Pavleska *et al*. *J Surveill Secur Saf* 2024;5:116-39
DOI: 10.20517/jsss.2023.40

Journal of Surveillance, Security and Safety

Original Article

Open Access

Crafting organizational security policies for critical infrastructures: an architectural approach

Tanja Pavleska[1], **Giovanni Paolo Sellitto**[2], **Helder Aranha**[3]

[1]Laboratory for Open Systems and Networks, Jozef Stefan Institute, Ljubljana 1000, Slovenia.
[2]Independent Scholar, Ciampino 00043, Italy.
[3]Independent Scholar, Lisbon 1000, Portugal.

Correspondence to: Dr. Giovanni Paolo Sellitto, Independent Scholar, v.Pignatelli 26 pal.B, Ciampino 00043, Italy. E-mail: gogiampaolo@gmail.com

How to cite this article: Pavleska T, Sellitto GP, Aranha H. Crafting organizational security policies for critical infrastructures: an architectural approach. *J Surveill Secur Saf* 2024;5:116-39. http://dx.doi.org/10.20517/jsss.2023.40

Received: 7 Nov 2023 **First Decision:** 29 Feb 2024 **Revised:** 24 Mar 2024 **Accepted:** 19 Apr 2024 **Published:** 6 May 2024

Academic Editor: Enrico Zio **Copy Editor:** Pei-Yun Wang **Production Editor:** Pei-Yun Wang

Abstract

Critical infrastructures (CIs) are an essential enabler of a nation's well-being and span a multitude of sectors. It is critical to ensure their resilience and protection against security threats, straight from the design phase. Nowadays, CIs take the form of complex socio-technical systems, which rely heavily on digital technology and off-the-shelf components, introducing challenges due to non-composability of security properties. A comprehensive approach for their protection integrates physical security, cybersecurity, risk management, and collaboration with the stakeholders, as they play a key role in identifying and managing vulnerabilities and in the impact evaluation of potential security breaches. Other key requirements in the context of CIs are interoperability, automation and governance, which are often neglected in the process of crafting security policies, as this takes as input the system as-is and disregards architectural design considerations. In this paper, we propose a methodology that takes care of the inter-dependencies between security goals in a given CI and the relevant countermeasures for its subsystems. Our approach considers the relationships between the CI subsystems, focusing on organizational security objectives and the requisite countermeasures to achieve these objectives. The methodology is supported by a context-independent Reference Model for Information Assurance and Security, which can be applied across diverse critical sectors. To complement the methodology, we propose a formal language that enables the verification of the fulfillment of the security goals in a specific solution architecture. The aim is to enable and support CI security, promoting resilience and adaptability in the face of evolving threats. Leveraging the formal language, the proposed methodology can be integrated into an open-source automated tool-chain for the validation of composite systems. Through these contributions, we effectively address the unique security challenges inherent in CI, facilitating automation and interoperability to enhance security and governance in these crucial domains.

Keywords: Security by design, critical infrastructures, modularity, automation, solution design, security policy

INTRODUCTION

Critical infrastructures (CIs) comprise the assets, systems, facilities, networks, and other elements upon which society relies to maintain national security, economic vitality, and public health and safety[1,2]. They encompass vital sectors, including energy, transportation, healthcare, water supply, and telecommunications. CIs should be designed with security and resilience in mind since safeguarding these critical systems against diverse security threats and vulnerabilities is of paramount importance, as their disruption could have far-reaching consequences for the well-being of citizens. Hence, CI security may be defined as the set of processes to reduce the risk that CI is disrupted by intrusions, attacks, or the effects of natural or man-made disasters. This entails the protection of CI through physical means or defensive cyber-related measures[3]. Resilience of a CI, on the other hand, is its ability to adapt to changing conditions. This means being able to withstand and recover rapidly from disruptions, deliberate attacks, accidents, or naturally occurring threats or incidents. A resilient infrastructure must also be robust, agile, and adaptable.

Strengthening the security and resilience of CIs is a shared responsibility between stakeholders, that is the CI owners and operators, and the various government entities and non-government organizations. Governments worldwide have established regulations and standards to enforce the security of CIs, especially regarding information technology (IT), demanding compliance through regular audits and reporting. Protecting CI usually entails the presence of backup systems and well-thought-out contingency plans to ensure the uninterrupted operation of CIs even in the face of disruptions or disasters. CIs, such as energy, transportation, and healthcare, involve a convergence of various technologies, processes, and human factors. Due to the complex and interconnected nature of these systems, when addressing the problem of their cybersecurity and resilience, it is crucial to adopt an interdisciplinary approach, bringing together experts from diverse fields such as cybersecurity, safety engineering, system architecture, human factors, and policy-making to collaboratively analyze, design, and implement robust security and safety measures. This holistic perspective helps in identifying potential vulnerabilities, understanding the interdependencies between different components, and developing comprehensive solutions that account for technical, human, and organizational aspects. The integration of expertise from multiple disciplines enhances the resilience of CIs against evolving threats and ensures a more effective and adaptive response to challenges. Central to this endeavor is a proactive (and cyclic) process for assessing and managing security measures to detect and address the existing vulnerabilities and devise security policies to counter possible threats. However, the modern landscape poses distinctive challenges with consequences that are not easily predictable, requiring frequent changes in security policies and protocols. In addition, CIs are increasingly relying on digital technologies as they incorporate a multitude of off-the-shelf components (COTS). This introduces complex issues due to non-composability of security properties, whereby the security characteristics of individual components do not necessarily extend to the entire system. Furthermore, it cannot be assumed that if a particular security property is observable across all networked subsystems, it holds true for the overall infrastructure.

Many existing methodologies for information security tend to overlook the importance of architectural design, which can leave systems exposed to vulnerabilities. Moreover, these methods can be difficult to implement and often demand specialized security knowledge, which is frequently in limited supply. The difficulties become more pronounced in socio-technical systems, where the interconnection of digital devices and people complicates matters. This complexity underscores the idea that "*the whole is different than the sum of its parts*". To effectively deal with these security challenges, it is essential to adopt interdisciplinary approaches.

Interoperability, automation, and governance stand as prerequisites in the security assurance realm. An effective information assurance model must encompass all these aspects, applied with practicality in mind. A previous study proposed a methodology for automated solution architecture design, known to enhance interoperability and governance[4]. This paper has three primary objectives: first, to design an information assurance model that takes care of the inter-dependencies between security goals in a given CI and the relevant countermeasures for its subsystems. Second, to implement this model in a formal manner, enabling full automation and context-independent applicability. Finally, to demonstrate the feasibility of this solution by integrating the formalism into an open-source tool-chain, making it freely accessible for further validation and widespread use. The aim is to enable and support CI security, promoting resilience and adaptability in the face of evolving threats.

Our work considers eight security goals (Accountability, Auditability, Authenticity/Trustworthiness, Availability, Confidentiality, Integrity, Non-repudiation, and Privacy), which is the widest set of security goals introduced by the Reference Model for Information Assurance and Security (RMIAS). Each of these security goals bears different importance in the context of CIs. Thus, accountability is crucial for tracking and attributing actions or events within the system. Identifying the source of a security incident or a disruption is essential for effective incident response and recovery in CI environments. Furthermore, the ability to trace activities helps in understanding the impact of events and holding responsible parties accountable for any breaches or disruptions. Furthermore, auditing helps in assessing compliance with security policies, identifying vulnerabilities, and detecting any anomalies or unauthorized activities. Auditable records provide valuable insights into the state of the infrastructure, supporting continuous improvement and risk management efforts. Establishing and maintaining the authenticity and trustworthiness of users, entities, and data is vital for ensuring that only authorized and trustworthy entities interact with the infrastructure helps prevent malicious activities and unauthorized access. Trustworthy data is essential for making critical decisions and maintaining the integrity of operations. Ensuring that systems, services, and data are consistently available is essential for maintaining the continuous operation of critical services. Downtime or disruptions in availability can have severe consequences on public safety, national security, and economic stability. Protecting the confidentiality of sensitive information is another key priority, as unauthorized access to classified or proprietary information can lead to severe consequences, including breaches of national security or compromise of critical systems. Clearly, safeguarding confidential data is essential for maintaining the integrity of critical operations, which is, in turn, critical in ensuring the information accuracy and reliability. Preventing unauthorized alterations or manipulations of data is essential for maintaining the overall system trustworthiness. To prevent denial of actions or transactions, non-repudiation ensures that parties cannot disown their responsibilities or activities within the infrastructure. This is crucial for legal and regulatory compliance and for maintaining trust among stakeholders. Finally, although CIs focus on public services and safety, privacy considerations are still important. Ensuring that personal and sensitive information is handled in accordance with privacy regulations is essential for maintaining public trust and complying with legal requirements.

To address the challenges outlined above, the paper is structured as follows: first, we position our work within the state of the art, explaining the complementary and novel aspects with respect to other relevant approaches. Then, we present some theoretical concepts needed to understand the presented work. This is followed by an outline of the methodological considerations and the development of the methodology, which is then applied to a real-world use case to demonstrate its practical viability and usefulness. Finally, we conclude and point to future research directions to be pursued in the upcoming stages of development and application of our approach.

RELATED WORK

The emergence of new societal risks resulting from rapid advancements in information and communication technologies and operational technologies has been the subject of broad analysis and discussions[5–7]. As, according to the European Commission, each member state is required to identify the "National Critical Areas/Sectors" and record and evaluate their systems or components[8]; many critical areas and evaluation methodologies have emerged, tailored to each identified sector[9,10]. In addition to introducing a great variance in the infrastructure design, they usually depend on the knowledge and experience of dedicated security and architecture experts and the manual inputs of numerous individuals.

