ZEROSWARM

ZERO-enabling Smart networked control framework for Agile cyber physical production systems of systems

D2.3 - Cyber-security implementation templates and methodological approach (Revised)

Project funded by Horizon Europe, Grant Agreement #101057083

Topic	HORIZON-CL4-2021-TWIN-TRANSITION-01-08
Project Title	ZERO-enabling Smart networked control framework for Agile cyber physical production systems of systems
Project Number	101057083
Project Acronym	Zero-SWARM
Deliverable No/Title	D2.3 Cyber-security implementation templates and methodological approach
Contractual Delivery Date	M10
Actual Delivery Date	M17 (revision 1)
Contributing WP	WP2 – Requirements, System Design & Architecture
Project Start Date, Duration	01/06/2022, 30 Months
Dissemination Level	Public
Nature of Deliverable	R

Document History			
Date	***Version***	***Author***	***Description***
31.03.2023	*1.0*	*CERTH*	*Submission of version 1.0*
08.05.2023	*1.1*	*CERTH*	*Reorganization & extension of ToC*
22.05.2023	*1.2*	*All*	*Approval of new ToC & distribution of work*
19.06.2023	*1.3*	*NX-SE & all*	*Contribution to the SoTA & common practices*
25.07.2023	*1.4*	*All*	*Contribution to all missing sections*
28.08.2023	*1.5*	*S21Sec & CERTH*	*Contribution to Zero-SWARM reference cybersecurity architecture*
19.09.2023	*1.6*	*CERTH*	*Internal review by UMH, AIM, HWE, CERTH*
02.10.2023	*1.7*	*All*	*Internal review comments addressing*
12.10.2023	*1.8*	*CERTH*	*Final draft to be checked by Coordinator & Technical Manager*
16.10.2023	*2.0*	*CERTH*	*Final submission of revision 1*

Authors List

Leading Author (Editor)				
Surname		***Initials***	***Beneficiary Name***	***Contact email***
Mpatziakas		*AM*	*CERTH*	ampatziakas@iti.gr
Co-authors (in alphabetic order)				
#	*Surname*	*Initials*	*Beneficiary Name*	*Contact email*
1	*Borne*	*RB*	*S21Sec*	rborne@s21sec.com
2	*Egaña*	*JE*	*S21Sec*	jegana@s21sec.com
3	*Hatzidiamantis*	*NH*	*CERTH*	*hatzidiamantis@iti.gr*
4	*Lazaridis*	*GL*	*CERTH*	glazaridis@iti.gr
5	*López*	*OL*	*S21Sec*	olopez@s21sec.com
6	*Drosou*	*AD*	*CERTH*	*drosou@iti.gr*
7	*Ros*	*SR*	*S21Sec*	*sros@s21sec.com*

Contributors (in alphabetic order)				
#	***Surname***	***Initials***	***Beneficiary Name***	***Contact email***
1	*Fernandez*	*RF*	*AIM*	roberto.fernandez@aimen.es
2	*Fritz*	*AF*	*NX-SE*	artur.fritz@se.com
3	*Kot*	*KK*	*IDSA*	Kateryna.Kot@internationaldataspaces.org
4	*Trakić*	*ST*	*NX-SE*	Sejla.Trakic@se.com

Reviewers List

List of Reviewers (in alphabetic order)				
#	***Surname***	***Initials***	***Beneficiary Name***	***Contact email***
1	*Barja*	*LB*	*AIM*	*lara.barja@aimen.es*
2	*khodashenas*	*PK*	*HWE*	*pouria.khodashenas@huawei.com*
3	*Sepulcre*	*MS*	*UMH*	*msepulcre@umh.es*

DISCLAIMER OF WARRANTIES

This document has been prepared by Zero-SWARM project partners as an account of work carried out within the framework of the contract no 101057083.

Neither Project Coordinator, nor any signatory party of Zero-SWARM Project Consortium Agreement, nor any person acting on behalf of any of them:

- makes any warranty or representation whatsoever, express or implied,
    - with respect to the use of any information, apparatus, method, process, or similar item disclosed in this document, including merchantability and fitness for a particular purpose, or
    - that such use does not infringe on or interfere with privately owned rights, including any party's intellectual property, or
- that this document is suitable to any particular user's circumstance; or
- assumes responsibility for any damages or other liability whatsoever (including any consequential damages, even if Project Coordinator or any representative of a signatory party of the Zero-SWARM Project Consortium Agreement, has been advised of the possibility of such damages) resulting from your selection or use of this document or any information, apparatus, method, process, or similar item disclosed in this document.

Zero-SWARM has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101057083. The content of this deliverable does not reflect the official opinion of the European Union. Responsibility for the information and views expressed in the deliverable lies entirely with the author(s).

