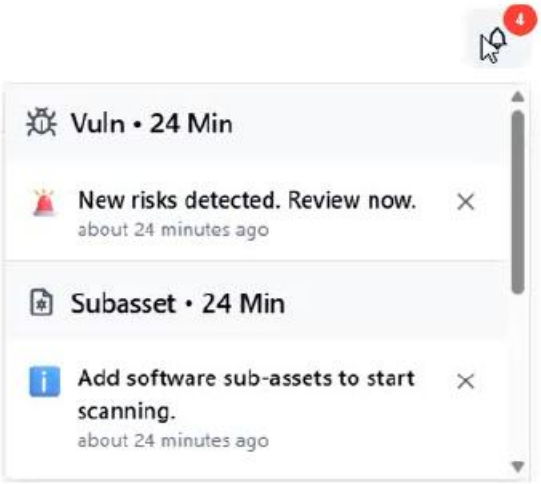

Figure 3.5: Alerts of the VulnQ System

The event-based alert mechanism enables real-time notifications, ensuring that users receive timely information about critical situations (Figure 3.5). For presentation purposes, an alert visualization interface has been developed, designed according to software architecture patterns that separate business logic from graphical representation. This component facilitates the consultation and tracking of notifications, providing a clear and structured user experience.

3.3.4 Risk estimation (ML)

The system incorporates a predictive risk estimation module based on Key Risk Indicators (KRI), designed to anticipate the exposure level of technological assets. This process relies on standardized attributes from CWE, CAPEC, and CVE catalogs to compute three distinct KRI scores.

For CWE, the KRI value is assigned using a heuristic scale determined by the likelihood of exploitation: high probabilities are rated with a maximum score of 10, medium with 6, low with 3, and a default value of 5 in any other case. For CAPEC, the KRI is calculated by multiplying two factors —likelihood of attack and typical severity— both mapped to normalized values (e.g., 1.0 for "High" or "Very High", 0.6 for "Medium", 0.3 for "Low", and default fallback values). The result is then scaled by a factor of 10 to obtain the final KRICAPEC score.

Finally, the CVE entry is computed as a weighted sum of four variables: the maximum CVSS base score (60%), the exploit count (10%), the previously calculated KRICWE (10%), and KRICAPEC (20%). This composite metric enables dynamic, context-aware risk estimation that supports more informed prioritization and proactive mitigation strategies.

These metrics are stored in a historical dataset used to train a regression-based predictive model that estimates the monthly evolution of asset risk. This model enhances traditional vulnerability management by enabling proactive and context-aware mitigation strategies.

3.3.5 Training and prediction

The system's predictive module focuses on building a model capable of anticipating the evolution of technological asset risk based on historical data (Figure 3.2, BackendML component).

In the training phase, a regression-based approach is applied to the historical dataset of KRI. This process captures the relationships among vulnerabilities, weaknesses, and attack patterns, as well as their cumulative impact on assets over time. The use of historical metrics as input ensures that the model incorporates both past trends and the temporal dynamics of threat exposure.

Subsequently, in the prediction phase, the model generates monthly risk estimates per asset. These projections are derived from the results of audits conducted and their correlation with the computed KRI values, allowing the system to anticipate short-term changes in exposure levels. The predictions are then integrated into the system's dashboard, providing security managers with a decisionsupport tool for preventive action and strategic planning.

3.3.6 Model integration and testing

The machine learning model was integrated into the system's overall architecture through the service layer, enabling direct communication with the other functional components. This integration followed a structured approach that ensures a clear separation between business logic and predictive logic, maintaining consistency with modular design principles.

As part of this integration, management operations were implemented to enable querying, storing results, and dynamically updating predictions. These operations provide a standardized interaction framework with the model, facilitating its incorporation into the system's workflows and its reuse in future analytical scenarios.

Finally, end-to-end validation tests were conducted to verify both the accuracy of the predictions and the proper communication among the different modules. These tests confirmed the system's capability to process historical data, generate risk estimates, and present the results in an integrated manner within the cybersecurity dashboard.

3.3.7 Visualization

The system incorporates a data visualization module designed to provide a consolidated view of the security status of technological assets. To achieve this, overview-type charts have been developed to clearly and succinctly represent critical information related to vulnerabilities, weaknesses, and risk levels.

The generation of these views relies on an ETL process that consolidates data from multiple catalogs and audit results. This process ensures the consistency and coherence of the information displayed in the interface, allowing users to access updated and reliable visual representations at all times.

Figure 3.6: VulnQ System Dashboard

In this way, in Figure 3.6, the system offers a visual analytics tool that enhances the interpretation of key indicators and improves analytical capability by enabling the rapid identification of trends and the prioritization of cybersecurity actions.

3.4 Conclusions

An information system has been developed to support organizational cybersecurity and asset management needs. It maintains a hierarchical asset inventory with full CRUD management and incorporates automated retrieval of official vulnerability data. An auditing module links assets to their associated vulnerabilities, generating categorized reports and alerts. Additionally, the system provides a dashboard with key metrics, including threat severity trends and monthly risk forecasts derived from a machine learning model. All functionality is delivered through a unified web interface, deployed with automated processes to balance operational efficiency and information quality. VulnQ meets the proposed objectives; however, to compete with established market tools, several development lines have been identified to enhance its precision, scalability, and applicability in various environments.

Future work will add continuous retraining feedback loops, expand risk features, and migrate the platform to a scalable microservices architecture with event-driven messaging and stronger security (e.g., MFA). VulnQ will be generalized beyond Smart Homes to smart city, ICS, and healthcare IoT domains via a federated layer enabling edge inference, secure model aggregation, and privacypreserving computation under intermittent connectivity. Crossdomain validation on multi-asset testbeds will report scalability, generalizability, and privacy/security metrics. The current regression model will serve as an interpretable baseline for comparison with Random Forest, XGBoost, and LSTM models under edge and federated deployment constraints.

4 Risk Management Middleware

4.1 Analysis of external system through HTTP traffic inspection

With the aim of automating the incident management process on the eMarisma external platform, a dynamic analysis of HTTP traffic generated during normal user interaction with the web application. Given that the system does not have a documented public API, it was necessary to empirically identify the endpoints, parameters, and call sequences that make up the complete business flow.

To do this, the BurpSuite and FoxyProxy tools were used, which allowed the interception and analysis of HTTP/HTTPS requests between the browser and the server.

4.1.1 Capture of the authentication flow

One of the main challenges of the analysis was capturing the authentication process authentication process, as the system uses reCAPTCHA as a protection mechanism. Initially, the intercepted requests contained invalid validation tokens, which prevented the login flow from being reproduced.

This problem was solved by:

- Installing the BurpSuite root certificate in the browser, allowing HTTPS traffic inspection.
- Configuring TLS passthrough, which enabled the complete capture of the parameters associated with the authentication process.

Once this configuration was applied, it was possible to correctly identify the endpoint responsible for the final validation of credentials, as well as the behaviour of the system upon a successful login, which is materialised by an HTTP redirect response (code 302) to the next step in the flow.

4.1.2 Identification of the complete functional flow

After authentication, the behaviour of the application was analysed while navigating through the different functional modules. The analysis made it possible to reconstruct the complete incident management flow, which includes:

- Loading projects and sub-projects.
- Registering a new incident.
- Consulting incidents.
- Association of threats, assets, and controls.
- Closing the incident and recalculating risk analysis.

This flow is not executed through a single call, but through a strict sequence of HTTP requests, which must be executed in the correct order for the system to function properly.

4.1.3 Summary of identified endpoints

Table 4.1 summarises the main endpoints identified during the analysis, grouped by functional phase. This abstraction allows us to focus on the architecturally relevant elements.

Phase	**Endpoint**	**Method**	**Main function**
Authentication	/login/authenticate	POST	Credential validation and session creation
Project upload	/proyecto/cargarProyectosTabla	GET	Retrieving the list of projects
Subproject upload	/subproyecto/cargarSubproyectosTabla/{subproject_id}	GET	Retrieving associated subprojects
New incident	/evento/save	POST	Creation of a new incident
Incident query	/evento/cargarEventoTabla/{subproject_id}	GET	List of incidents associated with a subproject
Add threat	/incidente/guardarAmenaza/{id_amenaza}	POST	Association of a threat to an incident
Severity management	/incidente/guardarGravedad	POST	Assigning severity to the incident
Asset linking	/incidente/vincularActivo	GET	Asset linking affected
Linking controls	/incidente/vincularControl	GET	Linking security controls
Incident conclusion	/evento/conclusion/{id_evento}	GET	Access to conclusion form

Incident storage	/evento/save/{id_evento}	POST	Persistence of the solution and event closure
Risk recalculation	/RSA/recalculateRAjax/{context_id}	POST	Recalculation of risk metric for the system

Table 4.1: identified endpoints

4.2 Middleware design

4.2.1 Architecture summary

The project is a REST API developed with Python and FastAPI, designed to act as an automation and orchestration layer on top of an external risk management system, Emarisma. Its main function is to receive simplified incident reports, validate them, and translate these reports into a complex sequence of operations in the external system.