The application of model-based approaches and the adoption of architectural principles to deal *by design* with various concerns is emerging as a valuable proposition in the critical sectors. They are part of the so-called Model-Based Systems Engineering (MBSE) approaches, which aim to address safety and security concerns in multi-disciplinary systems. The Software Platform for Embedded System (SPES) framework employs such strategies for providing guidelines in developing complex systems, emphasizing aspects such as requirements engineering, system architecture, and verification. Within SPES, to which our work also adheres, safety concerns are addressed through the modeling of safety requirements, hazard analysis, and the integration of safety-related information into the overall system model, whereas security concerns are accounted for through the modeling of security requirements, threat analysis, and the incorporation of security mechanisms at various levels of the system architecture. Although SPES addresses safety and security throughout the system development lifecycle, it is mainly requirement-oriented and focuses on traceability between different artifacts to ensure that safety and security are considered across the models and documentation used during design.

Enterprise Architecture (EA) emerged as a way to provide a holistic and strategic representation (a model) of structure, processes, information, and technology of an organization. The availability of such a conceptual model facilitates stakeholders in increasing their domain knowledge[11], thus enabling the business management to make informed strategic decisions[12]. EA frameworks, such as the Open Group Architecture Framework (TOGAF)[13], Zachman[14], and SAIF[15], have long offered means to ensure sustainability, whether it be technological, environmental, or economic, by enabling informed design-time decision-making based on abstract architectural elements. EA models can be used together with other modeling techniques, such as Unified Modeling Language (UML), Systems Modeling Language (SysML), or other semi-formal techniques, to address different aspects of safety-critical systems. For instance, EA can set the strategic context, aligning business goals with system architecture; UML and SysML can offer visual representation and modeling capabilities, while semi-formal approaches can provide the balance between visual expressiveness and formal analysis. An exemplary embodiment of such an approach can be found in the technique of Integrating the Healthcare Enterprise (IHE) (https://www.ihe.net/), which champions the coordinated use of established standards to enhance the interoperability of healthcare information systems, enjoying international recognition [16,17].

It is important to note that the healthcare is not the sole adopter of a model-based approach that leverages a reference EA. Fields such as Smart Grid and Industry 4.0 employ architectural frameworks grounded on a specific Reference Architectural model based on EA principles, exemplified by Smart Grid Architectural Model (SGAM)[18] and Reference Architectural Model Industrie 4.0 (RAMI4.0)[19], respectively. For instance, SGAM[18] delineates five architectural layers for Smart Grid projects, from the abstraction of field devices to business requirements and interoperability criteria. Furthermore, the work of Gottschalk *et al.*, which devised a methodology in the field of Electronic Mobility, Energy distribution and Smart Cities Architecture, has laid the basis for employing EAs in a wider range of critical sectors[20]. Building on these prior examples, our methodology aims to create a framework for the automated definition of security countermeasures. In addition to architectural methods, network theory approaches have been proposed to assess inter-dependency in CIs[21], mainly concerned with the understanding of the network topology, the correlations and the propagation paths in case of perturbations, remaining completely detached from the management practices and the decision-making by experts. In addition, El-Rewini *et al.* and Chattopadhyay *et al.* have made significant progress in categorizing security risks and creating security-by-design frameworks[22,23]. Some efforts for devising relevant cybersecurity countermeasures, without generalizing them into a specific framework, were also conducted by Checkoway *et al.* Our approach encapsulates these considerations as architectural concerns within the broader context of cybersecurity assurance in CIs[24].

As part of the cybersecurity approaches addressing concerns in critical domains, Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege (STRIDE) scholar[25] is a threat modeling framework used to identify and categorize different types of security threats and vulnerabilities in a system during the design phase. It is commonly applied to evaluate the security of software systems, particularly during the development process. Similarly, although not directly related to functional safety, Highly Extensible and Adaptable Vehicle Environment for Novel Safety (HEAVENS)[26] was developed to create an extensible and adaptable vehicle environment to enhance safety features in automobiles.

Our work goes further by accounting for the entire system lifecycle while introducing abstract countermeasure categories that can be addressed proactively at the design stage. This not only offers a more fine-grained blueprint for system design but also resolves the composition challenge for security properties at design time. Notably, this is done by leveraging the modularity of the system architecture (i.e., its building blocks), i.e., taking advantage of their ability to be reused as separate components across other architectures. In our previous work, we provided a means to evaluate the cybersecurity of CIs by considering the existing architectural design[4]. With the approach presented here, we can also automate the evaluation process by extending the architectural framework with proactive measures of addressing cybersecurity concerns.

The approach presented here integrates the considerations outlined above into the architectural aspects of CIs. A crucial aspect of the formalism developed as part of our approach is that it is not confined to a single sector. It can be applied to other CIs, as exemplified by the Integrating the Energy System (IES) project[27–29] in the realm of Smart Grids. IES focused on sustainable smart grid projects by promoting interoperability, relying on the IHE methodology. While sharing a common foundational base, our work takes a comprehensive perspective, supplementing the concept with a methodological approach for the automated definition of cybersecurity countermeasures. The major goal of this approach is to provide system designers and architects with a more comprehensive perspective, offering predefined countermeasure categories and establishing connections between these categories and security objectives.

THEORETICAL BACKGROUND

This section introduces some theoretical concepts that will be exploited in the rest of the paper. They are divided into two types: architecture-related and construct-relevant, for the formal modeling approach.

Architectural constructs

The principles, terminology, and artifacts surrounding the use of architectural constructs employed in this work adhere to TOGAF[30]. To define the security architecture for the CI, we utilize the Architecture Development Method (ADM) of TOGAF, which involves identifying stakeholders, defining security concerns, and creating a comprehensive security architecture vision. the Security Architecture phase of TOGAF is then leveraged to develop a detailed blueprint for securing the CI. This includes specifying security requirements, designing security controls, and defining security patterns. Two foundational concepts underlie the TOGAF architectural design process: the *Reference Architecture* and the *Solution Architecture*.

A **Reference Architecture** serves as a conceptual framework that offers guidance and options for crafting specific system architectures and solution implementations. In this context, it encompasses the entirety of available architecture elements, which are instrumental in shaping a solution architecture.

A **Solution Architecture**, on the other hand, delves into the specifics of business operations and activities, exploring how information systems and technology underpin and enhance them. Typically, it applies to a single project or organization. In our context, it translates into a coherent assembly of architectural components, often referred to as *profiles*, designed to address a particular business challenge.

Thus, Solution Architecture Design refers to the process of creating a comprehensive and structured solution for a particular problem or set of requirements. It involves designing the architecture of a system or solution that addresses specific needs and aligns with organizational goals, such as the crafting of organizational security policies. This process spans multiple layers, including technology, data, application, and business architecture.

Formalism building constructs

As CIs may encompass a wide variety of systems, in order to ensure interoperability and standardized approach, our work relies on the foundations established by IHE[31]. More in detail, we draw from its Information Technology Infrastructure Technical Framework and the related standards it incorporates and incorporate it in the overall governance scheme provided by TOGAF.

The following definitions from the IHE framework are relevant for our approach:

An **IHE Actor** represents a functional component within an organization. The standard-based specification governing the interactions between IHE Actors is called an **IHE Transaction**. A high-level functional unit consisting of interconnected IHE Transactions, which is designed to address specific IT infrastructure requirements for a particular use case, is called **IHE (Integration) Profile**.

A profile can incorporate functional dependencies from other profiles, as specified in grouping rules, and may inherently possess optionality and variability. Profile variability suggests that the range of transactions directly shapes and defines the functionality of the profile.

Grouping rules play a pivotal role in the organization of integration profiles. They are critical for the resolution of inter-profile dependencies, simplifying the decision-making process related to solution design. These predefined inter-profile dependencies aid in streamlining solution design by minimizing the decisions required during the design phase. For instance, if a grouping rule specifies that profile A depends on profile B, this implies that if profile A is integrated into a solution, profile B must also be included. The grouping rules are expressed in a straightforward language.

Grouping operations are instrumental in merging various profile functionalities in alignment with grouping rules. This process enables the creation of more complex use cases and lies at the core of solution architecture design.

In the IHE framework, the practical design of a solution architecture involves two key steps:

- The translation of business requirements into a solution vision, which generates an initial subset of profiles directly derived from the use cases and their requirements.
- The derivation of a comprehensive set of IT specifications, which expands upon the initial profiles by resolving all functional interdependencies, as outlined in the grouping rules.

Our work adopts the terms Actor, Transaction, and Profile as functional elements within the general domain of CIs, adapting the IHE concepts to suit the specific CI context. In devising the policies to apply, we rely on some well-established IHE profiles, such as the Audit Trail and Node Authentication (ATNA) Profile, which specifies the foundational elements needed by all forms of secure systems: node authentication, user authentication, event logging (audit), and telecommunications encryption. It is also used to indicate that other internal security properties, such as access control, configuration control, and privilege restrictions, are provided. Furthermore, the overall process adheres to common standards and regulatory guidelines on security, privacy and safety, industrial systems control and automation, and software architecture description, such as ISO/IEC 27002, ISO/IEC/IEEE 42010:2022, ISA/IEC 62443, IEC 61508, NIST SP 800-53 and European Union Agency for Cybersecurity (ENISA) recommendations and guidelines, among others. However, it is important to note that when implementing security and safety measures in CIs, organizations need to consider a combination of these standards based on their specific industry and operational context. Regular updates and compliance with the latest versions of relevant standards are essential to address evolving cybersecurity and safety challenges.

The novelty of our work is that it provides a way for the former to be formalized and the latter to be entirely amenable to automation, offering a structured and efficient approach to solution architecture design. In that process, the approach of TOGAF is used to identify security requirements across different architectural layers, including business, data, application, and technology layers, whereas IHE profiles are aligned with specific security requirements at each layer. In that way, we can address security goals related to confidentiality, integrity, authentication, accountability, authorization, availability, non-repudiation, and privacy across the information systems in the CI. As TOGAF emphasizes the importance of compliance with industry standards, IHE standards and profiles are integrated as part of the overall compliance strategy within the CI. The following sections present the practical implementation of these concepts and principles.

METHODOLOGICAL CONSIDERATIONS

This section introduces the major methodological considerations on which our work relies and specifies both the points of improvement and the novel elements that unite them into a single framework.

Principles of security design

At the core of the security principles guiding the design of the proposed methodology is the RMIAS[32], which identifies four security dimensions of an Information System: *Life Cycle*, *Information Taxonomy* (Classification), *Security Goals* and *Countermeasures*, as illustrated in Figure 1 and sublimed in Table 1.