Table of Contents

List of Abbreviations

Abbreviation	Description
API	Application Programming Interface
AUSF	Authentication Server Function
BSS	Business Support System
CaC	Compliance-as-Code
CAT	Composite Automation Type
CD	Continuous Delivery
CI	Continuous Integration
CPS	Cyber Physical System
CPSoS	Cyber Physical Systems of Systems
CT	Continuous Testing
DAST	Dynamic Application Security Testing
DC	Data Confidentiality
DevOps	Development & Operations
DevSecOps	Development, Security & Operations
DIN	Deutsches Institut für Normung
DoS	Denial of Service
DPMS	Data Protection Management System
EAP	Extensible Authentication Protocol
EDR	Endpoint detection and Response
ENISA	European Union Agency for Cybersecurity
ERP	Enterprise Resource Planning
FB	Function Block
FG	Functionality Group
FR	Foundation Requirements
GDPR	General Data Protection Regulation
HMI	Human Machine Interface
HTTP	Hypertext Transfer Protocol
HW	Hardware
IaC	Infrastructure-as-code
IAC	Identification & Authentication Control
IACS	Industrial Automation & Control Systems
IAM	Identity and Access Management
IDE	Integrated Development Environment
IDS	Intrusion Detection System
IEC	International Electrotechnical Commission
IIOT	Industrial Internet of Things
IIRA	Industrial Internet Reference Architecture
IOT	Internet of Things
IPS	Intrusion Prevention System
ISA	International Society of Automation
IT	Information Technology
KPI	Key Performance Indicator
MES	Manufacturing Execution System
NIST	National Institute of Standards and Technology

MQTT	Message Queuing Telemetry Transport
NRA	Network Resource Availability
OPC-UA	Open Platform Communications Unified Architecture
OT	Operational Technology
PASTA	Process for Attack Simulation and Threat Analysis
PERA	Purdue Enterprise Reference Architecture
PKI	Public Key Infrastructure
PLC	Programmable Logic Controller
PoLP	Principle of Least Privilege
QA	Quality Assurance
RA	Resource Availability
RA	Reference Architecture
RAMI	Reference Architectural Model Industry
RDF	Restricted Data Flow
RT	Run Time
RTU	Remote Terminal Unit
SaC	Security-as-Code
SEAF	Security Anchor Function
SAST	Static Application Software Testing
SBA	Service-based architecture
SbD	Security-by-Design
SCA	Static Composite Analysis
SCADA	Supervisory Control And Data Acquisition
SI	System Integrity
SIDF	Subscription Identifier De-concealing Function
SIEM	Security information and event management
SMM	Security Maturity Model
SOAR	Security Orchestration, Automation and Response
SoTA	State-of-the-art
SSH	Secure Shell
SSL	Secure Sockets Layer
SUCI	Subscription Concealed Identifier
SW	Software
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TRE	Timely Response to Event
TTL	Time-to-live
Tx.y	Task x.y
UC	User Control
UDM	Unified Data Management
UE	User Equipment
UP	User Plane
VPN	Virtual Private Network
WP	Work Package

List of Figures

List of Tables

Executive Summary

This document namely Zero-SWARM Deliverable D2.3 “Cyber-security implementation templates and methodological approach”, offers a set of methodological guidelines that are necessary for engineering cyber-secure Cyber Physical Systems of Systems (CPSoS) with a security-by-design approach. An initial version of D2.3 was submitted on time (M10 – March 2023), and the current revised version is submitted in (M17 – October 2023) to address various comments made by the project reviewers.

Initially, this report presents a brief introduction to the subject, some primary information regarding the purpose of writing the document, the relationship with other deliverables of the Zero-SWARM project, which have been already submitted and an overview of the document’s structure. To reach the objective of this task and establish a collection of security templates and a reference cybersecurity architecture for the project, specific actions were undertaken.

To begin with, a state-of-the-art analysis regarding cybersecurity in the Industrial domain along with the various technologies that will enable the Zero-SWARM project such as 5G and Federated Learning. Then, the DevOps and DevSecOps methodologies are thoroughly explained, along with specifics of utilizing the DevSecOps methodology in the scope of the Zero-SWARM project.

We proceed with an analysis and study of the project’s requirements from Task T2.1 and the architecture design from Task T2.2 and a presentation of the modules related to cybersecurity that will be used in the project, either existing or newly developed along with an analysis of the security-by-design addressed by them. ISO/IEC TS 19249 offers Security-by-Design Principles that are high-level recommendations. To help with the design of the modules developed inside a project, the deliverable provides a mapping of to these principles to 58 principles and recommendations of other standards and whitepapers that provide a finer level of granularity.

Then, the Zero-SWARM Cybersecurity templates based on IEC-62443, are presented. The templates contain clearly defined, specific requirements that if satisfy allow the system to achieve specific predefined security levels: These will be the guidelines throughout the project’s lifetime, for the development of the different Zero-SWARM trials based on a secure-by-design manner.