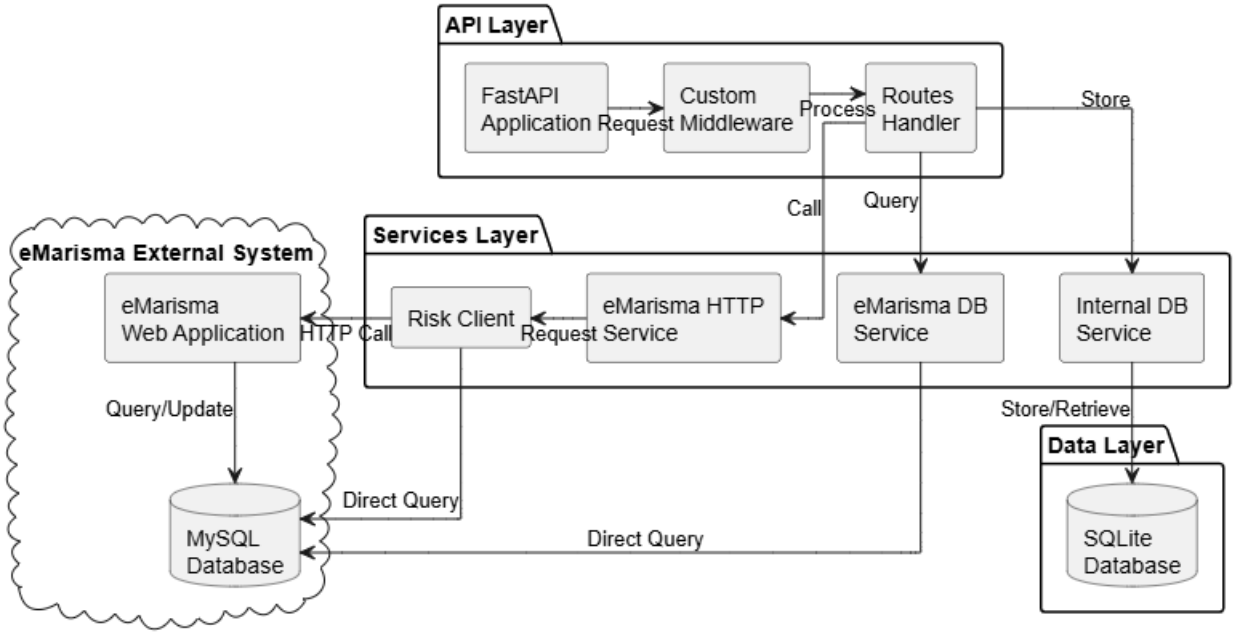

Figure 4.1: Architecture Diagram

4.2.2 Architectural Overview

Figure 4.1 depicts a layered service-oriented architecture designed to mediate between an external eMarisma environment and an internal persistence layer while exposing a clean REST interface. The architecture is structured into three main layers—API Layer, Services Layer, and

Data Layer—with an additional External System boundary representing the eMarisma platform and its database. This separation enforces modularity, improves testability, and isolates infrastructure concerns (HTTP, databases, middleware) from business workflow logic.

At runtime, the system receives HTTP requests through a FastAPI application, applies global request/response middleware, dispatches requests to route handlers, and then delegates the execution to domain services. These services orchestrate two distinct kinds of interactions: (i) HTTP calls to the eMarisma web application/API (via an asynchronous client wrapper), and (ii) direct database queries against the eMarisma MySQL database for identifier resolution and lookup operations. In parallel, selected request metadata and execution traces may be stored in an internal SQLite database to provide auditing and traceability.

4.2.2.1 API Layer

FastAPI Application

The FastAPI application acts as the system's ingress point and composition root. It is responsible for:

1. **Application instantiation and configuration**: creation of the FastAPI instance and registration of global configuration objects (routers, middleware, dependency providers).
2. **Lifecycle management**: initialization and teardown of shared resources required across requests, primarily:
    a. the **internal database connection/session** (for SQLite persistence), and
    b. the **asynchronous HTTP client session** used by the Risk Client to communicate with the external platform.
3. **Cross-cutting concerns integration**: attachment of the custom middleware component that intercepts requests and responses, enabling centralized logging, correlation IDs, and timing/monitoring.

By consolidating these responsibilities in the entry point, the remaining modules can remain focused on application logic rather than resource provisioning.

Custom Middleware

The middleware is executed before the request reaches the route handler and after the response is produced. Its function is to implement cross-cutting concerns consistently across endpoints, including:

- structured logging (request start/end, status code, execution time),
- observability hooks (metrics or traces),
- propagation of correlation identifiers for end-to-end tracking,
- uniform error shaping (where applicable).

This ensures that monitoring and operational requirements do not leak into business services and controllers.

Routes Handler

The routes layer implements the HTTP-facing contract of the system and serves as the controller layer. It is responsible for:

- defining public REST endpoints (paths, HTTP methods, status codes),
- validating and parsing request payloads using **Pydantic** models,
- enforcing basic request invariants (types, required fields, and schema constraints),
- mapping validated input into service calls, and
- assembling response models and HTTP semantics.

Importantly, the routes handler does not encode workflow logic; instead, it delegates to services through dependency injection (e.g., injection of RiskClient, DB services, or orchestration services). This controller/service separation avoids duplicating logic across endpoints and improves unit-test isolation.

4.2.2.2 Services Layer

The Services Layer encapsulates domain logic and integration logic. It is split into specialized modules to separate orchestration from low-level access to external resources.

eMarisma HTTP Service

This service contains the main business workflow logic. In the diagram, it corresponds to the "eMarisma HTTP Service" component that receives calls from the API routes and orchestrates a sequence of dependent actions against the external system. Concretely, it executes an ordered flow (e.g., run_all_flow) that may include operations such as:

- authentication and session establishment with the external system,
- navigation or context selection (e.g., selecting a project/workspace),
- creation or update of threat/risk-related entities,
- linking assets to threats or controls,
- triggering recalculation or refresh operations, and
- retrieving final status or computed risk values.

This orchestration is implemented as a deterministic sequence because later actions depend on artifacts produced by earlier actions (e.g., identifiers, session state, project context, or recalculation results). The service therefore acts as the "process coordinator" of the architecture.

Risk Client

The Risk Client is an abstraction over the raw asynchronous HTTP client. It centralizes all low-level concerns required to communicate reliably with the external eMarisma web application/API:

- endpoint path construction and request formatting,
- header and cookie handling (including authentication tokens),
- session reuse (connection pooling, keep-alive),

- retryable error classification (if implemented),
- serialization/deserialization boundaries for external payloads.

By isolating HTTP primitives inside the client wrapper, the orchestration service can express workflows in domain terms ("authenticate", "register threat", "recalculate") rather than "POST /x with headers y". This reduces coupling to external API specifics and localizes changes when the external platform evolves.

eMarisma DB Service

The eMarisma DB service provides read-oriented access to the external MySQL database for identifier and metadata resolution. In the diagram, it corresponds to "eMarisma DB Service" and the "Direct Query" arrows connecting the services layer to the external MySQL database.

Its key responsibility is **translation of human-meaningful names into internal identifiers** required by the external system (e.g., resolving project names, device names, asset identifiers, or other entity keys). This is particularly important when the external HTTP interface requires numeric IDs or opaque internal keys that are not readily available from user input.

Although direct database access increases integration power and can reduce the number of HTTP round trips, it must be handled carefully: the service should remain strictly scoped to lookup queries and avoid modifying the external database to prevent consistency issues and preserve ownership boundaries.

Internal DB Service

The internal database service manages persistence into the local SQLite database shown in the Data Layer. Its primary purpose is to provide:

- **auditability** (recording incoming requests, timestamps, and outcomes),
- **traceability** (storing correlation IDs, execution steps, or external references),
- operational diagnostics (capturing errors or external call summaries).

This service decouples operational storage from business logic, enabling compliance-oriented logging without contaminating workflow code. In the diagram, this corresponds to the "Internal DB Service" and the "Store/Retrieve" relationship to the SQLite database.

4.2.2.3 Data Layer

Internal SQLite Database

The system includes a local SQLite persistence layer used for lightweight operational storage (e.g., request logs, audit events, execution traces, or cached artifacts when needed). SQLite is appropriate here due to its minimal operational overhead and ease of embedding for a service that primarily needs traceability rather than high-throughput transactional workloads.

The design ensures that the internal database does not become a dependency for the external system's correctness; rather, it acts as an internal source of truth for observability and accountability.

4.2.2.4 External System Boundary (eMarisma)

eMarisma Web Application (HTTP interface)

The external eMarisma web application is treated as a third-party system accessed through HTTP calls. The Risk Client issues these calls as part of the workflows orchestrated in the eMarisma HTTP Service. This boundary is explicitly drawn to highlight that failures, latency, authentication, and API contracts are external to the system's control and must be handled defensively.

External MySQL Database (direct query access)

The architecture also integrates with the eMarisma MySQL database via direct queries for lookup/translation functions (as represented by "Direct Query" arrows). This supports efficient resolution of entity identifiers and complements HTTP-based operations.

4.2.2.5 End-to-End Request Execution Flow

A typical request follows this sequence:

1. **Ingress**: The FastAPI application receives a request.
2. **Interception**: Custom middleware logs and enriches context (e.g., correlation ID).
3. **Dispatch**: The route handler validates input via Pydantic and invokes the relevant service.
4. **Orchestration**: The eMarisma HTTP service executes a structured workflow.
5. **External interactions**:
    a. HTTP calls are issued through the Risk Client to the eMarisma web application.
    b. When required, identifier resolution queries are executed through the eMarisma DB service against the MySQL database.
6. **Persistence (optional/parallel)**: The internal DB service stores request metadata and outcomes in SQLite.
7. **Response**: The route handler returns a structured response; middleware finalizes logging and metrics.

4.2.2.6 General Functionality

When an incident request arrives (/new_incident):

1. Reception and Log**:** The middleware records the request entry.
2. Validation and Persistence**:** The data structure is validated and a copy of the request is saved in the internal database.
3. ID Resolution**:** Internal IDs (Project, Subproject, Asset/Device, Threat) are queried based on the names provided. If any do not exist, the request is rejected.
4. Orchestration (Flow)**:** *run_all_ffow* is started in services, which executes sequentially:
    - Authentication in the external system.
    - Loading of context (projects/subprojects).
    - Recording of severity and threat.

- Linking of assets and controls involved.
- Closing of the event and recalculation of risk metrics.

4.2.3 API endpoints

The endpoints available in the service are detailed in Table 4.2.

Method	**Endpoint**	**Parameters (Body/Path)**	**Description**
POST	/new_incident	**Body (JSON):** IncidentRequest • threat_id: Threat ID • user_id: User ID • device_id: Active name • detected_at: ISO date • threat_type: Threat type • severity: Level (low/high) • actions taken: Type action • status: status • project_name: Project name • subproject_name: Subproject Name • cause: cause of the incident	Main entry point. Creates a new incident. Performs validation of existence (project, active, etc.) against the internal database and, if correct, triggers the automatic flow of registration and remediation flow in the external system.
GET	/retrieve_incident/{incident_id}	**Path:** incident_id (string)	Retrieves or processes a specific incident based on its ID, executing the business flow associated associated with it.