Unlike traditional threat-based models, which rely on statistical data about real-world threats, RMIAS follows a goal-based approach. Instead of focusing on threats, it starts from defining a set of security objectives to establish a robust methodology to cater for security. While threat-based models are suitable in the case of existing systems, where a history of past threats exists, their effectiveness is limited to the past experiences of analysts. Notably, there is a shortage of models designed to address security properties during the design phase of systems or security solutions: our approach aims to address this gap by building upon RMIAS, extending it with a Countermeasures category containing reactive safeguards to enhance system security. In contrast to threat-focused models, this goal-based method offers some advantages, including the ability to communicate security concepts effectively with stakeholders who may lack technical expertise. Their involvement not only facilitates the clarification of business goals and of the assets to protect but also highlights the need for some trade-offs early in the design phase. This, in turn, streamlines the definition of security policies and enhances organizational self-awareness, a crucial ally in cybersecurity policy management. While RMIAS does not explicitly incorporate threats, its analysis across the four dimensions yields concrete Countermeasures. These can be as concrete and threat-oriented as the system design or the organization requires (see examples in column 6 of Table 2). They can be interpreted as *intended system behaviour* and used for providing sound security guidance for implementation.

RMIAS takes a holistic approach to information assurance, seeking a continuous improvement of security

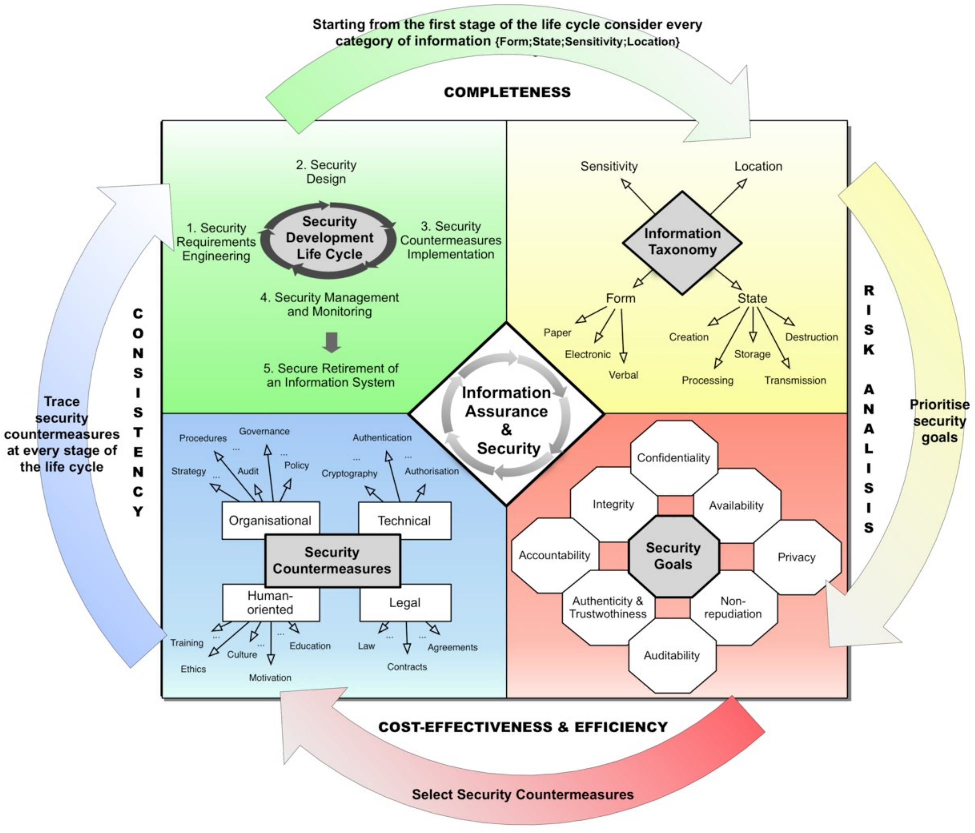

Starting from the first stage of the life cycle consider every category of information {Form;State;Sensitivity;Location}
COMPLETENESS
2. Security Design
1. Security Requirements Engineering
Security Development Life Cycle
3. Security Countermeasures Implementation
4. Security Management and Monitoring
5. Secure Retirement of an Information System
Sensitivity
Location
Information Taxonomy
Form
State
Paper
Electronic
Verbal
Creation
Destruction
Storage
Processing
Transmission
RISK ANALISIS
Prioritise security goals
Information Assurance & Security
CONSISTENCY
Trace security countermeasures at every stage of the life cycle
Procedures
Governance
Strategy
Audit
Policy
Authentication
Cryptography
Authorisation
Organisational
Technical
Security Countermeasures
Human-oriented
Legal
Training
Ethics
Culture
Motivation
Education
Law
Contracts
Agreements
Confidentiality
Integrity
Availability
Accountability
Security Goals
Privacy
Authenticity & Trustwothiness
Non-repudiation
Auditability
COST-EFFECTIVENESS & EFFICIENCY
Select Security Countermeasures

Figure 1. The RMIAS model, from Cherdantseva and Hilton. "A Reference Model of Information Assurance & Security"[32]. RMIAS: Refer-ence Model for Information Assurance and Security.

Table 1. Info taxonomy, security goals and security countermeasures dimensions in RMIAS

Dimension	Attributes	Values
Information taxonomy	1. Form	Paper, electronic, verbal
	2. Sensitivity	Top secret, secret, confidential, protect, unclassified
	3. Location	Controlled, partially controlled, uncontrolled
	4. State	Creation, transmission, storage, processing, destruction
Security goals	5. Security goal	Accountability, auditability, authenticity/trustworthiness, availability, confidentiality, integrity, non-repudiation, privacy
Security countermeasures	6a. Countermeasure | type	Organizational, uuman-oriented, technical, legal
	6b. Countermeasure | description	(Free text)

Derived from *UK Government classification and marking scheme, as employed in the case study presented in*[32]. RMIAS: Reference Model for Information Assurance and Security.

traits throughout its *Life Cycle dimension*. This ensures that security is not something achieved once and for all, but rather an ongoing process, encompassing planning, implementation, monitoring, and continuous improvement. This characteristic makes RMIAS suitable for CIs, where the components are added and removed based on the specific needs, and the system varies and adapts over time.

In this paper, we extend RMIAS to specify and support the evaluation of security properties in the specific case of CIs composed of COTS. Furthermore, we formalize this process, introducing a practical evaluation tool that can support the integration process by assessing the composite system, to check if the intended security

Table 2. The structuring of an Information Security Policy document using the RMIAS method (Excerpt reprinted from Cherdantseva and Hilton[32])

ID	1. Form	2. Sensitivity	3. Location	4. State	5. Security goal	6. Security countermeasure type: description
1	Paper	Secret	Controlled	Creation	Confidentiality	Organizational: apply protective marking (avoid over or under marking)
2	Any	Any	Controlled	Destruction	Availability	Organizational: no information, held on any media, can be destroyed unless it has been reviewed
3	Paper	Confidential	Partially controlled	Transmission	Accountability, confidentiality	Organizational: documents marked CONFIDENTIAL may be taken home only with written approval of a designated person. All actions with documents marked CONFIDENTIAL to be logged
4	Electronic	Protect	Uncontrolled	Storage, processing	Confidentiality, integrity	Technical: any data marked PROTECT must be encrypted when taken outside the office

RMIAS: Reference Model for Information Assurance and Security.

properties were preserved or went lost.

RMIAS places a significant emphasis on the creation and enforcement of security policies and procedures to guide and standardize security practices. These policies are instrumental in defining rules and responsibilities for information security. However, within RMIAS, these security policies primarily fall under the Lifecycle dimension and are not designed for modular use by other organizational processes. In contrast, a model developed by Zuccato *et al.*[33] offers a versatile approach that supports the modular application of security safeguards, making it adaptable to diverse organizational requirements and threat landscapes. The core concept of this model involves breaking down security safeguards into smaller, self-contained modules, each addressing a specific facet of security, such as access control, data encryption, intrusion detection, or network monitoring. This modular approach enables organizations to mix and match these modules to create more complex "security requirement profiles", tailored to their specific security needs.

Several significant parallels exist between the approach of Zuccato *et al.*[33] and RMIAS[32]:

- *Security goals* in the approach of Zuccato align with RMIAS Security goals.
- *Security safeguards* are equivalent to RMIAS Countermeasures.
- Information classification in the approach of Zuccato is akin to RMIAS's concept of Sensitivity within the Information Taxonomy dimension.

However, in addition to RMIAS, the approach proposed by Zuccato *et al.* introduces modularity and customization in the security design, which aligns with the realities and constraints faced by contemporary organizations[33]. It strives for practical security solutions rather than aiming for an unattainable perfect security standard. The incorporation of modular security requirements provides predefined, cohesive, and expert-vetted security mechanisms during system conception. This approach also reduces the demand for extensive security expertise in maintaining information systems. We must also highlight that the proactive approach adopted by Zuccato *et al.* leads to the adoption of the term Safeguards, instead of Countermeasures, as safeguards are wider in their scope and proactive, while countermeasures are reactive[33]. This is also a major difference between the RMIAS and our approach, as the dimension of Countermeasures in RMIAS mainly accounts for the reactive elements of countering cybersecurity threats.

Introducing modularity in the security properties

The profiles employed in our work acknowledge the interdependencies and synergies between modular safeguards and enhance them with higher-level security mechanisms, all aimed at achieving organizational security goals. However, the model proposed by Zuccato *et al.*[33] does not provide a comprehensive framework for the iterative integration of these security safeguards into the life cycle of an organization, which, on the other hand, is well-covered by RMIAS[32]. By integrating modular security requirements into the RMIAS model, a more adaptable and pragmatic approach to information security can be supported. Thus, merging these two complementary approaches into a unified framework can overcome their individual limitations.