Additionally, the Zero-SWARM cybersecurity templates and modules are mapped a) to the defence in depth approach introduced in IEC 62443-1-1 and b) the ISO/IEC TR 19249 cybersecurity architectural and design principles. Finally, based on the rest of the document, the Zero-SWARM reference is presented based on the architectural views proposed on D2.2 and relevant standards.

1. Introduction

Industry and Operational Technology (OT) environment have been traditionally isolated from the Information Technology (IT) world; however, the industries are evolving by connecting their infrastructures to IT technologies with the aim of boosting their potential. This leads to an accelerated interconnection of elements that have not been designed with robust security aspects along with the exposure of the production and manufacturing processes to the IT and the Internet world, exposing the industrial domain to several threats and risks.

Extending this to the scope of T2.3, this exposure also applies to Cyber Physical Systems (CPSs), Cyber Physical Systems of Systems (CPSoS) and Digital Twin (DT) environments, as part of the OT network. More specifically, CPSs are automated systems, which bridge the gap between the physical world and computing and communication infrastructures, facilitating the integration and synchronization of their operations [1]. On the other hand, CPSoS refer to large-scale interconnected systems that integrate physical elements with distributed IT systems, communication networks and human operators. CPSoS typically consist of complex physical systems like transportation networks, power grids or industrial plants, where multiple physical components interact with each other. These physical components are closely coupled with distributed IT systems responsible for monitoring, control, optimization, and human interaction. The IT systems and physical elements are interconnected through communication networks, forming a comprehensive networked infrastructure that enables coordinated operation and management of the overall system [2].

The term “Digital Twins” (DT) refers to virtual replicas or digital representations of physical objects, systems, or processes. DT are created by collecting and integrating real-time data from sensors, devices, or other sources to simulate the behaviour, characteristics and performance of their physical counterparts. DT provide a means to monitor, analyse and optimize the physical entity throughout its lifecycle by leveraging advanced technologies such as Internet of Things (IoT), Artificial Intelligence (AI) or data analytics. They give the opportunity to organizations to gain insights, make informed decisions and perform predictive and prescriptive analysis in various domains, including manufacturing, healthcare, smart cities, etc. The DT concept aims to bring together the physical and digital worlds, facilitating better understanding, control and optimization of complex systems and assets [3].

Even though CPSoS and DT play a vital role in transforming traditional manufacturing processes into more intelligent, connected, and efficient operations, they also introduce cybersecurity risks. With the integration of digital technologies and connectivity, the attack surface, i.e., the interfaces and services that can be used as a basis for an attack, in manufacturing environments expands. More entry points become available for potential cyber threats, including unauthorized access to physical systems, manipulation of data or disruption of operations. There are risks associated with CPSs, which a system designer needs to be aware of, such as unauthorized access, zero-days attacks and other vulnerabilities, e.g., those described in the MITRE ATT&ck table [4], all of which are potential hazards which can damage or destroy these systems.

Moreover, CPSoS rely on the interconnection of various systems and devices, which can introduce vulnerabilities. An exploit in one system could potentially propagate to other interconnected systems, causing widespread disruptions or compromise. Regarding the DT, the collection of real-time data raises concerns about data ownership, privacy, and protection. Manufacturers need to ensure proper data encryption, access controls, and secure storage to protect sensitive information and intellectual property from unauthorized access or theft. Another representative example of a cybersecurity risk includes malicious actors that may target manufacturing systems with malware or ransomware, disrupting operations, encrypting critical data, or demanding ransom payments for its release. The potential impact could range from financial losses to production downtime. Finally, manufacturing systems often have long lifecycles, making it challenging to apply timely security updates and patches. Outdated software or firmware in CPS or CPSoS components may contain known vulnerabilities that can be exploited by attackers.

Considering the above-mentioned risks, the need for adopting security-by-design principles as well as industrial security standards has raised, to include cybersecurity considerations into the OT world. The term security-by-design describes an approach that tries to design systems and products that having enough inherent security traits so that they can reasonably defend against malicious actors successfully gaining access to devices, data, and connected infrastructure.

Some standards, such as the NIST-800-53 [9] take a broader approach in defining desired outcomes where others like IEC-62443 [5] dive deeper into the details of how to deploy security and if security is ideal. On the other hand, ISA-99 [8] and IEC-62443 are two standards that need to be considered. ISA/IEC 62443 is also identified in the NIST-800 framework as an informative reference. Industry has referenced IEC-62443 in the development of the security features for the sector, and thus, the Zero-SWARM project is focusing on it as a basis for defining the cybersecurity requirements and templates. The approach proposed by IEC-62443 is enhanced by considering multiple recommendations and approaches from other related standards such as ISO/IEC TR 19249.

Considering the security-by-design principles, will help ensure that security is a fundamental aspect of the design, development, implementation, and operation of the industrial systems demonstrated by the project.