Table 4.2: API endpoints

4.2.4 Middleware analysis (CustomMiddleware)

The project implements custom middleware in middleware/custom_middleware.py that intercepts all incoming HTTP requests.

Operation:

1. Interception (Pre-processing): Before the request reaches the route, it captures the start time and logs the HTTP method and requested URL (POST /new_incident).
2. Execution: Passes control to the next handler (the actual route).
3. Interception (Post-processing): Once the route has responded, it captures the end time, calculates the total duration in milliseconds, and records the response status and latency (POST /new_incident 200 (150.ms)).

Usefulness:

- Observability: Allows you to see in real time which requests the server is receiving.
- Performance: Helps identify bottlenecks by showing the execution time of each endpoint.
- Debugging: Facilitates tracing the execution flow in the logs.

5 Sustainable Security

In recent years, Cyber–Physical Systems (CPS) have attracted increasing attention from industry, governments, and society due to their significant impact on critical infrastructures and industrial environments [5.1, 5.2]. CPS are highly complex systems composed of heterogeneous hardware and software components, sensors, controllers, communication protocols, and networking technologies [5.3, 5.4]. While this complexity enables advanced automation and innovative services, it also introduces substantial challenges related to safety and security [5.5]. In particular, the definition and validation of security requirements during the early stages of CPS design remains a critical and non-trivial task [5.6, 5.7, 5.8], as inadequate or inconsistent security requirements may lead to misconfigurations and catastrophic consequences [5.9, 5.10].

To address this complexity, several works have proposed formal models and semantic approaches to support security requirements engineering in CPS and IoT environments [5.11, 5.12]. Among these, Onto-CARMEN represents a significant advancement by proposing an ontology-driven approach for modelling, verifying, and diagnosing security requirements in CPS. Onto-CARMEN formalizes security requirements in terms of assets, security features, properties, constraints, and security levels, following the security recommendations provided by ENISA [5.3] and OWASP [5.14]. By leveraging ontological reasoning and SPARQL rules, Onto-CARMEN enables the automatic verification of security requirements and provides diagnostic feedback to correct invalid configurations at design time, thereby improving the consistency and robustness of CPS security specifications [5.13].

Ontologies play a fundamental role in this context, as they provide a shared and formal representation of domain knowledge, allowing stakeholders to reason about security requirements in a systematic manner [5.15, 5.16]. In Onto-CARMEN, security requirements are validated not only against syntactic constraints but also against semantic relationships among assets, security features, and security levels, enabling early detection of inconsistencies that would otherwise propagate to later development stages [5.13].

Despite these advantages, sustainability aspects are only marginally addressed in current security ontologies. Onto-CARMEN introduces the concept of a SustainabilityLabel as an attribute that can be associated with different classes (e.g., devices, infrastructures, platforms), but its use is limited and does not directly influence the reasoning process [5.13]. This limitation is increasingly relevant, as security mechanisms and sustainability concerns are deeply interconnected. Cryptographic algorithms, authentication mechanisms, security devices, and communication protocols have a direct impact on energy consumption, computational overhead, maintenance costs, and hardware lifecycle, all of which affect the environmental footprint and long-term viability of CPS deployments.

In this context, ECOSEC is proposed as a sustainability-oriented extension that complements ontology-based security frameworks such as Onto-CARMEN. ECOSEC aims to integrate sustainability as a first-class concern in security analysis by introducing a structured classification of security mechanisms and devices based on sustainability criteria. These criteria include technological adoption, computational efficiency, operational costs, maintenance requirements, and, in the case of physical devices, full lifecycle considerations such as manufacturing, transportation, usage, replacement, and end-of-life management. Sustainability is expressed through standardized labels ranging from A (most sustainable) to G (least sustainable), enabling systematic comparison and reasoning across alternative security configurations.

By integrating ECOSEC into an ontological framework for CPS security requirements, it becomes possible to reason not only about traditional security properties—such as confidentiality, integrity, or authentication—but also about their environmental and lifecycle impact. This integration supports a holistic notion of sustainable security, in which design-time decisions balance security strength with long-term sustainability, contributing to more efficient, resilient, and environmentally responsible CPS architectures.

5.1 EcoSec Ontology

EcoSec is a comprehensive sustainability ontology designed to systematically assess and label the sustainability impact of security mechanisms and devices across their lifecycle or operational scope (see **Error! Reference source not found.**re 5.2 and Figure 5.3). By assigning sustainability labels ranging from A (most sustainable) to G (least sustainable), EcoSec provides a nuanced framework for evaluating the environmental and operational sustainability of various elements (see **Error! Reference source not found.**re 5.1).

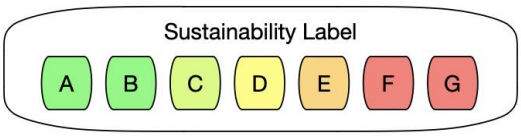

Figure 5.1: Sustainability labels

For security mechanisms, EcoSec assesses key aspects such as adoption, maintenance, and efficiency to understand their sustainability footprint.

In the context of devices, the ontology adopts a lifecycle approach, meticulously evaluating the sustainability performance at each stage (from raw material sourcing, manufacturing, and packaging, to transport, usage, spare parts availability, and end-of-life management).

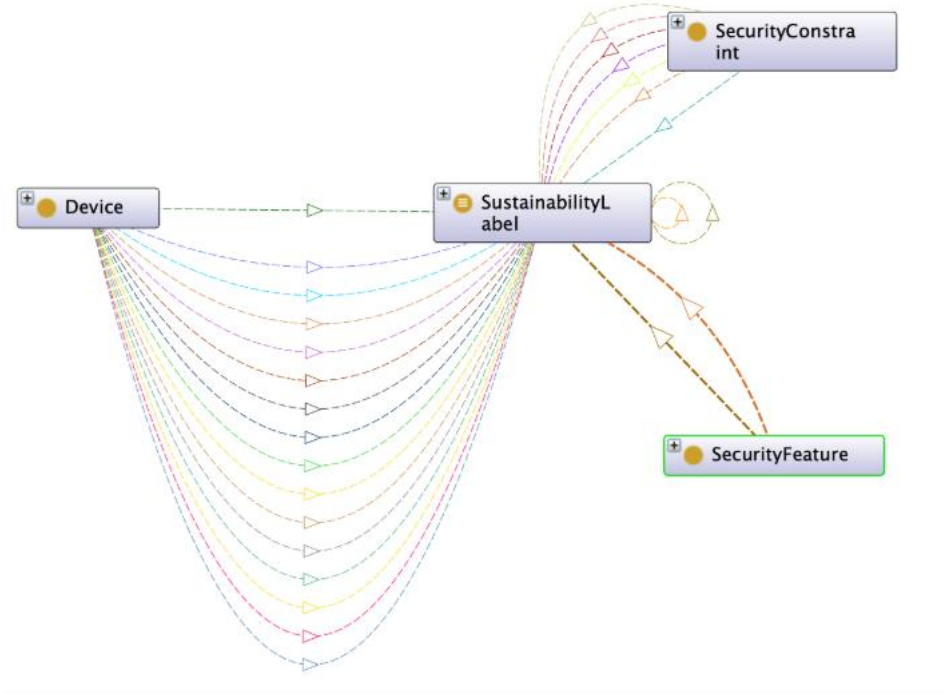

Figure 5.2: Ontology classes overview

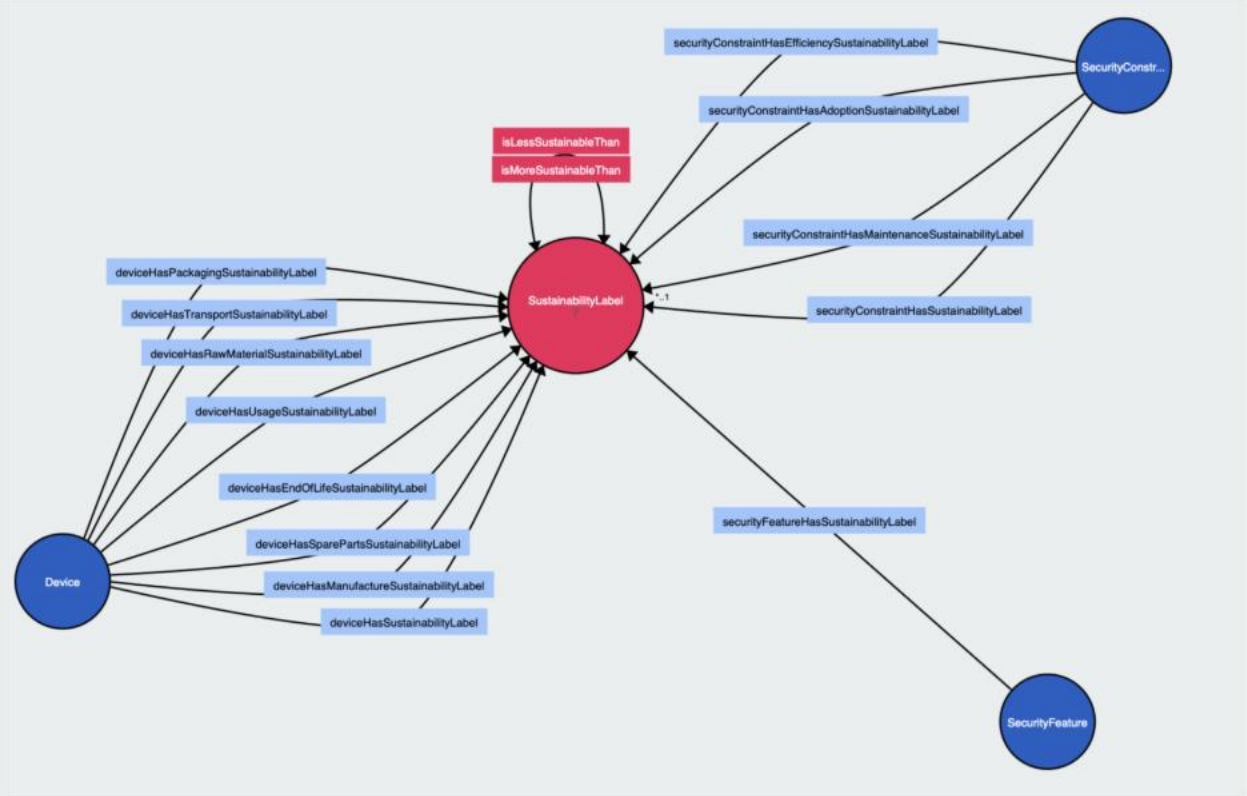

Figure 5.3: Ontology elements overview

5.1.1 Sustainability Labels

Class SustainabilityLabel

Represents the sustainability performance of various entities, including devices, security mechanisms, and other products, across their lifecycle or operational practices. Labels range from A (most sustainable) to G (least sustainable), providing a clear and standardized measure of environmental impact, resource efficiency, and overall sustainability.