It is worthwhile noting that the process of introducing modular security properties also adheres to the procedure for ensuring the interlinkage and usability of various system abstraction levels as defined by the Sustainability Performances, Evidence and Scenarios (SPES) framework, particularly within safety and security assessment. More concretely, SPES utilizes a hierarchical modeling technique, where systems are decomposed into subsystems and components, while each level of the hierarchy is a specific abstraction of the system. Our approach can be incorporated within SPES to use the information of detected potential hazards, failure modes, and safety-critical components and identify security threats, vulnerabilities, and countermeasures in order to ensure that security considerations are embedded throughout the entire system architecture. This ensures that models at different abstraction levels are represented in a standardized and interoperable manner, enhancing communication and collaboration among stakeholders.

Defining countermeasure categories

One of the main ideas of the empirical approach presented in Zuccato *et al.* is the ex-ante definition of abstract *countermeasure categories*[33]. Such a task requires practical knowledge on cybersecurity and is, therefore, performed by security experts. The resulting artifact is named Security Mechanism Reference Table (SMRT), in which the categories of countermeasures are grouped by *scope*: System-dependent, Interdependent and Network-dependent. *System-dependent* countermeasures relate to the internal behavior of the individual systems. *Interdependent* countermeasures, on the other hand, refer to the interactions of cross-systems. *Network-dependent* countermeasures address the generic interactions between any two systems and encompass the pervasive infrastructure of the System of systems - the data network. The procedure of devising the SMRT and its concrete form will be explained in the next section. After defining the categories that are tailored to the organization, the next step is establishing the dependencies between Security goals and Countermeasure categories. This is reusable and, thus, generic information at the level of the organization that facilitates the definition of concrete countermeasures in future projects.

In this paper, we extend the concept of *dependency* between Security goals and Countermeasure categories to enable the evaluation of security countermeasures during solution architecture design.

Evaluation frame of reference

To ensure that the effectiveness of security countermeasures against the relevant security goals can be measured in a consistent and reproducible manner, it is crucial to establish a stable reference framework. In this context, we suggest a reference framework designed for assessing security during the design or redesign phases. To create this reference framework, the following functionalities need to be in place:

- The security countermeasures should be described in a more functional rather than technical way, in order to foster reuse;
- The definition of dependencies between Security goals and Countermeasure categories should be clear and stable, allowing comparable results;
- The definition of the scope for the Countermeasure categories should be adapted to encompass the whole system;
- The countermeasure categories should act as a proxy to evaluate if the concrete (solution-dependent) countermeasures meet the intended security goals.

The methodology presented in the upcoming section facilitates the implementation of these functionalities and the establishment and maintenance of such a reference framework as an integral part of the security policy of a CI. The creation of this evaluation framework is described next.

METHODOLOGY

As outlined previously, one of the primary goals of RMIAS is to provide support for crafting and refining the Information Security Policy of an organization[32]. As part of the theoretical background, we elucidated the principles that guide the shaping of such a policy. To offer a practical glimpse into the content of a security policy, in Table 2, we present a fragment of this document. In order to construct a security policy as the one presented in Table 2, it is necessary to begin by comprehensively understanding the CI, its components, operations, dependencies, and potential vulnerabilities. This involves identifying the key assets, systems, and processes that are critical to the functioning of the infrastructure and categorizing them according to their Form, Sensitivity, Location and State (the first four columns of Table 2). Then, using the Security Goals dimension of RMIAS (column 5 in Table 2), the overarching security objectives are identified, which are to be achieved to protect the CI.

Once the security goals are identified, a thorough risk assessment is conducted to identify potential threats, vulnerabilities, and risks. This involves analyzing both internal and external threats and the likelihood and impact of various security incidents. Based on the identified risks and security goals, a set of countermeasures is defined to mitigate and manage the risks effectively. These countermeasures should address specific security concerns and vulnerabilities identified during the risk assessment process. From an organizational perspective, the Information Taxonomy dimension of RMIAS is used to categorize the security requirements based on the sensitivity and criticality of the information assets involved. This involves defining specific security controls, policies, and procedures that need to be implemented to protect these assets, as represented by the last column of Table 2. The identified countermeasures and security requirements are then incorporated into the organizational security policy while ensuring that the security mechanisms are aligned with the security goals and effectively address the identified risks.

In order to ensure that the security policy is effectively implemented and enforced across the CI, governance processes and compliance measures need to be defined in a proactive framework for security and safety assurance. This involves establishing roles and responsibilities, defining reporting structures, and monitoring compliance with security standards and regulations. Regularly reviewing and updating the security policy is also needed to adapt to evolving threats, technologies, and regulatory requirements in order to ensure that the CI remains resilient and secure against emerging security challenges. However, the countermeasures dimension of RMIAS cannot respond to these challenges, as it is based on a reactive account of the system vulnerabilities. To enhance RMIAS with a comprehensive framework for evaluating security and the requisite methods for conducting security assessments, we propose a systematic, actionable step-by-step process (reported in the form of a high-level flowchart in Figure 2), which plays a pivotal role in both formulating the Security Policy and evaluating the solution architectures.

Modeling security requirements

It is important to note that the RMIAS expansion does not merely hold profound implications for the security of the CI but also bears significance in the broader context of organizational management. From a technical perspective, this augmentation embodies two fundamental components. First, it involves the extension of the existing RMIAS tuple with an additional dimension, enabling the systematic categorization of countermeasures. Second, it facilitates the creation of a formal representation for this entire process. Henceforth, this enriched tuple is designated as “Security requirement”. The culmination of this effort is a comprehensive set of Security requirements, forming the bedrock of the Security Policy.

The newly introduced dimension, supplementing the existing four RMIAS dimensions, is denoted as the *Countermeasure category*. This category, from a formal viewpoint, includes a predefined set of values. Selecting these values is based on established dependencies within the broader spectrum of Security goals. As a result, the process involves classifying the information to be protected based on the Information Taxonomy, identifying the

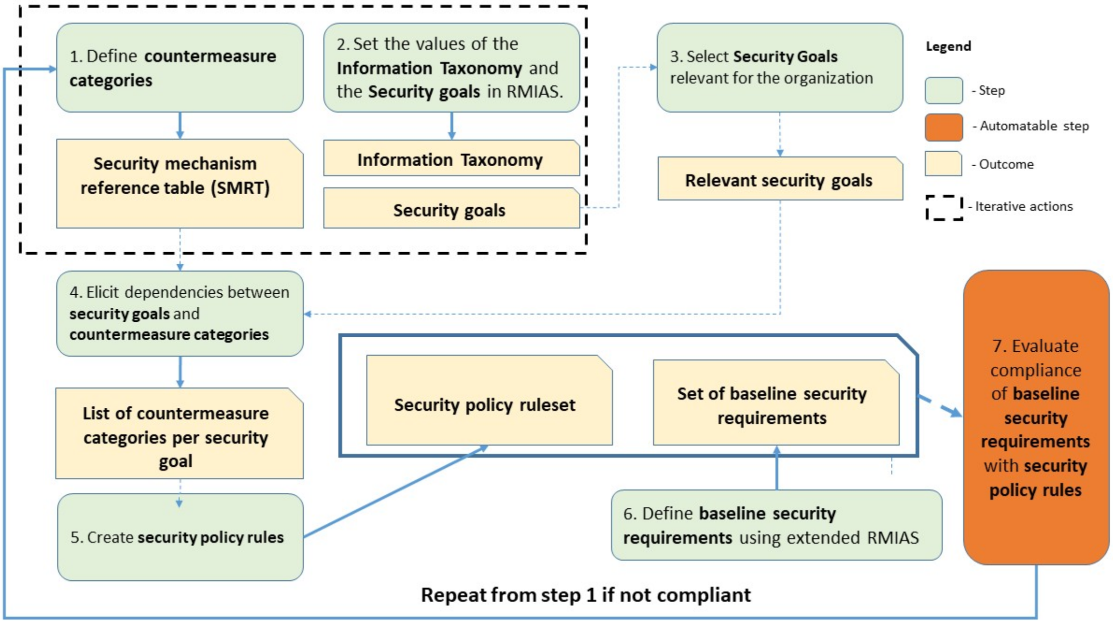

1. Define countermeasure categories
Security mechanism reference table (SMRT)
2. Set the values of the Information Taxonomy and the Security goals in RMIAS.
Information Taxonomy
Security goals
3. Select Security Goals relevant for the organization
Relevant security goals
Legend
- Step
- Automatable step
- Outcome
- Iterative actions
4. Elicit dependencies between security goals and countermeasure categories
List of countermeasure categories per security goal
5. Create security policy rules
Security policy ruleset
Set of baseline security requirements
6. Define baseline security requirements using extended RMIAS
7. Evaluate compliance of baseline security requirements with security policy rules
Repeat from step 1 if not compliant

Figure 2. Building a security policy.

subset of Security goals through rigorous risk analysis, and deriving the System-dependent, Interdependent, and Network-dependent Countermeasures categories from a table of pre-defined dependencies. This extension further introduces an additional task: identifying suitable modular countermeasures from the SMRT for each specific category.

Each *Security requirement*, for the sake of comprehensiveness, encompasses the following attributes: *Form*, *Sensitivity*, *Location*, *State*, *Security Goal*, *Countermeasure Category*, and *Countermeasure*.

The concept of a *Security requirement* stands at the core of our work and plays a pivotal role in shaping the definition of a Security Policy. Its presence confers several notable advantages, including:

- The capacity to classify and reuse security requirements;
- A focused reassessment of security properties during the design phase;
- The harmonization of system requirements with the intended security goals.

Moreover, this approach lends itself to formalization, thereby enabling the automated evaluation of a specified set of quality attributes.

The next section describes how to develop the Security Policy documents, starting from the security requirements.

Building a security policy

The development of a Security Policy document typically requires the expertise of security professionals. In this work, we provide empirical evidence regarding the practicality of Security Policy and its applicability in everyday project design scenarios. Our objective is to steer away from a scenario where Security Policies serve as neglected repositories, seldom utilized and rarely revisited, moving towards a scenario where they are stable and equipped with systemic tools for compliance, maintenance, and evolution. Figure 2 offers a visual representation of the Security Policy development process.