Additionally, the project also proposes a DevSecOps approach as a cybersecurity methodology. This has been chosen because DevOps is a well-known industry standard for software development in a continuous, fluid, and agile way. The DevSecOps approach is an evolution from DevOps to include security concerns and controls in all the phases of the software (SW) development cycle, so that cybersecurity can be considered and included by design. This approach has been selected to be also in line with other project tasks, such as “T4.4 - Federated transparent, flexible, and trustable data infrastructure and DevOps tools for continuous data-driven models” and “T5.4 - Ad-Hoc penetration and hypothesis testing plugins”, dealing with DevOps and Continuous Integration and Continuous Delivery (CI/CD) collaborative environments and security testing tools for vulnerability detection. The appliance of this approach is not strictly connected to only these aforementioned tasks, but could also be applied by other technical tasks of the Zero-SWARM project.

1.1. Purpose of the document

This deliverable provides the appropriate templates to be filled in by the relevant consortium partners regarding the technical specifications and design of their cybersecurity related implementation. This document will also present the defined methodological approach be followed in such cases along with a reference cybersecurity architecture proposed by the Zero-SWARM project.

1.2. Relationship with other deliverables

As written bellow, D2.3 receives input from D2.1 and D2.2 but outputs the cybersecurity implementation templates and the methodological approach to be used by all technical Work Packages (WP) of the Zero-SWARM project in the development phase of the components. Some indicative deliverables, where this information will be used extensively are the below presented documents, but not limited to these:

- **Input:** D2.1 Definition & analysis of trials, KPIs &GDPR compliance (Task T2.1)
- **Input:** D2.2 Eco designed architecture, specifications & benchmarking (Task T2.2)
- **Output:** D4.4 Federated data infra & toolkit for data-driven model v1 (Task T4.4)
- **Output:** D4.8 Federated data infra & toolkit for data-driven model v2 (Task T4.4)
- **Output**: D5.4 Penetration and hypothesis testing diagnostic plugins v1 (Task T5.4)
- **Output**: D5.5 Anomaly detection and countermeasure selection tools v1 (Task T5.5)
- **Output**: D5.9 Penetration and hypothesis testing diagnostic plugins v2 (Task T5.4)
- **Output**: D5.10 Anomaly detection and countermeasure selection tools v2 (Task T5.5)

1.3. Rationale behind the structure

This section describes the structure of the document. The first two sections are introductory: Section 1 provides an introduction and a general description of the document along with other important content such as the actions performed to enhance this revised version of the deliverable. Section 2 introduces the state-of-the-art (SoTA) with the research on the security aspects and requirements of the technologies present in the project along with common cybersecurity practices in the industry. Section 3 describes the fundamentals of DevOps methodology as the base for the DevSecOps methodology evolution and includes a Secure DevOps approach to Cyber Physical systems along with an inspection of where the why DevSecOps is suitable and applicable to the Zero-SWARM project. Based on the SoTA work presented in section 3, Section 4 introduces a number of base cybersecurity requirements for various technologies and approaches, both existing and developed within the project that will be used in the projects along with the security-by-design aspects covered by them. Section 6 defines a set of security templates for the project, based on IEC 624443, aimed to be considered by project partners in the implementation phase of the project. Based on the work presented in sections 1 to 5, section 0 presents the reference architecture that will be utilized in the Zero-SWARM project along with a mapping of the requirements of the Zero-SWARM cybersecurity templates to the security-by-defence aspects defined in in IEC 623443. Finally, Section 7 summarizes the conclusions of the document.

1.4. Actions performed to address Reviewer recommendations

The following section contains the Reviewer recommendations to enhance the overall quality of deliverable D2.3 along with the actions undertaken to address these recommendations.

Recommendation 1: *"Improve deliverable D2.3 by defining the Zero-SWARM cybersecurity architecture according to best practices (see in particular the IEC 62443 series more deeply regarding Defense-in-Depth and the ISO/IEC TS 19249:2017 "Information technology – Security techniques – Catalogue of architectural and design principles for secure products, systems and applications")".*

Actions to address recommendation:

Section 2 containing the SoA and common approaches has been reworked to become more oriented towards the aim of defining the projects' architecture and some new material was introduced: Section 2 was expanded to focus on ENISA good practises for I4.0 and new sections were added to cover cybersecurity in Federated Learning (2.6) and Security-by-Design Approaches (2.7.1). More importantly, a new section, section 0 named "Zero-SWARM reference cybersecurity architecture", was introduced. In this section, initially a description to assess the Zero-SWARM cybersecurity template based on IEC-62443 is provided. Then the projects' reference cybersecurity architecture is provided. Finally in subsection 6.2, we cover the security-by-design aspects of the reference architecture and its' underlying approach by clearly mapping the modules and approaches used in the project a) to the IEC 62443 Défense-in-Depth layers and b) to the design and architectural principles defined by the ISO/IEC TS 19249:2017 standard.