Object property isLessSustainableThan

Indicates that one sustainability label denotes a lower level of sustainability compared to another. This hierarchical relationship helps in understanding and comparing the sustainability performance across different labels.

Object property isMoreSustainableThan

Indicates that one sustainability label represents a higher level of sustainability compared to another. It establishes a hierarchy among labels, facilitating comparison and decision-making based on sustainability criteria.

Individual ASustainabilityLabel

Represents the highest standard of sustainability. Products or practices with this label are considered to have the lowest environmental impact and highest efficiency and sustainability in their category.

Individual BSustainabilityLabel

Denotes a high level of sustainability, second only to A. Products or practices labeled B are highly sustainable with minor areas for improvement compared to A.

Individual CSustainabilityLabel

Reflects a moderate level of sustainability. C labeled entities offer a balanced sustainability performance but with significant room for improvement to reach higher standards.

Individual DSustainabilityLabel

Indicates a sustainability performance that is below average, with D being more sustainable than E, F, and G but less so than C, B, and A. As the labels progress to E, F, and G, they represent decreasing levels of sustainability, indicating increasing environmental impacts and lower efficiency.

Individual ESustainabilityLabel

Signifies a lower level of sustainability, indicating a need for significant improvements. Products or practices with an E label have a notable environmental impact and efficiency issues. This label suggests that while some sustainability measures may be in place, there are considerable areas for improvement to move towards a more sustainable outcome.

Individual FSustainabilityLabel

Represents a sustainability level that is second to last, highlighting substantial environmental impacts and a lack of efficiency. F labeled entities require major changes and enhancements in their sustainability practices and operations to reduce their negative environmental footprint.

Individual GSustainabilityLabel

Denotes the lowest level of sustainability, indicating the highest environmental impact and the least efficient use of resources. Entities with a G label are considered the least sustainable, underscoring an urgent need for comprehensive measures to significantly improve their sustainability performance.

5.1.2 Sustainability Requirements

Sustainability Requirements allow expressing sustainability-related expectations linked to existing security requirements.

A Security Requirement can be associated with one or more Security Features, which represent the security functionalities needed to satisfy that requirement.

Each Security Feature (inherited from OntoCarmen) includes the following properties:

- Security constraints.
- Security level.
- Sustainability label, added in EcoSec.

This sustainability label indicates the expected environmental and operational impact of the security functionality, independently of its specific implementation. In this way, security requirements can specify not only what security features are needed, but also the level of sustainability at which they should be implemented (See Figure 9.2).

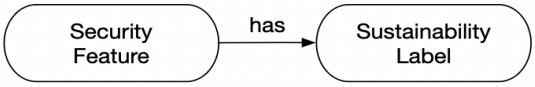

Figure 5.4: Security Feature

Object property securityFeatureHasSustainabilityLabel

Connects a security feature to a minimum sustainability label requirement, indicating the lowest acceptable level of sustainability across its associated security mechanisms. This label, which ranges from A to G, serves as a benchmark to ensure that all security constraints tied to the feature meet or exceed a specified standard of environmental, economic, and operational sustainability. It ensures that the security feature aligns with broader sustainability goals by aggregating the sustainability performance of its multiple constraints.

5.1.3 Sustainability of Security Mechanisms

This section associates sustainability labels with security mechanisms. Each Security Constraint has four sustainability labels:

- General label: overall sustainability of the mechanism.
- Adoption: reflects how easily the mechanism can be adopted and integrated.
- Efficiency: measures operational and computational efficiency.
- Maintenance: assesses long-term maintenance effort and impact.

The three partial labels (Adoption, Efficiency, Maintenance) are used to calculate the general sustainability label using simple aggregation rules.

Additionally, EcoSec defines a set of individuals representing typical security mechanisms, each pre-labeled with a sustainability score and accompanied by an explanation of the reasoning see Figure 5.5 and Figure 5.6).

This allows these mechanisms to be reused directly, providing a library of pre-evaluated, sustainability-aware security components for design and analysis.

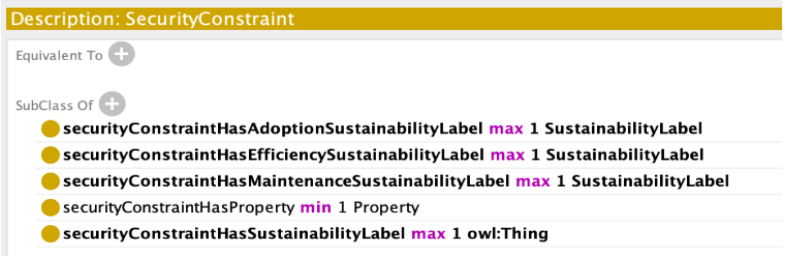

Figure 5.5: Security Constraint description

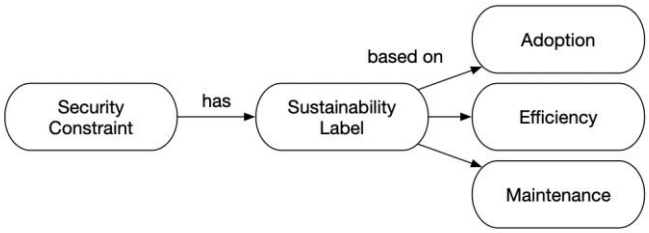

Figure 5.6: Security Constraint

Object property securityConstraintHasSustainabilityLabel

Associates a security mechanism with its overall sustainability label, which represents a comprehensive evaluation of the mechanism's sustainability across three key dimensions: adoption, maintenance, and efficiency. This label, ranging from A to G, is calculated based on the combined assessments of the mechanism's adoption label, maintenance label, and efficiency label. The overall sustainability label provides a holistic view of the environmental, economic, and operational sustainability of the security mechanism, reflecting its long-term viability and impact.

Object property securityConstraintHasAdoptionSustainabilityLabel

Associates a security mechanism with its adoption label, reflecting the extent to which the technology has been accepted and adopted within the industry. This label, ranging from A to G, indicates the mechanism's prevalence and popularity among users and developers, providing insight into its widespread or limited adoption. The adoption label is a sustainability label, categorizing the mechanism's environmental and operational sustainability in terms of its adoption.

Object property securityConstraintHasEfficiencySustainabilityLabel

Links a security mechanism to its efficiency label, indicating the resource effectiveness and performance of the technology in operational environments. This label, which ranges from A to G, evaluates the mechanism's sustainability by measuring its computational efficiency, energy consumption, and the optimization of resources required for its functionality. The efficiency label is a sustainability label that underscores the environmental and economic impact of deploying and operating the security mechanism.

Object property securityConstraintHasMaintenanceSustainabilityLabel

Connects a security mechanism to its maintenance label, which quantifies the effort and resources required to keep the technology secure and up-to-date over time. This label, ranging from A to G, assesses the mechanism's sustainability in terms of ongoing support, frequency of updates needed, and overall lifecycle management. The maintenance label is a sustainability label that highlights the operational efficiency and long-term viability of the security mechanism.

Table 5.1 defines basic rules for aggregating sustainability labels from sub-characteristics. For example, if all sub-characteristics of a mechanism or device are rated A or B, the overall element is assigned as an A.

These rules are applied both to security mechanisms and devices, providing a simple and consistent method to calculate general sustainability labels from partial evaluations (see example in Figure 5.6).

Table 5.1 Sustainability rating rules

Sustainability Rating	**Rules**
A	If all categories have at least one A or B label, then the device receives a sustainability label A.
B	If all categories have at least one B or C label, then the device receives a sustainability label B.
C	If all categories have at least one C or D label, then the device receives a sustainability label C.
D	If all categories have at least one D or E label, then the device receives a sustainability label D.
E	If all categories have at least one E or F label, then the device receives a sustainability label E.
F	If all categories have at least one F or G label, then the device receives a sustainability label F.
G	If all categories have at least one G label, then the device receives a sustainability label G.

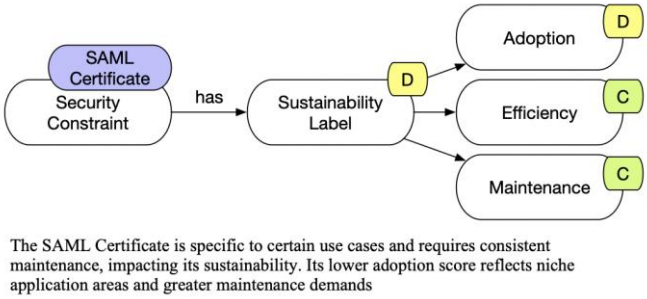

Figure 5.6: Example with SAML Certificate

Individual OpenPGPCertificateIndividual

- Sustainability Label: C
- Efficiency Label: C
- Maintenance Label: C
- Adoption Label: C
- Explanation: “This security mechanism is universal in its application but entails some complexity, which moderately impacts its overall sustainability rating. Its universality is tempered by complexities that moderate its efficiency, maintenance, and adoption scores, reflecting a balanced sustainability performance."

Individual OpenSSLCertificateIndividual

- Sustainability Label: B
- Efficiency Label: B
- Maintenance Label: B
- Adoption Label: B
- Explanation: "Efficiency and compatibility drive the OpenSSL Certificate to a 'B' sustainability rating across all aspects. It demonstrates high efficiency and broad compatibility, making it a sustainable choice for security implementations."