Table 3. Target security goals *vs.* dependent countermeasure categories

Security goal - target	Countermeasure category - dependencies
Confidentiality	One of: • Stored data confidentiality, key management • Access control, authentication, audit trails
...	...

Step 1 revolves around the definition of Countermeasure categories, organized by scope (e.g., System-dependent, Interdependent, and Network-dependent), and tailored to the specific context at hand. Following the approach in[33], the outcome of this step is a SMRT specifically designed for CI applications, designed by security experts.

Step 2 is targeted at establishing the values of the Information Taxonomy (classifiers) and the Security goals within the RMIAS. While the former can be adapted to the CI's context, the latter typically comprises the eight major RMIAS security goals: Accountability, Auditability, Authenticity/Trustworthiness, Availability, Confidentiality, Integrity, Non-repudiation, and Privacy (see[32] for detailed description and rationale).

In **Step 3**, the organization must determine its level of ambition, selecting which security goals it aims to achieve based on the output from Step 2 and performing business risk analysis. Although, in principle, all identified security goals should be pursued, this step serves as the initial point for discussions among security experts, architects, and management, considering practical business constraints. The reason for this is that it often appears that business constraints may (at least temporarily) dictate different ambitions for the security requirements.

In **Step 4**, the experts scrutinize the security goals agreed upon in Step 3. This analysis aims to identify the dependencies of the countermeasure categories to be included in the SMRT. The outcome, a list of dependent countermeasure categories per target security goal (see an example in Table 3), should be a stable artifact within the Security Policy document, constituting the first pillar of the *evaluation frame of reference*. These dependencies can subsequently be formalized to enable the automated assessment of security properties.

Step 5 creates the Security Policy rule set, forming the second pillar of the *evaluation frame of reference*. To evaluate the security properties of a given solution architecture, one approach is to ascertain whether the Countermeasure categories in the concrete security properties can address dependencies of all security goals identified in Step 4. However, this approach focuses on one-to-one dependencies, excluding broader interdependencies.

Therefore, we introduce a *set of policy rules* designed to manage all artifacts simultaneously: the SMRT, the collection of security requirements, and the dependency table. These rules not only ensure compliance with security goals and countermeasure categories but also streamline security evaluation during the design phase. When it comes to security requirements, specific rules cannot be expressed in terms of concrete countermeasures since those are not predetermined. To address this, we use established countermeasure categories as substitutes or proxies in our rule formulations.

In our work, we formalize these rules to enable a more detailed evaluation of security requirements within solution architecture design.

Step 6 centers around the definition of baseline security requirements. Employing the methodology delineated here, the security team orchestrates a second round of interactions between security experts, architects, and business stakeholders, critically assessing the trade-offs presented by each security requirement.

Step 7 represents a major milestone in our work, offering a tool to validate the alignment of baseline security requirements with dependencies and the policy rule set established in previous steps. This step ensures not only the consistency of baseline countermeasure requirements but also their coherence with the Security Policy rule set, establishing holistic consistency for the Security Policy.

Finally, we introduce a feedback mechanism, underlining the Information System Security Life-cycle dimension within the RMIAS model to guarantee the active maintenance, evolution, and relevance of the Security Policy. The subsequent section expounds on the process of evaluating solution architecture by leveraging the resultant Security Policy.

Evaluating architecture solutions

Building upon the RMIAS model and the formulation of a stable, coherent Security Policy, we have provided a framework that facilitates a consistent and repeatable security assessment of solution architectures. In a similar manner, we can automate the evaluation process by extending the architectural framework that was presented in our previous work[4] based on the mechanisms depicted in Figure 2.

The proposed structure for security requirements, detailed in Section , can be expressed in a formal syntax. Therefore, we can expand the *profile* construct defined earlier, in order to enable modularity and interoperability.

Due to their modular and self-contained nature, we can draw a clear parallel between a system assembled from a set of base profiles and a complex system resembling a CI. Thus, the findings derived from solutions composed of profiles are readily applicable to the CI context.

Let us begin by assuming that, for the purposes of this paper, solution architectures are assembled from modular profiles, which act as their component building blocks. As explained earlier, these profiles incorporate various modular security requirements and undergo the same consistency tests described in Steps 6 and 7 of Figure 2. Nevertheless, the design team, ideally including individuals with security knowledge and expertise, can introduce additional, project and CI-specific security requirements as required. Consequently, after assembling the profiles to compose a new solution, the following security requirements become integral to the architecture:

- **Baseline security requirements**: originating from the Security Policy document and, by definition, applicable to all solutions;
- **Profile security requirements**: inherent to the selected profiles;
- **CI-specific security requirements**: project-specific and CI-specific requirements.

The entirety of these security profiles is evaluated as a cohesive unit, irrespective of their origin. Figure 3 outlines the steps of the proposed evaluation process.

In **Step 8**, the architectural team conducts CI-specific analysis following the extended RMIAS process, as previously elaborated. This step may also reveal additional security requirements.

Step 9 involves the evaluation of the merged set of security requirements (baseline, built-in, and project-specific) against the same *frame of reference* provided by the Security Policy, as described earlier. Incorporating baseline security requirements is obligatory, but it does not guarantee consistency, compatibility among all security requirements, or compliance with the specific solution. Therefore, the inclusion of baseline security requirements is performed as a mandatory step.

Finally, **Step 10** presents the feedback loop of the evaluation, enabling architects to follow a test-driven, iterative

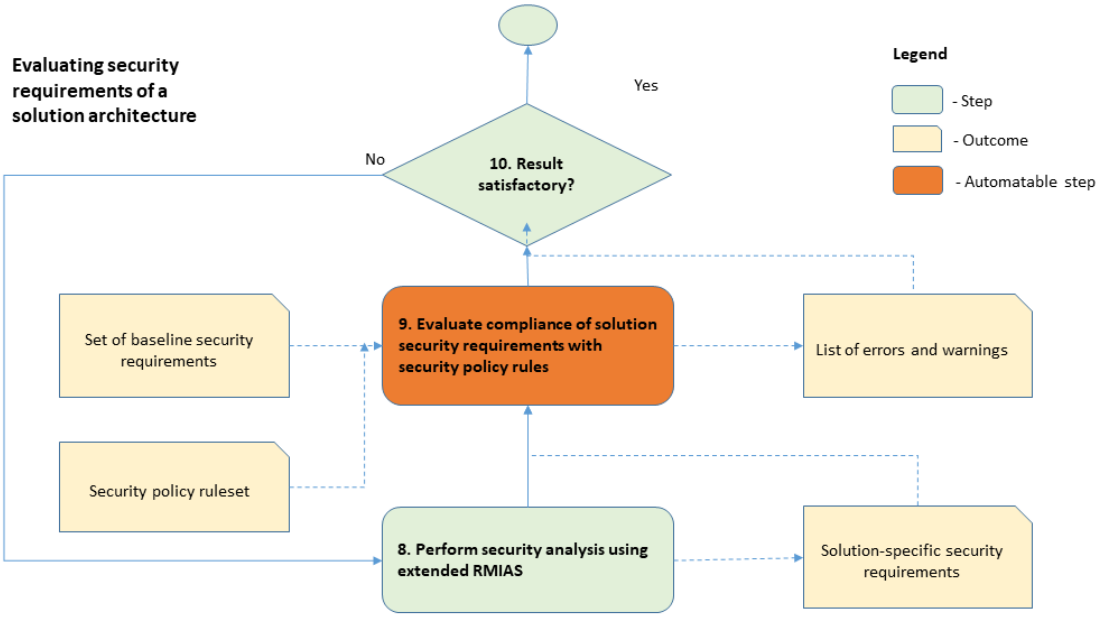

Evaluating security requirements of a solution architecture
Yes
No
10. Result satisfactory?
Legend
🟩 - Step
🟨 - Outcome
🟧 - Automatable step
Set of baseline security requirements
9. Evaluate compliance of solution security requirements with security policy rules
List of errors and warnings
Security policy ruleset
8. Perform security analysis using extended RMIAS
Solution-specific security requirements

Figure 3. Evaluating security requirements of solution architectures.

approach in building the solution architecture.

As illustrated in Figures 2 and 3, Steps 7 and 9 are conducive to automation. In the subsequent section, we delve into the formal mechanisms that facilitate this automated process. It is important to underscore that the entire design and evaluation cycle is carried out either by the architectural team of an organization or by experts, albeit with the assistance of workflow automation tools. This consideration is crucial for leveraging efficient reasoners and tools, which are grounded in propositional logic and calculus. Moreover, it aids in devising new countermeasures, such as in Step 8, where the predefined set of countermeasures may not suffice to address the risk. In such cases, a fully automated workflow would not yield a conclusive outcome.

FORMAL SYNTAX

The complexity of an architecture increases proportionally to the number of components, constraints and requirements required by the business cases. As a result, architects often *describe* the architectural building blocks in terms of concepts and requirements in natural language. One such framework used for that purpose is the IHE Framework, where the architectural building blocks are expressed in terms of profiles. However, when the complexity increases, a tool is needed to *formally* evaluate the impact of the architectural change on the overall system created or assess the compliance with a given set of requirements. Therefore, in a previous work, we introduced a formal language to express the IHE profiles and their composition, enabling automatic reasoning over architectural constructs[34].

The formalism specifies the expression of architectural elements through a specific semantics [Table 4]. At the core of the syntax of this proposed language is the Profile, which can be combined with other profiles using Transactions. Similar to IHE, transactions enable the structuring of reference and solution architectures. The way to structure such architectures is defined by the construct Rule, which expresses dependency between profiles. Thus, a reference architecture is expressed as a chain of Rules, as it will be explained through the use case presented in the following section.