Recommendation 2: *"[…] when revising deliverable D2.3, clearly separate or mark what is background and what is foreground […]."*

Actions to address recommendation: The revised version of D2.3 clearly separates the foreground content described in the deliverable, via declaring the foreground related parts in the name of each respective section e.g., section 7

Zero-SWARM reference cybersecurity architecture . Additionally, the original material has been expanded to better describe the cybersecurity functionalities offered by the project in sections 4 and 0.

2. State of The Art & Common Practices

The SoTA section is oriented to research on the security aspects and requirements of the technologies present in the project. The innovations promised by the Zero-SWARM involve the use of multiple technologies and approaches from different knowledge domains including but not limited to Industry Security Standards, the 5-layer industrial communications architecture, IIoT (Industrial IoT) References, 5G Architectures and CPSoS.

The aim of the deliverable is to initially review the current knowledge concerning cybersecurity related approaches on the technologies through the analysis of related published work. The focus, however, is the cybersecurity aspects and the reference architectures, some topics of which are shown below:

- **Threat Modelling & Risk Assessment**: Conduct a thorough threat modelling exercise to identify potential security risks and vulnerabilities specific to the domain or industry. Perform a comprehensive risk assessment to understand the impact and likelihood of these risks.
- **Security by Design**: Incorporate security principles and best practices into the design of the reference architecture. This includes considering security controls, secure configurations and robust authentication and authorization mechanisms from the start.
- **Data Protection & Privacy**: Ensure that reference architectures include measures to protect sensitive data and uphold privacy requirements. Employ encryption, access controls, data anonymization and compliance with relevant data protection regulations.
- **Secure Communication & Network Architecture**: Consider secure communication protocols, such as Transport Layer Security (TLS), for transmitting data within the architecture. Implement network segmentation, firewalls and intrusion detection/prevention systems to protect against unauthorized access and network-based attacks.
- **Identity & Access Management**: Incorporate strong identity and access management (IAM) practices to control user access, manage privileges and authenticate users within the reference architecture. Implement multi-factor authentication and least privilege principles to minimize the attack surface.
- **Threat Detection & Response**: Include mechanisms for detecting and responding to security incidents within the reference architecture. This may involve the use of security information and event management (SIEM) systems, log monitoring, and real-time threat intelligence feeds.
- **Security Governance & Compliance**: Establish security governance processes and frameworks to ensure ongoing compliance with relevant security standards and regulations. This includes conducting regular security audits, implementing security policies and procedures and training personnel on security awareness.
- **Vendor & Supply Chain Security**: Address security considerations related to third-party vendors and supply chain partners. Perform due diligence when selecting vendors and ensure they adhere to robust security practices. Establish contractual agreements, which include security requirements and periodic security assessments.

- **Security Testing & Validation**: Regularly test and validate the security of the reference architecture through activities such as penetration testing, vulnerability scanning and security code reviews. Conduct security assessments during the architecture's development lifecycle and after any significant changes or updates.
- **Continuous Monitoring & Improvement**: Implement continuous monitoring of the reference architecture's security posture. Regularly assess and update security controls, apply patches and updates as well as stay informed about emerging security threats and vulnerabilities within the industry.

2.1. Industry Security Standards

2.1.1. IEC 62443

IEC-62443 [5] is the main Cybersecurity standard chosen to be applied to check the security level of the Zero-SWARM solutions. We must take note that IT and OT have different perspective in Cybersecurity, as summarized in Figure 1. For that reason, the needed OT cybersecurity standard has been developed and in D2.3 we focus in the most deployed option: IEC-62443 [5]. This standard is described in multiple documents: a list is presented in Appendix A.

Business Priority	Confidentiality	Availability
Major Focus	Data integrity is key	Control processes cannot tolerate downtime
Protection Targets	Windows computers, servers	Industrial legacy devices: PLC, HMI, meters
Environmental Conditions	Air-conditioned	Harsh environments: extreme temperatures, vibrations & shocks

Figure 1. IT and OT cybersecurity perspective [82]

Figure 2 illustrates some of the most common standards available in the market. Some standards like NIST-800-53 [9] take a broader approach by defining desired outcomes while others like IEC-62443 [5] dive deeper into the details of how to deploy security and how much security is ideal. Industry has referenced IEC-62443 in the development of the security features for the sector. ISA-99 [8] and IEC-

62443 are essentially the same. ISA/IEC 62443 is also identified in the NIST-800 framework as an informative reference.