Individual SAMLCertificateIndividual

- Sustainability Label: D
- Efficiency Label: C
- Maintenance Label: C
- Adoption Label: D
- Explanation: "The SAML Certificate is specific to certain use cases and requires consistent maintenance, impacting its sustainability. Its lower adoption score reflects niche application areas and greater maintenance demands."

Individual X509CertificateIndividual

- Sustainability Label: A
- Efficiency Label: A
- Maintenance Label: A
- Adoption Label: A
- Explanation: "Broad adoption and interoperability grant the X.509 Certificate an 'A' rating in all sustainability categories. Its widespread acceptance and efficiency make it an exemplary model of sustainable security practice."

Individual HTTPSChannelIndividual

- Sustainability Label: A
- Efficiency Label: A
- Maintenance Label: A
- Adoption Label: A
- Explanation: "The universality and efficiency in the implementation of the HTTPS Channel secure it an 'A' rating across the board, showcasing optimal sustainability in secure communications."

Individual SSLTLSChannelIndividual

- Sustainability Label: B
- Efficiency Label: B
- Maintenance Label: B
- Adoption Label: B
- Explanation: "As the foundational technology for HTTPS, the SSL/TLS Channel exhibits high efficiency and broad version support, justifying its 'B' sustainability ratings."

Individual TunnelingChannelIndividual

- Sustainability Label: C
- Efficiency Label: C
- Maintenance Label: C
- Adoption Label: C
- Explanation: "The Tunneling Channel's sustainability is moderated by its type-dependent resource requirements, resulting in 'C' ratings across efficiency, maintenance, and adoption."

Individual AES128GCMCipherIndividual

- Sustainability Label: A
- Efficiency Label: A
- Maintenance Label: A
- Adoption Label: A
- Explanation: "Marked by its efficiency and broad adoption, the AES-128-GCM Cipher achieves 'A' ratings, highlighting its role as a sustainable cipher choice."

Individual CamelliaCipherIndividual

- Sustainability Label: C
- Efficiency Label: B
- Maintenance Label: C
- Adoption Label: C
- Explanation: "Though efficient, the Camellia Cipher's lower adoption and moderate maintenance requirements reflect its 'C' sustainability label, with room for broader acceptance."

Individual ChaCha20CipherIndividual

- Sustainability Label: B
- Efficiency Label: B

- Maintenance Label: B
- Adoption Label: B
- Explanation: "The ChaCha20 Cipher's good efficiency, particularly on mobile devices, secures it 'B' ratings, standing out for its performance and sustainable security approach."

Individual MultiFactorPasswordIndividual

- Sustainability Label: A
- Efficiency Label: A
- Maintenance Label: A
- Adoption Label: A
- Explanation: "Greater security and a reduced frequency of changes earn the Multi-Factor Password an 'A' rating in all categories, underscoring its exceptional sustainability in safeguarding access."

Individual StrongPasswordIndividual

- Sustainability Label: B
- Efficiency Label: B
- Maintenance Label: B
- Adoption Label: B
- Explanation: "The Strong Password balances security and usability well, achieving 'B' sustainability ratings. It represents a solid choice for secure yet user-friendly access controls."

Individual WeakPasswordIndividual

- Sustainability Label: G
- Efficiency Label: D
- Maintenance Label: D
- Adoption Label: D
- Explanation: "The frequent need for changes due to security vulnerabilities places the Weak Password at a 'G' sustainability rating, highlighting significant sustainability concerns."

Individual SHA2SecureHashingIndividual

- Sustainability Label: B
- Efficiency Label: B
- Maintenance Label: B
- Adoption Label: B

- Explanation: "Widely adopted yet transitioning towards SHA-3, SHA-2 Secure Hashing maintains 'B' ratings, reflecting its current relevance and sustainability in secure hashing."

Individual SHA3SecureHashingIndividual

- Sustainability Label: A
- Efficiency Label: A
- Maintenance Label: A
- Adoption Label: A
- Explanation: "The new, efficient design of SHA-3 Secure Hashing designed for longevity secures it an 'A' rating across all categories, marking it as a highly sustainable choice for secure hashing."

Individual PSKSignatureIndividual

- Sustainability Label: C
- Efficiency Label: C
- Maintenance Label: C
- Adoption Label: C
- Explanation: "The PSK Signature's niche application and specific use result in 'C' sustainability ratings, indicating moderate sustainability performance."

Individual SRPSignatureIndividual

- Sustainability Label: B
- Efficiency Label: B
- Maintenance Label: B
- Adoption Label: B
- Explanation: "Balancing efficiency with security, the SRP Signature achieves 'B' ratings, demonstrating its sustainable application in secure signature processes."

5.1.4 Sustainability of Devices

This section evaluates the sustainability of devices by extending the existing Device class with sustainability-related properties.

Each device is associated with a general sustainability label, which is calculated from partial labels corresponding to its lifecycle stages. These partial labels reflect the environmental and operational impact of each stage, enabling a fine-grained assessment.

It follows the life cycle of the product and evaluates the sustainability of each stage: raw material, manufacture, packaging, transport, usage, spare parts and end of life (see Figures 5.7 and 5.8).

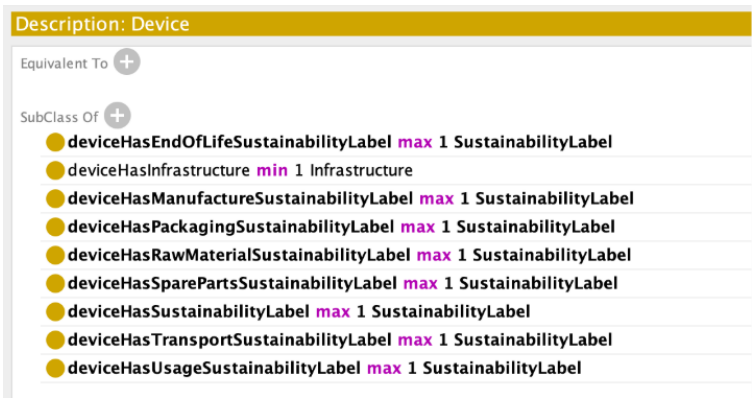

Figure 5.7: Sustainability of Device descripcion

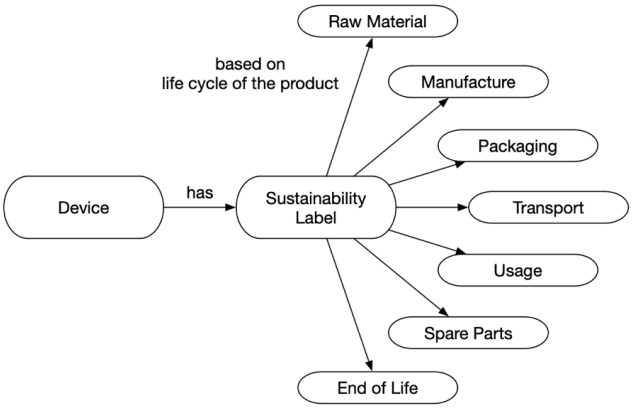

Figure 5.8: Sustainability of Device

Object property deviceHasSustainabilityLabel

Associates a device with its overall sustainability label, calculated based on the sustainability labels of each stage in the device's lifecycle, including raw material acquisition, manufacture, packaging, transport, usage, spare parts, and end of life. This comprehensive label, ranging from A to G, encapsulates the device's total environmental impact, reflecting a holistic assessment of its sustainability performance across all lifecycle stages.

Object property deviceHasRawMaterialSustainabilityLabel

Associates a device with a sustainability label for the raw materials stage, evaluating material origin, extraction impacts, efficiency, recycled content, and compliance with hazardous substance regulations. This comprehensive assessment covers types of materials (recycled, renewable, conflict-free), environmental and social impacts of extraction, material usage efficiency, percentage of recycled materials, and adherence to standards like RoHS and REACH.

Object property deviceHasManufactureSustainabilityLabel

Links a device to a sustainability label for its manufacturing stage, considering CO2 emissions, water consumption, renewable energy usage, and waste treatment. Factors assessed include CO2 equivalent emissions, water usage efficiency, the percentage of renewable energy in manufacturing processes, and the effectiveness of production waste recycling and disposal.

Object property deviceHasPackagingSustainabilityLabel

Connects a device to a sustainability label for packaging, focusing on material types, efficiency, and sustainability information. Evaluates the use of recyclable, biodegradable, and reusable materials, optimization of packaging design to reduce waste, and the presence of labeling with recycling and disposal guidelines.

Object property deviceHasTransportSustainabilityLabel

Associates a device with a sustainability label for transport, assessing transport mode efficiency, logistics optimization, and transport packaging. Considers the efficiency of air, sea, and land transportation, measures to reduce transportation distances and consolidate shipments, and efforts to minimize volume and weight for transport efficiency.

Object property deviceHasUsageSustainabilityLabel

Links a device to a sustainability label for usage, covering energy efficiency, durability, repairability, and software updates. Reviews energy consumption, physical resistance, estimated lifespan, ease of repair, spare parts availability, and the impact of long-term software support on functionality and efficiency.

Object property deviceHasSparePartsSustainabilityLabel

Connects a device to a sustainability label for spare parts, evaluating availability, compatibility, and sustainability. Addresses ease of access to spare parts, their compatibility across different sources or models, and the environmental impact of materials and manufacturing processes for spare parts.

Object property deviceHasEndOfLifeSustainabilityLabel

Associates a device with a sustainability label for end-of-life management, focusing on recyclability, return programs, and reusability. Assesses the ease of device disassembly and material sorting, initiatives for device return to manufacturers or recycling points, and the potential for component or entire device reuse.