Table 4. Correspondence between the IHE Syntax and the SGAM profiles, derived from Masi *et al.*[4]

Architectural element	*ArchElement*	::=	ArchitecturalElement $Name : Name$ { Function: $Function^*$ Rules: {$Rule^*$ } $Profile^+$ }
IHE profile	*Profile*	::=	Profile $ProfKey$ { Name: $Name$ Function: $Function$ Description: $Description$ actors: $Actor^+$ transactions: $Transaction^+$ domains: $Domain^*$ quality_attributes: $QAttr^+$ security_requirements: {$SecRe^+$ }}
Actor	*Actor*	::=	*String*
Transaction	*Trans*	::=	($Actor$, $String(ProfName?)$, $Actor$)
Domain	*Dom*	::=	*String*
Quality attributes	*QualityAtt*	::=	Identifier [String]
Rule	*Rule*	::=	$Ident\ (Actor\vert'*')/Profile \rightarrow (Actor\vert'*')/Profile$ $\vert \leftarrow$ Profile
Function	*Function*	::=	*String*
Identifier	*Identifier*	::=	*String*
Name	*Name*	::=	*String*

IHE: Integrating the Healthcare Enterprise; SGAM: Smart Grid Architectural Model.

It is important to note here that, although IHE has initially been envisaged for use in the healthcare sector, it can be generalized to other critical sectors and contexts as well, as it has been demonstrated for energy[28], smart grids[34], and road infrastructures[35]. Similarly, the IHE profile and the formal syntax describing the architectural elements are generic and reusable constructs across these domains.

USE CASE: THE HEALTHCARE SECTOR

To demonstrate the application of the methodology presented in this paper, this section specifies the development of an architecture for the healthcare sector as a representative of a CI. The specific focus will be on the regional healthcare system exchanging patient data among clinics to shorten patient waiting time for treatments.

Within regional healthcare systems, the exchange of documents among clinics and a *central registry* is performed using a pattern known as *Cross Enterprise Document Sharing* (*XDS*) [Figure 4]. The XDS pattern facilitates the submission and retrieval of documents, ensuring a seamless information flow. Namely, the clinic, acting as the (*document source*), *submits* a document to the *document repository*. This repository extracts the essential metadata pertaining to the document and *feeds* it to a *document registry*. A *document consumer*, typically the workstation of a doctor, *queries* the registry using metadata to *retrieve* documents that match the query criteria.

Given the critical need for accountability in healthcare projects, it is mandatory for each clinic to execute all transactions through a Secure Gateway, often implemented as a Transport Layer Security (TLS)-enabled load balancer. Moreover, each transaction must undergo *strong authentication*, typically involving the inclusion of an identity token (such as Security Assertion Markup Language - SAML, or JSON Web Token - JWT) obtained from an Identity Management system. Notably, for historical reasons, an External Source is in place to dispatch batches of documents, including digitized versions of existing paper-based documents.

From an accountability perspective, a clear imbalance emerges among data sources: one group (the clinics) operates within a strongly authenticated framework, while the other (the External Source) lacks authentication. This situation is addressed by the project through the implementation of *security zones*, aligning with the ISO

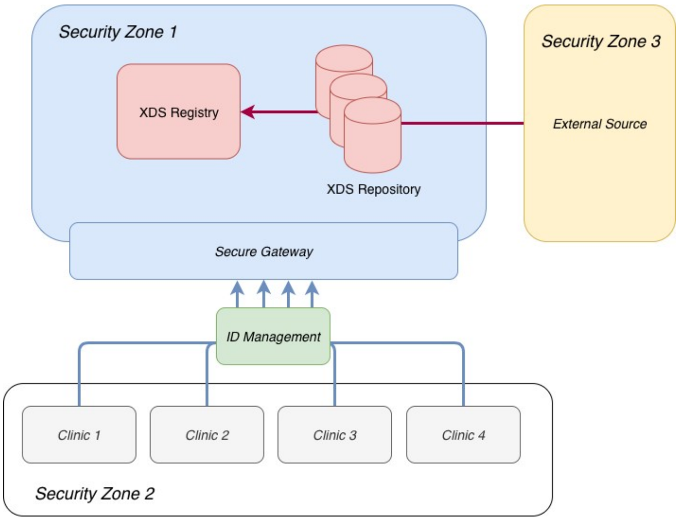


Figure 4. Initial system architecture: the central registry use case: the diagram illustrates the initial configuration of the Regional Healthcare System architecture, where the external source can access the repository without authentication, introducing a problematic connection from Security Zone 3 to Security Zone 1.

27002 definition. The project identifies three distinct security zones:

- *Zone 1*, characterized by physical security measures for machine access.
- *Zone 2*, the public internet, where clinics submit documents.
- *Zone 3*, a potentially vulnerable zone.

In the description of the XDS profile, some italicized names can be noticed that correspond to actors (e.g., *document source and registry*) and transactions (e.g., *feed and submits*). These actors and transactions play crucial roles in the XDS framework, ensuring the smooth flow of documents and information.

This hypothetical example shows how the high-level software architecture emerges from the descriptions of the IHE profile, and how the selected standards shall be profiled to achieve immediate interoperability for the particular use case.

Defining the formal language

To leverage the formal syntax from Section and demonstrate its practical utility, we have developed a comprehensive tool-chain that enables architects to build-up solutions, which can subsequently be assessed against a predefined set of metrics. These metrics may encompass various aspects, such as security goals and Service Level Agreements (SLAs).

Similar to any other software asset, architectural blueprints can require changes over time due to factors such as evolving requirements, variations in the legal landscape, or the natural progression of the project. To facilitate this dynamic process, we automate the evaluation of the impact of these modifications on the desired metrics through a Satisfiability Modulo Theories (SMT) solver. These metrics can include security- and non-security-related SLAs. In this way, we provide the architects with the capability to modify the architecture and its components and evaluate the impact of the changes.

Consider the scenario where stakeholders request cost reduction. In response, the architect can explore various strategies by exchanging different modules and identify new combinations that potentially reduce costs. For each proposed change, our solution identifies dependencies among the modules and assesses how these changes affect the metrics, in a semi-automatic fashion, ensuring compliance with the customer's specifications

Table 5. Security requirements table after applying the grouping rule in the Central Registry use case

ID	Form	Sensitivity	Location	State	Goal	Countermeasure	Category
es1	Electronic	Confidential	Partially controlled	Transmission	Authentication	Authenticate the user who is submitting the document	Data Integrity transfer protection
xds1	Electronic		Any				
satna2	Electronic	Confidential	Controlled	Transmission	Accountability	Rfc5424 syslog and DICOM audits	Accountability
satna1	Electronic	Confidential	Controlled	Transmission	Authentication	TLS channels mutually authenticated	Data confidentiality transfer protection
clinic1	Electronic	Confidential	Controlled	Transmission	Strong authentication	Clinics strongly authenticate via login screen	User authentication

TLS: Transport Layer Security.

for service level agreements.

To support this dynamic capability, we employ a formal (denotational) syntax for the IHE methodology [Table 4]. In addition, we present a method for encoding security goals (and quality attributes) in a manner that enables an automated evaluation of their fulfillment. The encoding is expressed as SMT-LIB, a standard language for formal reasoning, allowing for the efficient and systematic assessment of architectural changes against predefined metrics, thereby ensuring the ongoing alignment of the architecture with the evolving needs and objectives of the project.

After the definition of a formal syntax, we can introduce the related semantics, mapping the items of our theory to our real-world use case, described above.

Two profiles are encoded for that purpose: XDS (mentioned above) and an ATNA profile, due to the security requirements explained. Being a pure content profile (e.g., providing clinical document management), XDS only states as security_requirements the form of its data (Electronic) and where XDS documents can be disclosed (providing no constraint on the location, XDS being document agnostic). On the other hand, ATNA provides for a strict security requirement: it targets as goal *confidentiality* during transmission.

Building security policy and evaluating security requirements

While the accountability of transactions is effectively managed by the audit trails sent via ATNA, a critical piece of information is still missing: the identity of the individual who performed the action. The question arises: How can we authenticate and confirm the identity of the principal, and how can this authentication be corroborated by a third party?

ATNA currently establishes mutually authenticated channels, primarily achieved through TLS. However, there is a notable infrastructure limitation to this approach. In practice, active intermediaries must be assumed because it is neither realistic nor feasible to envision a direct connection from a device such as an X-ray machine to a regional repository. Consequently, it becomes imperative to address principal authentication at a higher ISO/OSI level, specifically at the application layer. Now, let us consider a scenario where we have an ArchitecturalElement encompassing these profiles, all of which are equipped with a mandatory grouping rule linked to ATNA. Notably, external sources rely on TLS channels for communication, while the clinics leverage robust authentication provided by Cross-Enterprise User Assertion (XUA) registry to access documents.

Upon passing this architectural element to the grouping function, the resultant output will include the profiles and the security requirements, as outlined in Table 5. This table comprises the evaluated security requirements inherited from the amalgamation of all the profiles. It serves as a comprehensive reference that can be consulted to verify whether the security policy, as defined by the RMIAS model, aligns with it.

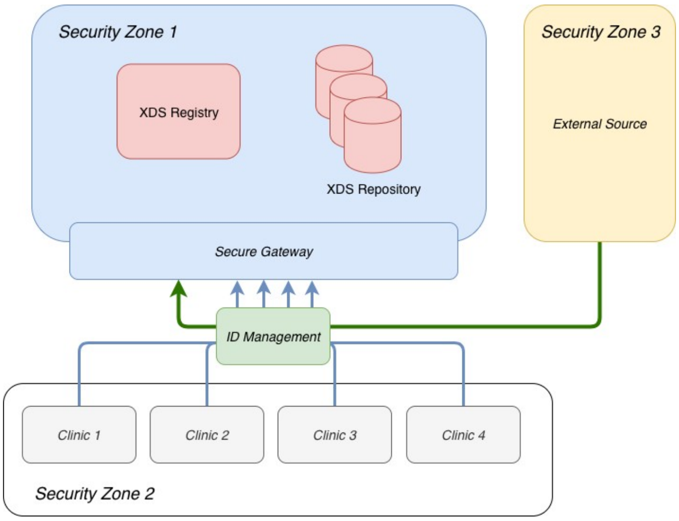


Figure 5. Case II: The Central Registry use case: this diagram illustrates the final configuration of the Regional Healthcare System architecture, after the evaluation and the fixing performed using the illustrated methodology. In this case, the external source must pass through the Identity Management Subsystem to access the Repository.