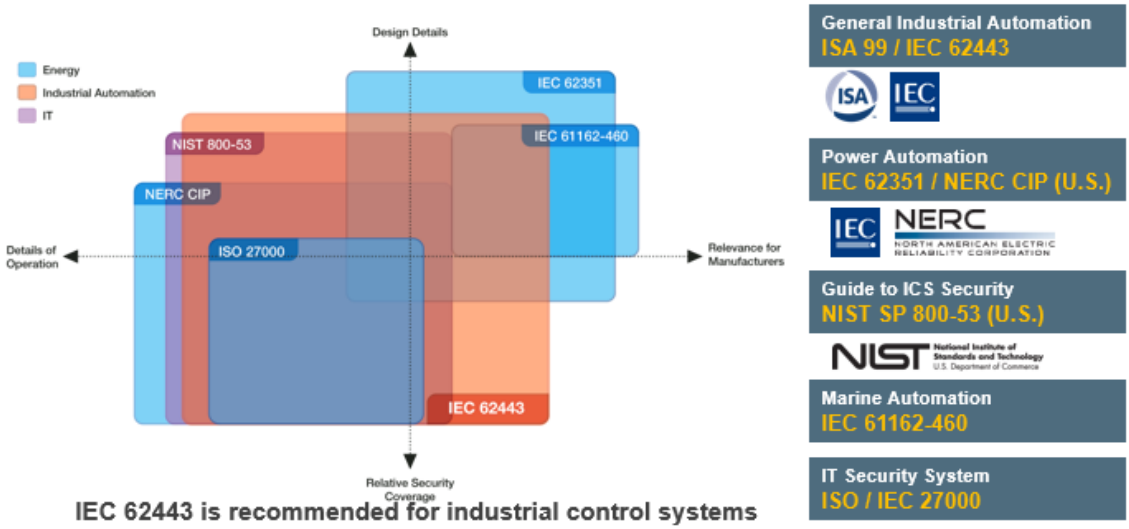


Figure 2. Common Cybersecurity Standards for Industry [80]

An automation system can be viewed as a series of level, from individual components or devices to systems and more complex systems of systems. Although it is difficult to accommodate all these perspectives in a single structure, the series description that is most commonly used is shown in Figure 3. A Complete list of the IEC 62433 with their description is available in Appendix A.

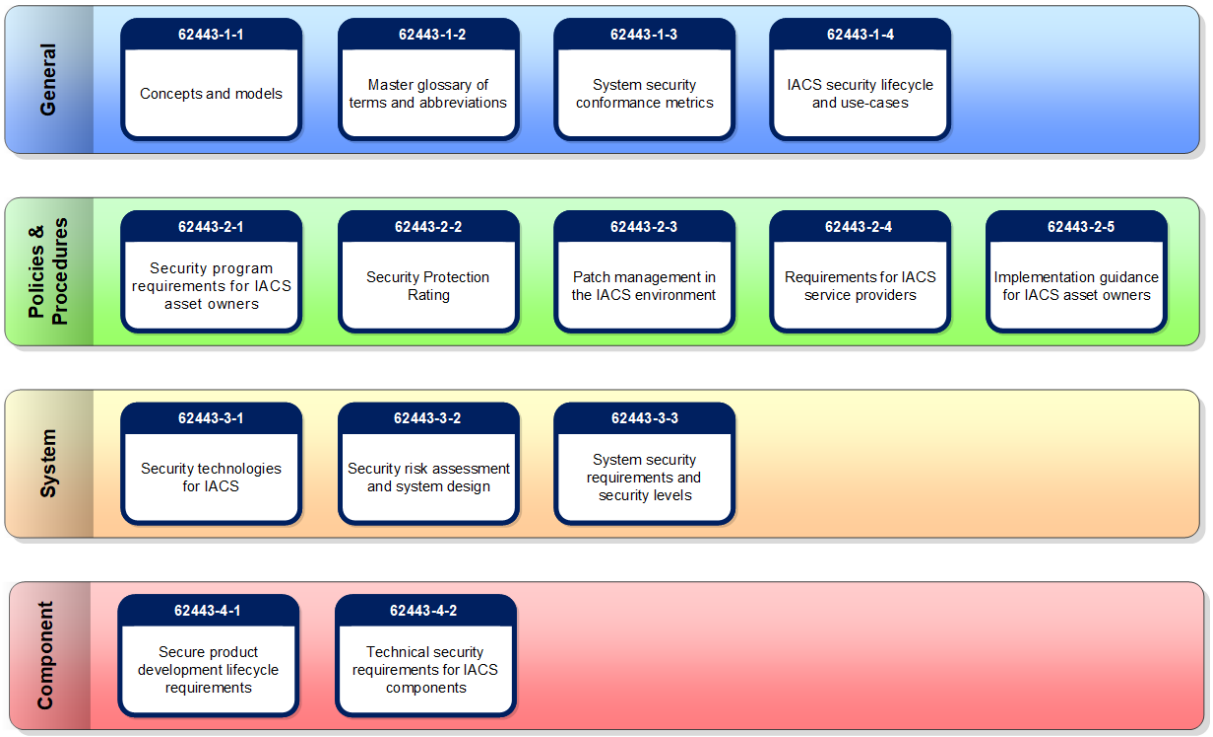


Figure 3. IEC-62443 series

In the text below, the seven (7) Foundation Requirements are defined, as mentioned above.