If we want to go to a higher level of detail in the sustainability of devices, we can use the Tables 5.2 - 5.9:

Table 5.2 Categories of Devices

Category	Subcategory	Description	Data Type
Raw Material	Material Origin	Types of materials used	Enumerated values: Recycled, Renewable, Conflict
	Material Extraction	Environmental, social, and economic impact of material extraction.	Score: 0-100
	Material Efficiency	Efficiency in material usage, design optimization for material use.	Score: 0-100
	Recycled Material Content	Percentage of recycled materials used in the device.	Percentage: 0-100%
	Use of Hazardous and Restricted Substances	Presence of hazardous substances, adherence to regulations like RoHS, REACH.	Enumerated values: Compliant, Non-Compliant
Manufacturing	CO2 Emissions	CO2 emissions during production.	Score: 0-100 based on CO2 equivalent
	Water Consumption	Efficiency and sources of water used in production.	Score: 0-100 based on liters or cubic meters
	Renewable Energy Usage	Percentage of renewable energy used in manufacturing processes.	Percentage: 0-100%
	Waste Treatment	Recycling and disposal of production waste.	Score: 0-100 based on efficiency
Packaging	Packaging Materials	Types of packaging materials used	Enumerated values: Recyclable, Biodegradable, Reusable
	Packaging Efficiency	Optimization of material and size to reduce waste.	Score: 0-100
	Sustainability Information	Labeling with recycling and disposal information.	Score: 0-100 based on the presence of information
Transport	Transport Mode	Efficiency of transportation modes used.	Enumerated values: Air, Sea, Land
	Logistics Optimization	Measures to reduce transportation distance, shipment consolidation.	Score: 0-100
	Transport Packaging	Reduction in volume and weight for transport efficiency.	Score: 0-100

Usage	Energy Efficiency	Energy consumption during normal use.	Score: 0-100 based on watts or kilowatt-hours
	Durability	Includes Physical Resistance and Estimated Life Span.	Score: 0-100 for Physical Resistance, Range in years for lifespan
	Repairability	Ease of repair, availability of spare parts.	Score: 0-100
	Software Updates	Long-term support impact on efficiency and functionality.	Score: 0-100
Spare Parts	Availability	Ease of access to spare parts.	Score: 0-100
	Compatibility	Ability to use spare parts from various sources or models.	Score: 0-100
	Sustainability of Spare Parts	Materials and manufacturing processes for spare parts.	Score: 0-100
End of Life	Recyclability	Ease of disassembly, material sorting.	Score: 0-100
	Return Programs	Initiatives for returning the device to the manufacturer or recycling points.	Score: 0-100
	Reusability	Potential for component or entire device reuse.	Score: 0-100

Table 5.3 Raw Material

Sustainability Rating	**Material Origin**	**Material Extraction Score**	**Material Efficiency Score**	**Recycled Material Content**	**Use of Hazardous and Restricted Substances**
A	Renewable	81-100	81-100	>80%	Compliant
B	Recycled	61-80	61-80	60-80%	Compliant
C	Recycled	41-60	41-60	40-59%	Compliant
D	Recycled/Renewable	21-40	21-40	20-39%	Compliant
E	Conflict	11-20	11-20	10-19%	Non-Compliant
F	Conflict	1-10	1-10	1-9%	Non-Compliant
G	Conflict	0	0	0%	Non-Compliant

Table 5.4 Manufactoring

Sustainability Rating	**CO2 Emissions Score**	**Water Consumption Efficiency**	**Renewable Energy Usage**	**Waste Treatment Efficiency**
A	Very Low (81-100)	Very Efficient (81-100)	>80% Renewable	Very Efficient (81-100)
B	Low (61-80)	Efficient (61-80)	60-80% Renewable	Efficient (61-80)
C	Moderate (41-60)	Moderately Efficient (41-60)	40-59% Renewable	Moderately Efficient (41-60)
D	Medium (21-40)	Average Efficiency (21-40)	20-39% Renewable	Average Efficiency (21-40)
E	High (11-20)	Low Efficiency (11-20)	10-19% Renewable	Low Efficiency (11-20)
F	Very High (1-10)	Very Low Efficiency (1-10)	1-9% Renewable	Very Low Efficiency (1-10)
G	Extremely High (0)	No Efficiency (0)	0% Renewable	No Efficiency (0)

Table 5.5 Packaging

Sustainability Rating	**Packaging Materials**	**Packaging Efficiency**	**Sustainability Information**
A	Recyclable, Biodegradable, Reusable	Very Efficient (81-100)	Comprehensive (81-100)
B	Recyclable, Biodegradable, Reusable	Efficient (61-80)	Good (61-80)
C	Recyclable, Biodegradable, Reusable	Moderately Efficient (41-60)	Moderate (41-60)
D	Recyclable, Biodegradable, Reusable	Average Efficiency (21-40)	Minimal (21-40)
E	Recyclable, Biodegradable, Reusable	Low Efficiency (11-20)	Low (11-20)
F	Recyclable, Biodegradable, Reusable	Very Low Efficiency (1-10)	Very Low (1-10)
G	Non-Recyclable (0)	No Efficiency (0)	None (0)

Table 5.6 Transport

Sustainability Rating	Transport Mode	Logistics Optimization	Transport Packaging
A	Land, Sea, Air	Highly Optimized (81-100)	Very Efficient (81-100)
B	Land, Sea, Air	Optimized (61-80)	Efficient (61-80)
C	Land, Sea, Air	Moderately Optimized (41-60)	Moderately Efficient (41-60)
D	Land, Sea, Air	Average Optimization (21-40)	Average Efficiency (21-40)
E	Land, Sea, Air	Low Optimization (11-20)	Low Efficiency (11-20)
F	Land, Sea, Air	Very Low Optimization (1-10)	Very Low Efficiency (1-10)
G	Land, Sea, Air	No Optimization (0)	No Efficiency (0)

Table 5.7 Usage

A	High Efficiency (81-100)	Highly Durable (81-100)	Highly Repairable (81-100)	Comprehensive (81-100)
B	Efficient (61-80)	Durable (61-80)	Repairable (61-80)	Regular (61-80)
C	Moderate Efficiency (41-60)	Moderately Durable (41-60)	Moderately Repairable (41-60)	Occasional (41-60)
D	Average Efficiency (21-40)	Average Durability (21-40)	Average Repairability (21-40)	Infrequent (21-40)
E	Low Efficiency (11-20)	Low Durability (11-20)	Low Repairability (11-20)	Rare (11-20)
F	Very Low Efficiency (1-10)	Very Low Durability (1-10)	Very Low Repairability (1-10)	Minimal (1-10)
G	Inefficient (0)	Non-Durable (0)	Non-Repairable (0)	None (0)

Table 5.8 Spare Parts

A	Highly Available (81-100)	Highly Compatible (81-100)	Highly Sustainable (81-100)
B	Available (61-80)	Compatible (61-80)	Sustainable (61-80)
C	Moderately Available (41-60)	Moderately Compatible (41-60)	Moderately Sustainable (41-60)
D	Limited Availability (21-40)	Limited Compatibility (21-40)	Limited Sustainability (21-40)

E	Rarely Available (11-20)	Rarely Compatible (11-20)	Unsustainable (11-20)
F	Very Rarely Available (1-10)	Very Rarely Compatible (1-10)	Very Unsustainable (1-10)
G	Not Available (0)	Not Compatible (0)	Not Sustainable (0)

Table 5.9 End of Life

Sustainability Rating	**Recyclability**	**Return Programs**	**Reusability**
A	Highly Recyclable (81-100)	Comprehensive (81-100)	Highly Reusable (81-100)
B	Recyclable (61-80)	Good (61-80)	Reusable (61-80)
C	Moderately Recyclable (41-60)	Moderate (41-60)	Moderately Reusable (41-60)
D	Limited Recyclability (21-40)	Minimal (21-40)	Limited Reusability (21-40)
E	Low Recyclability (11-20)	Rare (11-20)	Low Reusability (11-20)
F	Very Low Recyclability (1-10)	Very Rare (1-10)	Very Low Reusability (1-10)
G	Non-Recyclable (0)	None (0)	Not Reusable (0)

6 Conclusions

This deliverable consolidates the main WP4 outcomes by delivering a coherent set of **methods and technical enablers** that support dynamic, evidence-driven security decision-making in distributed and decentralised environments. In particular, it formalises **MARISMA-DS** as a DS-tailored risk management pattern grounded in recognised international standards and guidance, and it demonstrates how explicit dependency modelling (assets, threats, controls and security dimensions) can be used to represent interrelationships and support systematic reuse across scenarios.

In parallel, the work operationalises continuous monitoring and prioritisation through **VulnQ**, integrating standard security taxonomies and a predictive module that estimates the **evolution of asset risk over time** using historical KRIs and regression-based modelling, supported by end-to-end validation of the integrated workflow and dashboard visualisation. To complement this, a **Risk Management Middleware** layer is presented to automate incident and risk workflows over an external platform without a documented API, reconstructing the full business flow via HTTP inspection and implementing a REST orchestration service to ensure traceability and repeatability of operations.

Overall, the deliverable meets WP4 objectives by (i) providing a structured and adaptable basis for **dynamic risk analysis and treatment** in DS contexts, (ii) enabling **evidence-based prioritisation** through predictive analytics and automation, and (iii) incorporating **sustainable security** through ontology-driven constructs and sustainability labelling to support more resource-aware security choices.

These results will directly serve as a baseline for **WP5 integration activities and the elaboration of test cases**, where the proposed patterns, workflows, predictive services and middleware orchestration can be instantiated and verified in integrated scenarios, validating both functional behaviour and measurable performance across representative DS deployments.

7 References

[2.1] RESISTO, Decision support system for protecting communication infrastructures against cyber-physical threats based on Software Defined Security, 2023.

[2.2] Agence nationale de la sécurité des systèmes d'information (ANSSI), EBIOS Risk Manager – Expression des Besoins et Identification des Objectifs de Sécurité, 2020.

[2.3] C.J. Alberts, A.J. Dorofee, OCTAVE Allegro: Improving the Information Security Risk Assessment Process, Pittsburgh, PA, USA, 2007.

[2.4] S. Bhatta, J. Dang, Use of distributed systems for structural health monitoring of civil engineering structures: a state-of-the-art review, Urban Lifeline 2 (2024) 17, http://dx.doi.org/10.1007/s44285-024-00031-2.

[2.5] British Standards Institution, BSI 7799-3:2017 — Information security management systems. Guidelines for information security risk management, 2017.

[2.6] I. Butun, P. Österberg, H. Song, Security of the distributed systems: Vulnerabilities, attacks, and countermeasures, IEEE Communications Surveys & Tutorials 22 (2020) 616–644, http://dx.doi.org/10.1109/COMST.2019.2953364.