However, it is worth noting that since security policies are expressed in plain text, and given the multitude of available formalism for encoding security policies, we opt to return the table as is. This approach provides flexibility, allowing the organization or the architect to select the most suitable format for their specific needs (For instance, our implementation offers support for LUA and BeanShell as languages to query the security requirement table, ensuring that the chosen format aligns seamlessly with the preferred practices and requirements of an organization.).

In the interest of clarity in the following discussion, we introduce a trivial operator, denoted as $\sigma_{\underline{k}}$. The operator functions akin to the *selection* operation in relational algebra. In practical terms, when applied to a security requirement set denoted as S, which results from a grouping operation, $\sigma_{\underline{k}}$ returns all the values for the columns specified by $\underline{k}$. Our project encompasses a comprehensive security policy set, as outlined in Table 6. When we execute the σ operator defined in the table, we uncover a critical architectural incongruity, one that does not align with the prescribed security policy. Specifically, let us examine the security policies within the set:

- Security policy with ID 1 returns a favorable *ok* status, as it meets the condition of having at least one category for data confidentiality transfer protection.
- Security policy with ID 2 similarly returns an *ok* status because it possesses an accountability building block, as provided by *satna2*.
- However, security policy with ID 3 encounters an *error*, as there exists an entry that lacks strong authentication for confidential data, specifically health data. This policy, denoted as *es1*, permits users without strong authentication to access confidential data within a partially controlled location. This issue is illustrated in Figure 4 and is rectified in Figure 5.

The depicted scenario [Figure 6A] shows the "External Source" within Security Zone 3 (Partially Controlled) accessing health data without passing through the ID Management module. The straightforward architectural solution is to reroute the "External Source" to the ID Management component and adjust *es1*'s goal to incorporate *Strong authentication*.

In modular architectural models, including the SGAM and IHE, security profiles play a crucial role by acting as mandatory groupings for all other profiles. This practice ensures that security considerations are integrated into all phases of the system deployment lifecycle. This concept is explicitly illustrated in Figure 6B, where

Table 6. Security policy set

ID	Policy	σ Notation
1	Data confidentiality transfer protection shall be in place	$\begin{cases} \text{ok} & \text{if } \exists\ e \in \sigma_{Category}(S) : e \downarrow_{Category} = \mathit{Data\ confidentiality\ transfer\ protection} \\ error & \text{otherwise} \end{cases}$
2	Access to medical data shall be recorded	$\begin{cases} ok & \text{if } \exists e \in \sigma_{Goal}(S),\ e \downarrow Goal = \mathit{Accountability} \\ error & \text{otherwise} \end{cases}$
3	All components shall have strong authentication in all the locations, to avoid disclosure of health data	$\begin{cases} error & \text{if } \begin{array}{l} \forall e : e \in \sigma_{sensitivity,location,goal}(S),\ \exists e : e \downarrow_{goal} != \mathit{Strong\ Authentication} \wedge \\ e \downarrow_{sensitivity} = \mathit{Confidential} \end{array} \\ ok & \text{otherwise} \end{cases}$

the ATNA profile is inherited by all other profiles, thereby providing TLS transactions for all IT-based health projects within the IHE framework.

Similarly, Figure 6A illustrates the same principle, with ATNA being referenced by "clinic", "XDS", and "ExtSource" (for External Source). These examples highlight the importance of the tools that allow for a graphical visualization of the dependencies, as they improve the overall usability of the conceptual framework and, more generally, streamline the whole process. For example, looking at Figure 6A, it is easy to note that a security gap is present where "ExtSource" and "XUA" lack a direct connection. This underscores a similar concern identified by security policy *es1*, indicating that "ExtSource" lacks strong authentication since it is not grouped with "XUA". This architectural insight is essential to maintain the integrity of the system from the security point of view.

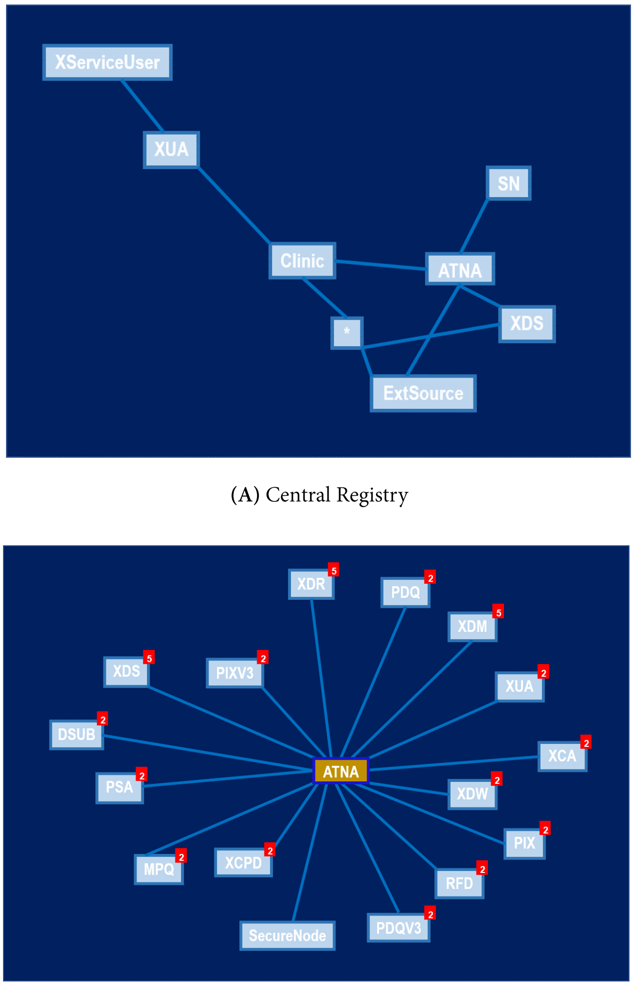
OVERVIEW OF CONTRIBUTIONS

The development of a comprehensive evaluation framework is essential not only to validate the embedded cybersecurity principles within system architecture design, but also in advancing the growing field of Critical Infrastructures Security. This, coupled with our architecture-based modeling approach that prioritizes interoperability and governance in solution design, underscores the significance and applicability of our methodology.

Moreover, the establishment of a unified system security policy, in conjunction with the formal modeling language and the integrated tool-chain, offers a holistic framework for creating systems secure-by-design, that are both repeatable and verifiable. This aspect addresses a notable gap in the broader research on CI security.

Our approach upgrades existing reference models in the information assurance and security domain in order to provide proactive means for addressing security concerns in CIs from design, and throughout the entire system lifecycle. At the same, it complies with well-established standards and frameworks while accounting for governance and interoperability of the overall solution. With this work, we also go a step further in our research on ensuring security of CIs. Notably, by introducing proactive measures through a process that is formally supported, we can devise a model that automates the evaluation of architectural security in CIs. This will lower the need for security expert intervention while aiding the designing of more resilient solution architectures.

It is important to emphasize, however, that our present approach does not encompass establishing distributed security policies. This represents an ongoing area of research, which we intend to explore in the forthcoming framework iterations. The forthcoming phase aims to facilitate modularity, not solely in policy definition but



(A) Central Registry

(B) IHE IT Technical Framework

Figure 6. Graph showing dependencies between IHE profiles (Grouping Dependencies): in (A), the diagram shows the dependencies among the profiles for the initial configuration of the *Central Registry* use case, as illustrated in (B), it is noteworthy that a security gap exists, as some profiles do not depend on XUA, while Figure 4 displays the dependencies between the profiles in the IHE IT Infrastructure Technical Framework, we can immediately detect that all the profiles depend on ATNA. IHE: Integrating the Healthcare Enterprise; XUA: cross-enterprise user assertion; IT: information technology; ATNA: Audit Trail and Node Authentication.

also in policy management and evolution. This evolution signifies a commitment to adaptability and ongoing enhancements within our approach to meet the ever-evolving challenges of securing CIs.

CONCLUSION

In this paper, we tackled the challenge of supporting the teams responsible for the design of solution architectures within the realm of CIs. We recognized the scarcity of security experts and the intermittent availability of their expertise throughout the life cycle of these infrastructures. Our approach focuses on leveraging the RMIAS, a model initially developed by the University of Cardiff and previously applied in EU Large Scale Projects, to facilitate the development of secure CIs.

To address the limited availability of security expertise, we extend RMIAS by introducing security countermeasure categories. Our approach adopts a goal-oriented perspective on security development, placing emphasis on meeting security goals rather than a solely threat-oriented approach. We provided a formal exposition of our model by introducing a modular architectural framework inspired by the IHE process, offering a syntax that is both human and machine-readable and explaining its semantics. The security requirements are evaluated against security policies formulated using our extended RMIAS approach, streamlining the formal verification of their compliance. Our solution is adaptable to various methods of expressing security policies and conducting subsequent checks, but we propose a simple, relational algebra-inspired σ notation. Moreover, the overall methodology is applicable to any critical sector, not limited to the one used in our healthcare use case.

While we believe that the proposed syntax is sufficiently clear for architects and the design team, we recognize the potential benefits of a graphical tool for representing building blocks and their dependencies, which is slated for future development. Additionally, this paper also aims to cater for a continuous update of security countermeasures[33], which may be somewhat outdated and no longer maintained. Our intent is to introduce a parametric SMRT table, using the ENISA security guidelines[36]. ENISA categorizes the technical guidelines into 25 high-level security objectives grouped into seven domains, aligning with the framework presented in this paper. We intend to delve deeper into these properties concerning the context of CIs, further classifying them within a defined scope.

DECLARATIONS

Acknowledgments
The authors acknowledge the valuable contribution of Dr. Massimiliano Masi in terms of comments and feedback on the content of the paper, as well as his suggestions on the use case provided. The authors highly appreciate his time and effort on making the paper technically solid.

Authors' contributions
Made substantial contributions to conception and design of the study: Pavleska T, Sellitto GP
Contributed to the definition of the methodology and the choice of the use case, as well as the full revision of the paper: Pavleska T, Sellitto GP, Aranha H
Defined the application of the methodology to the use case and the overall conceptual framework: Pavleska T
Performed the adjustments of the formal syntax to the use case: Aranha H
Set the scope and further revised the paper to meet the objectives: Sellitto GP

Availability of data and materials
The data model and the formal language supporting our findings can be found at: https://github.com/mascanc/MOSA2.