1. **Identification and Authentication Control (IAC):** Reliably identification and authentication of all users (humans, software processes and devices) attempting to access the IACS.

Obstacles:

- Lack of Identification and Authentication Control: Everyone can access important assets.
- Lack of Use Control: Unauthorized people can do what they are not allowed to do.

Solutions:

- Account and Password management: It allows administrators to control user access to and from IT resources based on different access levels.
- Password Policy: A set of rules designed to enhance computer security by employing strong passwords.
- Account Lockout and Logout: It allows administrators to specify the number of unsuccessful login attempts that can be made before the account is disabled.

2. **User Control (UC):** It enforces the assigned privileges of an authenticated user (human, software process or device) to perform the requested action on the system or assets and monitor the use of these privileges.

Obstacles:

- Lack of activity log: Not possible to track who and when accessed the network.
- Lack of network monitoring: When having disruption, it is not generating any alarms and not possible to identify the origin of the issue.

Solutions:

- Event Log: It is a basic "logbook" that stores records of events from various sources in a standard and centralized way.
- Syslog: Syslog is a standard for message logging. Each message is labelled with a facility code, indicating the software type generating the message, and assigned a severity level. When operating over a network, syslog uses a client-server architecture where a syslog server listens for and logs messages coming from clients.

3. **System Integrity (SI):** Ensure the integrity of the IACS to prevent unauthorized manipulation.

Obstacles:

- Lack of Data Integrity: Devices didn't check the configuration file or firmware.

Solutions:

- System File Encryption: File encryption protects individual files or file systems by encrypting them with a specific key, making them accessible only to the keyholder. Full disk encryption, on the other hand, secures an entire disk or drive but doesn't encrypt individual files within the disk.
- Secure Boot: Secure boot is designed to protect a system against malicious code being loaded and executed early in the boot process, before the operating system has been loaded. This is to prevent malicious software from installing a "bootkit" and maintaining control over a computer to mask its presence.

4. **Data Confidentiality (DC):** It ensures the confidentiality of information on communication channels and in data repositories to prevent unauthorized disclosure.

Obstacles:

- Lack of Encryption: Data is transmitted as plain text and can be manipulated.

Solutions:

- Secure Sockets Layer (SSL_ certificate management: SSL certificate management is the process of monitoring and managing the life cycles—from acquisition and deployment to tracking renewal, usage, and expiration—of all SSL certificates deployed within a network. This process provides IT administrators with complete visibility and control over their SSL environments and helps them pre-empt security breaches, outages, and compliance issues.

5. **Restricted Data Flow (RDF):** Limit or control the amount of data that is transmitted through communication channels. User A in Area A can access the data in Area B which is only allowed to access by a User B.

Obstacles:

- Lack of Restricted Data Flow: Unauthorized user access from area A to area B without any protection.

Solutions:

Access Control List: usage of a list of rules that specifies which users or systems are granted or denied access to a particular object or system resource.

Accessible IP address list: usage of a list of IP addresses that specifies which Ips are allowed to connect to a particular resource.

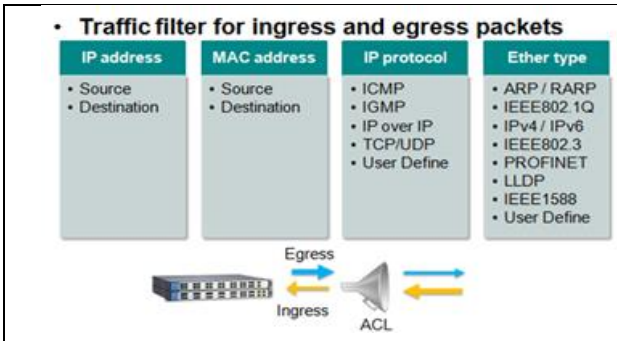


Figure 4. ACL solution for RDF FR

Figure 5. Accessible IP address list solution for RDF FR

6. Timely Response to Event (TRE): Long recover time due to no auditable log function.

Obstacles:

- Lack of Timely Response Event: Process might shutdown due to security violations, such as important settings are wrongly changed. How to find root cause faster?

Solutions:

- Enable system event log to trace users conducting configuration change.