[2.7] Z. Cao, Z. Zhao, W. Shang, S. Ai, S. Shen, Using the TON-DS dataset to develop a new intrusion detection system for industrial distributed systems devices, Multimedia Tools and Applications 84 (2025) 16425–16453, http://dx.doi.org/10.1007/s11042-024-19695-7.

[2.8] P.G. Chiara, Holistic risk analysis for distributed systems, in: The Distributed Systems and EU Law: Cybersecurity, Privacy and Data Protection Challenges, Springer Nature Switzerland, 2024, pp. 203–241, http://dx.doi.org/10.1007/978-3-031-67663-5_6.

[2.9] Club de la Sécurité de l'Information Français (CLUSIF), MEHARI: MEthod for Harmonized Analysis of RIsk, version 2010, 2010.

[2.10] J. Deogirikar, A. Vidhate, Security attacks in distributed systems: A survey, in: 2017 International Conference on I-SMAC, IEEE, 2017, pp. 32–37, http://dx.doi.org/10.1109/I-SMAC.2017.8058363.

[2.11] M.F. Elrawy, A.I. Awad, H.F.A. Hamed, Intrusion detection systems for distributed systems-based smart environments: a survey, Journal of Cloud Computing 7 (2018) 21, http://dx.doi.org/10.1186/s13677-018-0123-6.

[2.12] European Telecommunications Standards Institute (ETSI), ETSI EN 303 645 V2.1.1: Cyber Security for Consumer Distributed Systems – Baseline Requirements, 2020.

[2.13] European Union Agency for Cybersecurity (ENISA), Risk Management: Implementation principles and inventories for risk management/risk assessment methods and tools, 2006.

[2.14] European Union Agency for Cybersecurity (ENISA), Cybersecurity for SMEs: Challenges and Recommendations, 2021.

[2.15] European Union Agency for Cybersecurity (ENISA), ENISA Threat Landscape 2023, 2023.

[2.16] European Union Agency for Network and Information Security (ENISA), Threat Landscape and Good Practice Guide for Smart Home and Converged Media, 2015.

[2.17] European Union Agency for Network and Information Security (ENISA), Baseline Security Recommendations for Distributed Systems in the Context of Critical Information Infrastructures, 2017.

[2.18] G. Gopichand, T. Sarath, A. Dumka, et al., Use of distributed systems sensor devices for efficient management of healthcare systems: a review, Discover Distributed Systems 4 (2024) 8, http://dx.doi.org/10.1007/s43926-024-00062-9.

[2.19] Bundesamt für Sicherheit in der Informationstechnik (BSI), IT-Grundschutz, 2023.

[2.20] International Organization for Standardization, ISO 31000:2018 – Risk Management – Guidelines, 2018.

[2.21] ISO/IEC, ISO/IEC 21827:2008 — Systems Security Engineering — Capability Maturity Model (SSE-CMM), 2008.

[2.22] ISO/IEC, ISO/IEC 27005:2022 — Information security, cybersecurity and privacy protection — Guidance on managing information security risks, 2022.

[2.23] ISO/IEC, ISO/IEC 27400:2022 — Cybersecurity — Guidelines for Distributed Systems security, 2022.

[2.24] ISO/IEC, ISO/IEC 27001:2023 — Information security, cybersecurity and privacy protection — Information security management systems — Requirements, 2023.

[2.25] Distributed Systems Security Foundation (IoTSF), Distributed Systems Security Assurance Framework, Release 3.0, 2023.

[2.26] ISACA, COBIT 2019: Governance and Management Objectives, Schaumburg, IL, USA, 2019.

[2.27] ISO/IEC JTC 1/SC 27, ISO/IEC TR 15443-1:2012 — Security assurance framework — Part 1: Introduction and concepts, 2012.

[2.28] ISO/IEC JTC 1/SC 27, ISO/IEC TR 15443-2:2012 — Security assurance framework — Part 2: Analysis, 2012.

[2.29] V. Jayakumar, J. Kannan, N. Kausar, et al., Multicriteria group decision making for prioritizing distributed systems risk factors, Granular Computing 9 (2024) 56, http://dx.doi.org/10.1007/s41066-024-00480-8.

[2.30] L. Jia, Z. Li, Z. Hu, Applications of the distributed systems in renewable power systems: A survey, Energies 17 (2024), http://dx.doi.org/10.3390/en17164160.

[2.31] Joint Task Force Transformation Initiative, Guide for Conducting Risk Assessments, NIST SP 800-30 Rev. 1, 2012, http://dx.doi.org/10.6028/NIST.SP.800-30r1.

[2.32] Joint Task Force Transformation Initiative, Risk Management Framework for Information Systems and Organizations, NIST SP 800-37 Rev. 2, 2018, http://dx.doi.org/10.6028/NIST.SP.800-37r2.

[2.33] K. Kamatchi, E. Uma, Securing the edge: privacy-preserving federated learning for insider threats in distributed systems networks, Journal of Supercomputing 81 (2024) 246, http://dx.doi.org/10.1007/s11227-024-06752-z.

[2.34] A. Khraisat, A. Alazab, et al., Federated learning for intrusion detection in distributed systems environments, Discover Distributed Systems 5 (2025) 72, http://dx.doi.org/10.1007/s43926-025-00169-7.

[2.35] M.S. Lund, B. Solhaug, K. Stølen, Model-Driven Risk Analysis: The CORAS Approach, Springer, 2011.

[2.36] Ministry of Finance and Public Administrations, MAGERIT: Methodology for Risk Analysis and Management of Information Systems, Version 3.0, Madrid, Spain, 2012.

[2.37] National Institute of Standards and Technology, NIST Cybersecurity Framework (CSF) 2.0, NIST CSWP 29, 2024.

[2.38] National Institute of Standards and Technology, NIST SP 800-160 Vol. 1: Systems Security Engineering, 2016, http://dx.doi.org/10.6028/NIST.SP.800-160v1.

[2.39] National Institute of Standards and Technology, NISTIR 8228: Managing Distributed Systems Cybersecurity and Privacy Risks, 2020, http://dx.doi.org/10.6028/NIST.IR.8228.

[2.40] National Institute of Standards and Technology, NISTIR 8259: Foundational Cybersecurity Activities for Distributed Systems Device Manufacturers, 2020, http://dx.doi.org/10.6028/NIST.IR.8259.

[2.41] National Institute of Standards and Technology, NISTIR 8259A: Profile of Selected Security Capabilities for Distributed Systems Devices, 2020, http://dx.doi.org/10.6028/NIST.IR.8259A.

[2.42] National Institute of Standards and Technology, NIST SP 800-160 Vol. 2: Developing Cyber Resilient Systems, 2021, http://dx.doi.org/10.6028/NIST.SP.800-160v2.

[2.43] J. Qi, P. Yang, et al., Sensor-based physical activity recognition using distributed systems, Journal of Biomedical Informatics 87 (2018) 138–153, http://dx.doi.org/10.1016/j.jbi.2018.09.002.

[2.44] R. Rahim, M.A. Chishti, Distributed systems security innovations: Recent technologies, threats, and solutions, SN Computer Science 6 (2025) 593, http://dx.doi.org/10.1007/s42979-025-04106-x.

[2.45] D.G. Rosado, J. Moreno, et al., MARISMA-BiDa pattern: Integrated risk analysis for big data, Computers & Security 102 (2021) 102155, http://dx.doi.org/10.1016/j.cose.2020.102155.

[2.46] D.G. Rosado, A. Santos-Olmo, et al., Managing cybersecurity risks of cyber-physical systems: The MARISMA-CPS pattern, Computers in Industry 142 (2022) 103715, http://dx.doi.org/10.1016/j.compind.2022.103715.

[2.47] D.G. Rosado, L.E. Sánchez, et al., Enabling security risk assessment and management for business process models, Journal of Information Security and Applications 84 (2024) 103829, http://dx.doi.org/10.1016/j.jisa.2024.103829.

[2.48] SecurityMadeIn.lu, MONARC: Method for the Optimised Risk Analysis, 2024.

[2.49] N. Sharma, P. Dhiman, A survey on distributed systems security: challenges and their solutions using machine learning and blockchain technology, Cluster Computing 28 (2025) 313, http://dx.doi.org/10.1007/s10586-025-05208-0.

[2.50] A. Smith, J. Boyens, et al., Cybersecurity Supply Chain Risk Management Practices, NIST SP 800-161 Rev. 1, 2022, http://dx.doi.org/10.6028/NIST.SP.800-161r1.

[2.51] ISO/IEC, ISO/IEC 27002:2022 — Information security controls, 2022.

[2.52] K. Stouffer, J. Falco, K. Scarfone, Guide to Industrial Control Systems (ICS) Security, NIST SP 800-82 Rev. 2, 2015, http://dx.doi.org/10.6028/NIST.SP.800-82r2.

[2.53] L.E. Sánchez, A. Santos-Olmo, et al., MARISMA: A modern and context-aware framework for assessing and managing information cybersecurity risks, Computer Standards & Interfaces 92 (2025) 103935, http://dx.doi.org/10.1016/j.csi.2024.103935.

[2.54] D. Ushakov, E. Dudukalov, et al., The distributed systems impact on smart public transportation, Transportation Research Procedia 63 (2022) 2392–2400, http://dx.doi.org/10.1016/j.trpro.2022.06.275.

[2.55] H. Yang, H. Yuan, L. Zhang, Risk assessment method of distributed systems host based on attack graph, Mobile Networks and Applications 29 (2024) 1504–1513, http://dx.doi.org/10.1007/s11036-023-02198-4.