Financial support and sponsorship
None.

Conflicts of interest
All authors declared that there are no conflicts of interest.

Ethical approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Copyright

REFERENCES

1. Council Directive 2008/114/EC of 8 December 2008 on the identification and designation of European critical infrastructures and the assessment of the need to improve their protection (Text with EEA relevance). Available from: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv%3AOJ.L_.2008.345.01.0075.01.ENG. [Last accessed on 25 Apr 2024].
2. DIRECTIVE (EU) 2022/2557 of the European Parliament and of the Council of 14 December 2022 on the resilience of critical entities and repealing Council Directive 2008/114/EC (Text with EEA relevance). Available from: https://eur-lex.europa.eu/eli/dir/2022/2557/oj. [Last accessed on 25 Apr 2024].
3. Directive (EU) 2022/2555 of the European Parliament and of the Council of 14 December 2022 on measures for a high common level of cybersecurity across the Union, amending Regulation (EU) No 910/2014 and Directive (EU) 2018/1972, and repealing Directive (EU) 2016/1148 (NIS 2 Directive) (Text with EEA relevance). Available from: https://eur-lex.europa.eu/eli/dir/2022/2555/oj. [Last accessed on 25 Apr 2024].
4. Masi M, Pavleska T, Aranha H. Automating smart grid solution architecture design. In: 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm); 2018 Oct 29-31; Aalborg, Denmark. IEEE; p. 1-6. DOI
5. Luiijf E, Klaver M. Resilience approach to critical information infrastructures. In: Gritzalis D, Theocharidou M, Stergiopoulos G, editors. Critical infrastructure security and resilience. advanced sciences and technologies for security applications. Springer, Cham. 2019. pp. 3-16. DOI
6. Petrakos N, Kotzanikolaou P. Methodologies and strategies for critical infrastructure protection. In: Gritzalis D, Theocharidou M, Stergiopoulos G, editors. Critical infrastructure security and resilience. advanced sciences and technologies for security applications. Springer, Cham. 2019. pp. 17-33. DOI
7. Rathnayaka B, Siriwardana C, Robert D, Amaratunga D, Setunge S. Improving the resilience of critical infrastructures: evidence-based insights from a systematic literature review. *Int J Disast Risk Re* 2022;78:103123. DOI
8. Official Journal of the European Union. L 345. 23 December 2008. Available from: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=OJ%3AL%3A2008%3A345%3ATOC. [Last accessed on 25 Apr 2024].
9. Gritzalis D, Stergiopoulos G, Kotzanikolaou P, Magkos E, Lykou G. Critical infrastructure protection: a holistic methodology for Greece. In: Cuppens-Boulahia N, Lambrinoudakis C, Cuppens F, Katsikas S, editors. Security of industrial control systems and cyber-physical systems. Cham: Springer International Publishing. 2017. pp. 19–34. Available from: https://doi.org/10.1007/978-3-319-61437-3_2. [Last accessed on 25 Apr 2024].
10. Petr N, Petr R. Perspective of cross-cutting criteria as a major instrument to determination of critical infrastructure in the czech republic. Available from: https://intapi.sciendo.com/pdf/10.2478/rput-2014-0010. [Last accessed on 25 Apr 2024].
11. Franke U, Cohen M, Sigholm J. What can we learn from enterprise architecture models? An experiment comparing models and documents for capability development. *Softw Syst Model* 2018;17:695–711. DOI
12. Iacob ME, Meertens LO, Jonkers H, Quartel DAC, Nieuwenhuis LJM, van Sinderen MJ. From enterprise architecture to business models and back. *Softw Syst Model* 2014;13:1059–83. DOI
13. The Open Group. The TOGAF® Standard, Version 9.2 Overview. Available from: https://www.opengroup.org/togaf. [Last accessed on 25 Apr 2024].
14. Gerber A, le Roux P, Kearney C, van der Merwe A. The zachman framework for enterprise architecture: an explanatory is theory. In: Hattingh M, Matthee M, Smuts H, Pappas I, Dwivedi Y, Mäntymäki M, editors. Responsible Design, Implementation and Use of Information and Communication Technology. Cham: Springer International Publishing. 2020, pp. 383–96. DOI
15. HL7. Saif architecture program. 2020. Available from: https://wiki.hl7.org/SAIF_Architecture_Program. [Last accessed on 25 Apr 2024].
16. WHO Guideline. Recommendations on digital interventions for health system strengthening. 2019. Available from: https://www.who.int/publications/i/item/9789241550505. [Last accessed on 25 Apr 2024].
17. HealthIT.gov. Appendix I – Sources of security standards and security patterns. 2020. Available from: https://www.healthit.gov/isa/appendix-i-sources-security-standards-and-security-patterns. [Last accessed on 25 Apr 2024].
18. CEN-CENELEC-ETSI Smart Grid Coordination Group. SG-CG-M490/K_SGAM usage and examples. 2014. Available from: https://www.cencenelec.eu/media/CEN-CENELEC/AreasOfWork/CEN-CENELEC_Topics/Smart%20Grids%20and%20Meters/Smart%20Grids/4_sgcg_methodology_sgamusermanual.pdf. [Last accessed on 25 Apr 2024].
19. Industrie 4.0: The reference architectural model industrie 4.0 (RAMI 4.0). 2015. Available from: https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikationen/2015/april/Das_Referenzarchitekturmodell_Industrie_4.0__RAMI_4.0_/ZVEI-Industrie-40-RAMI-40-English.pdf. [Last accessed on 25 Apr 2024].
20. Gottschalk M, Uslar M, Delfs C. The use case and smart grid architecture model approach. The IEC 62559-2 use case template and the SGAM applied in various domains. Cham: Springer, 2017. Available from: https://link.springer.com/book/10.1007/978-3-319-49229-2. [Last accessed on 25 Apr 2024].
21. Galbusera L, Giannopoulos G. Leveraging network theory and stress tests to assess interdependencies in critical infrastructures. In: Gritzalis D, Theocharidou M, Stergiopoulos G, editors. Critical infrastructure security and resilience. Advanced sciences and technologies for security applications. Cham: Springer International Publishing. 2019. pp. 135–55. DOI
22. El-Rewini Z, Sadatsharan K, Selvaraj DF, Plathottam SJ, Ranganathan P. Cybersecurity challenges in vehicular communications. *Veh Commun* 2020;23:100214. DOI
23. Chattopadhyay A, Lam KY, Tavva Y. Autonomous vehicle: security by design. *IEEE T Intell Transp* 2020;22:7015-29. DOI
24. Checkoway S, McCoy D, Kantor B, et al. Comprehensive experimental analyses of automotive attack surfaces. In: Proceedings of the

20th USENIX Conference on Security. 2011. Available from: http://www.autosec.org/pubs/cars-usenixsec2011.pdf. [Last accessed on 25 Apr 2024].

25. Kim KH, Kim K, Kim HK. Stride-based threat modeling and DREAD evaluation for the distributed control system in the oil refinery. *ETRI J* 2022;44:991–1003. DOI
26. Lautenbach A, Almgren M, Olovsson T. Proposing HEAVENS 2.0 – an automotive risk assessment model. In: Proceedings of the 5th ACM Computer Science in Cars Symposium; New York, USA. Association for Computing Machinery; 2021. DOI
27. Smartgrids Austria. The initiative IES – Integrating the energy system. Available from: https://www.smartgrids.at/integrating-the-energy-system-ies.html. [Last accessed on 25 Apr 2024].
28. Gottschalk M, Franzl G, Frohner M, Pasteka R, Uslar M. From integration profiles to interoperability testing for smart energy systems at connectathon energy. *Energies* 2018;11:3375. DOI
29. Franzl G, Gottschalk M, Pasteka R. The IES cookbook - Enabling interoperability the IES way. Edition 0.8 - 21st January 2019. Available from: https://www.researchgate.net/publication/332319249_The_IES_Cookbook_-_Enabling_interoperability_the_IES_way_-_Edition_08. [Last accessed on 25 Apr 2024].
30. The Open Group. The Open Group Architectural Framework version 9.2. 2011. Available from: https://www.opengroup.org/togaf-standard-version-92-overview. [Last accessed on 25 Apr 2024].
31. Integrating the Healthcare Enterprise. IHE IT Infrastructure (ITI) technical framework. Volume 1 (ITI TF-1) Integration profiles. Available from: https://www.ihe.net/uploadedFiles/Documents/ITI/IHE_ITI_TF_Rev16-0_Vol1_FT_2019-07-12.pdf. [Last accessed on 25 Apr 2024].
32. Cherdantseva Y, Hilton J. A reference model of information assurance security. In: 2013 International Conference on Availability, Reliability and Security; 2013 Sep 02-06; Regensburg, Germany. IEEE; 2013. pp. 546–55. DOI
33. Zuccato A, Daniels N, Jampathom C, Nilson M. Report: modular safeguards to create holistic security requirement specifications for system of systems. In: Massacci F, Wallach D, Zannone N, editors. Engineering secure software and systems. Springer: Berlin; 2010. pp. 218–30. DOI
34. CEN-CENELEC-ETSI Smart Grid Coordination Group. Smart grid reference architecture. 2012. Available from: https://www.cencenelec.eu/media/CEN-CENELEC/AreasOfWork/CEN-CENELEC_Topics/Smart%20Grids%20and%20Meters/Smart%20Grids/reference_architecture_smartgrids.pdf. [Last accessed on 25 Apr 2024].
35. Masi M, Sellitto GP, Aranha H, Pavleska T. Securing critical infrastructures with a cybersecurity digital twin. *Softw Syst Model* 2023;22:689–707. DOI
36. Dekker M, Karsberg C. Technical guideline on security measures. 2014. Available from: https://resilience.enisa.europa.eu/article-13/guideline-for-minimum-security-measures/Article_13a_ENISA_Technical_Guideline_On_Security_Measures_v2_0.pdf. [Last accessed on 25 Apr 2024].