[3.1] Ehsan Aghaei, Ehab Al-Shaer, Waseem Shadid, and Xi Niu. 2023. Automated CVE Analysis for Threat Prioritization and Impact Prediction. doi:10.48550/arXiv. 2309.03040

[3.2] Fatemeh Akbarian, William Tärneberg, Emma Fitzgerald, and Maria Kihl. 2023. Attack Resilient Cloud-Based Control Systems for Industry 4.0. IEEE Access PP (01 2023), 1–1. doi:10.1109/ACCESS.2023.3259063

[3.3] Seyed Ali Akhavani, Behzad Ousat, and Amin Kharraz. 2025. Open Source, Open Threats? Investigating Security Challenges in Open-Source Software. arXiv:2506.12995 [cs.CR] https://arxiv.org/abs/2506.12995

[3.4] Naeem Allahrakha. 2024. Impacts of Cybercrimes on the Digital Economy. Uzbek Journal of Law and Digital Policy 2 (08 2024), 29–36. doi:10.59022/ujldp.207

[3.5] Benjamin Ampel, Sagar Samtani, Hongyi Zhu, and Jay Nunamaker. 2024. Improving Threat Mitigation Through a Cybersecurity Risk Management Framework: A Computational Design Science Approach. Journal of Management Information Systems 41 (03 2024), 236–265. doi:10.1080/07421222.2023.2301178

[3.6] Khalid Bennouk, Dorra Mahouachi, Nawal Ait Aali, Younès El Bouzakri El Idrissi, Bechir Sebai, and Abou Zakaria Faroukhi. 2025. Dynamic Data Updates and Weight Optimization for Predicting Vulnerability Exploitability. IEEE Access PP (01 2025), 1–1. doi:10.1109/ACCESS.2025.3558990

[3.7] CyberPress. 2025. Over 40,000 CVEs published in 2024. Retrieved June 30, 2025 from https://cyberpress.org/over-40000-cves-published-in-2024/

[3.8] NIS2 Directive. 2025. NIS2 Directive - Official Information Portal. Retrieved June 23, 2025 from https://www.nis-2-directive.com/

[3.9] Martin Eling, Mauro Elvedi, and Greg Falco. 2022. The Economic Impact of Extreme Cyber Risk Scenarios. North American Actuarial Journal 27 (03 2022), 1–15. doi:10.1080/10920277.2022.2034507

[3.10] Adebola Folorunso, Viqaruddin Mohammed, Ifeoluwa Wada, and Bunmi Samuel. 2024. The impact of ISO security standards on enhancing cybersecurity posture in organizations. World Journal of Advanced Research and Reviews 24 (10 2024), 2582–2595. doi:10.30574/wjarr.2024.24.1.3169

[3.11] International Organization for Standardization (ISO). 2022. ISO/IEC 27001:2022 - Information Security, Cybersecurity and Privacy Protection. Retrieved June 23, 2025 from https://www.iso.org/obp/ui/en/#iso:std:iso-iec:27001:ed-3:v1:en

[3.12] Le Ha. 2022. Are digital business and digital public services a driver for better energy security? Evidence from a European sample. Environmental Science and Pollution Research 29 (01 2022), 27232–27256. doi:10.1007/s11356-021-17843-2

[3.13] Aleksy Kwilinski, Lyulyov Oleksii, and Tetyana Pimonenko. 2023. Environmental Sustainability within Attaining Sustainable Development Goals: The Role of Digitalization and the Transport Sector. Sustainability 15 (07 2023), 11282. doi:10. 3390/su151411282

[3.14] Takashi Minohara, Masaya Shimakawa, and Sora Okada. 2025. Security Vulnerability Risk Growth Model based on CVSS 4.0. doi:10.1109/DSN-S65789.2025.00074

[3.15] Marc Mitchell and Lena Kan. 2019. Digital Technology and The Future of Health Systems. Health Systems & Reform 5 (02 2019). doi:10.1080/23288604.2019.1583040

[3.16] Henock Mulugeta. 2023. Context-Based and Adaptive Cybersecurity Risk Management Framework. Risks 11 (05 2023). doi:10.3390/risks11060101

[3.17] National Institute of Standards and Technology (NIST). 2018. Framework for Improving Critical Infrastructure Cybersecurity, Version 1.1. Retrieved September 26, 2025 from https://www.nist.gov/cyberframework

[3.18] National Institute of Standards and Technology (NIST). 2020. Zero Trust Architecture (NIST SP 800-207). Retrieved June 23, 2025 from https://nvlpubs.nist.gov/ nistpubs/specialpublications/NIST.SP.800-207.pdf

[3.19] David Pecl, Yehor Safonov, Zdenek Martinasek, Matej Kacic, Lubomir Almer, and Lukas Malina. 2021. Manager Asks: Which Vulnerability Must be Eliminated First?. In Innovative Security Solutions for Information Technology and Communications. Springer International Publishing, Cham, 146–164.

[3.20] Saqib Saeed, Salha Altamimi, Norah Alkayyal, Ebtisam Alshehri, and Dina Alabbad. 2023. Digital Transformation and Cybersecurity Challenges for Businesses Resilience: Issues and Recommendations. Sensors 23 (07 2023). doi:10.3390/s23156666

[3.21] Kamran Shaukat, Suhuai Luo, Vijay Varadharajan, Ibrahim Hameed, Shan Chen, Dongxi Liu, and Jiaming Li. 2020. Performance Comparison and Current Challenges of Using Machine Learning Techniques in Cybersecurity. Energies 13 (05 2020), 2509. doi:10.3390/en13102509

[3.22] Elham Tabassi. 2023. Artificial Intelligence Risk Management Framework (AI RMF 1.0). doi:10.6028/NIST.AI.100-1 [23] Tenable. 2025. Nessus Vulnerability Scanner. Retrieved June 1, 2025 from https: //www.tenable.com/products/nessus

[3.24] European Union. 2016. General Data Protection Regulation. Retrieved September 26, 2025 from https://eur-lex.europa.eu/eli/reg/2016/679/oj

[3.25] Apostol Vassilev, Alina Oprea, Alie Fordyce, and Hyrum Anderson. 2024. Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations. doi:10.6028/NIST.AI.100-2e2023

[3.26] Wazuh. 2025. Wazuh - Open Source Security Platform. Retrieved June 4, 2025 from https://documentation.wazuh.com/current/user-manual/capabilities/ vulnerability-detection/index.html

[3.27] Ferda Özdemir Sönmez, Chris Hankin, and Pasquale Malacaria. 2022. Attack Dynamics: An Automatic Attack Graph Generation Framework Based on System Topology, CAPEC, CWE, and CVE Databases. Computers & Security 123 (10 2022). doi:10.1016/j.cose.2022.102938

[5.1] O. Mörth, C. Emmanouilidis, N. Hafner, M. Schadler, Cyber-physical systems for performance monitoring in production intralogistics, Comput. Ind. Eng. 142 (2020) 106333, http://dx.doi.org/10.1016/j.cie.2020.106333.

[5.2] A.W. Colombo, G.J. Veltink, J. Roa, M.L. Caliusco, Learning industrial cyber-physical systems and industry 4.0-compliant solutions, in: 2020 IEEE Conference on Industrial Cyberphysical Systems, Vol. 1, ICPS, IEEE, 2020, pp. 384–390.

[5.3] Baseline security recommendations for IoT, 2018, ENISA. URL https://www.enisa.europa.eu/publications/baseline-security-recommendations-for-iot.

[5.4] Y. Maleh, Machine learning techniques for IoT intrusions detection in aerospace cyber-physical systems, in: Machine Learning and Data Mining in Aerospace Technology, Springer, 2020, pp. 205–232.

[5.5] H. Mokalled, C. Pragliola, D. Debertol, E. Meda, R. Zunino, A comprehensive framework for the security risk management of cyber-physical systems, in: Resilience of Cyber-Physical Systems, Springer, 2019, pp. 49–68.

[5.6] J. Geismann, C. Gerking, E. Bodden, Towards ensuring security by design in cyber-physical systems engineering processes, in: Proceedings of the 2018 International Conference on Software and System Process, ICSSP '18, Association for Computing Machinery, New York, NY, USA, 2018, pp. 123–127, http://dx.doi.org/10.1145/3202710.3203159.

[5.7] S. Peisert, J. Margulies, D.M. Nicol, H. Khurana, C. Sawall, Designed-in security for cyber-physical systems, IEEE Secur. Priv. 12 (5) (2014) 9–12,

[5.8] S. ur Rehman, C. Allgaier, V. Gruhn, Security requirements engineering: A framework for cyber-physical systems, in: 2018 International Conference on Frontiers of Information Technology, FIT, IEEE, 2018, pp. 315–320.

[5.9] M. Lezzi, M. Lazoi, A. Corallo, Cybersecurity for industry 4.0 in the current literature: A reference framework, Comput. Ind. 103 (2018) 97–110, http://dx.doi.org/10.1016/j.compind.2018.09.004.

[5.10] Cyber-physical systems security: Limitations, issues and future trends, Microprocess. Microsyst. 77 (2020) 103201, http://dx.doi.org/10.1016/j.micpro. 2020.103201.

[5.11] M. Span, L.O. Mailloux, R.F. Mills, W. Young, Conceptual systems security requirements analysis: Aerial refueling case study, IEEE Access 6 (2018) 46668–46682.

[5.12] A.M. Shaaban, T. Gruber, C. Schmittner, Ontology-based security tool for critical cyber-physical systems, in: 23rd International Systems and Software Product Line Conference - Volume B, SPLC '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 207–210, http://dx.doi.org/10.1145/3307630.3342397.

[5.13] Á.J. Varela-Vaca, D.G. Rosado, L.E. Sánchez, M.T. Gómez-López, R.M. Gasca, E. Fernández-Medina, CARMEN: A framework for the verification and diagnosis of the specification of security requirements in cyber-physical systems, Comput. Ind. 132 (2021) 103524, http://dx.doi.org/10.1016/j.compind.2021.103524.

[5.14] OWASP internet of things project, 2021, Available from OWASP. URL https://wiki.owasp.org/index.php/OWASP_Internet_of_Things_Project.

[5.15] N. Noy, D. Mcguinness (Hrsg.), Ontology Development 101: A Guide To Creating Your First Ontology, Technical Report, Stanford knowledge systems laboratory technical report KSL-01-05 , 2001.

[5.16] W. Yun, X. Zhang, Z. Li, H. Liu, M. Han, Knowledge modeling: A survey of processes and techniques, Int. J. Intell. Syst. 36 (4) (2021) 1686–1720 http://dx.doi.org/10.1109/MSP.2014.90