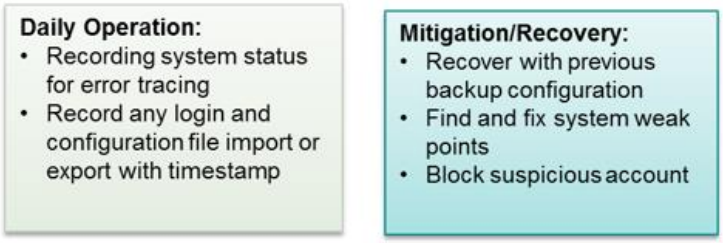


Figure 6. Timely Response to Event (TRE) mitigation

7. Network Resource Availability (NRA): System resource degradation due to Denial-of-Service (DoS) attack.

Obstacles:

- Lack of network resource availability: A system resource might be unavailable because of DoS attack if a limit is not imposed.

Solutions:

- Disable unencrypted or unused interfaces (e.g., HTTP, Telnet): It limits the maximum login users to prevent device overload with superfluous requests, as shown in Figure 7.

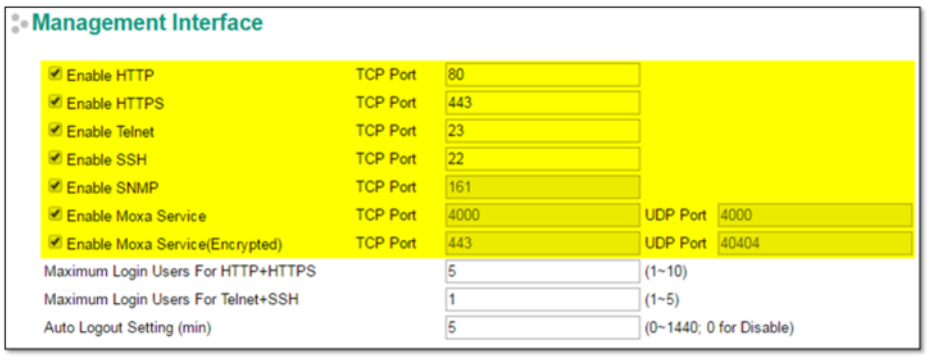


Figure 7. Disable unencrypted / unused interface solution for NRA FR

Table 1 summarizes the Foundational Requirements of IEC 62443.

Table 1. Summary of Foundational requirements [41]

Foundational Requirement	Associated Process
FR1 – Identification, Authentication, and Access Control	User authentication and Authorization
FR2 -Use Control	Enforcement of roles and responsibilities
FR3 – System Integrity	Change management
FR4 – Data Confidentiality	Use of Encryption

FR5 – Restrict Data Flow	Network segmentation
FR6 - Timely Response to Event	Audit logs
FR7 – Resource Availability	System backup and recovery

2.1.2. DIN SPEC 27070: 2020-3

Another standard inspired from IEC-62443 is the DIN-27070 [7], that comes from the German Institute for Standardization. In this case, the DIN-27070 "Requirements and reference architecture of a security gateway for exchange of industry data and services" specifies the requirements for establishing virtualised roots of trust in the scope of the exchange of industrial data. This standard is part of the ISO/DIN-27000 related to IT security techniques.

DIN SPEC 27070: 2020-3 is a new data protection standard developed by the German Institute for Standardization, also known in Germany as the Deutsches Institut für Normung (DIN). It is designed to help organizations implement effective data protection measures and comply with data protection regulations, such as the European Union's General Data Protection Regulation (GDPR). The standard provides guidelines for establishing a Data Protection Management System (DPMS), which includes policies, procedures, and controls for managing personal data. It covers a wide range of topics related to data protection, including data classification, risk management, incident management, and training and awareness.

One of the key features of DIN SPEC 27070: 2020-3 is its focus on risk-based data protection. This means that organizations are encouraged to assess the risks associated with their processing activities and implement measures to mitigate those risks. The standard provides guidance on how to conduct risk assessments and how to identify appropriate measures to address identified risks. Another important aspect of the standard is its emphasis on the importance of transparency and accountability in data protection. Organizations are required to be transparent about their data processing activities, including the purposes for which data is collected, the types of data collected, and the legal basis for processing the data. They must also be accountable for their data protection measures and demonstrate compliance with applicable regulations. DIN SPEC 27070: 2020-3 is designed to be flexible and adaptable to different types of organizations and data processing activities. It can be used by organizations of all sizes and in all industries, and can be customized to meet specific needs and requirements.

Overall, DIN SPEC 27070: 2020-3 is a comprehensive and practical standard for data protection management. It provides organizations with a framework for implementing effective data protection measures and complying with data protection regulations. By following the guidelines set out in the standard, organizations can enhance their data protection practices and build trust with their customers and stakeholders.

2.2. 5-layer security architecture

Purdue model was adopted from the Purdue Enterprise Reference Architecture (PERA) model by ISA-99 and used as a concept model for ICS network segmentation. It is an industry adopted reference

model that shows the interconnections and interdependencies of all the main components of a typical ICS [12]. Industrial communications are commonly organized in 5 levels [12] , where each one has their own protocols, devices and specifications. This is shown in Figure 8.

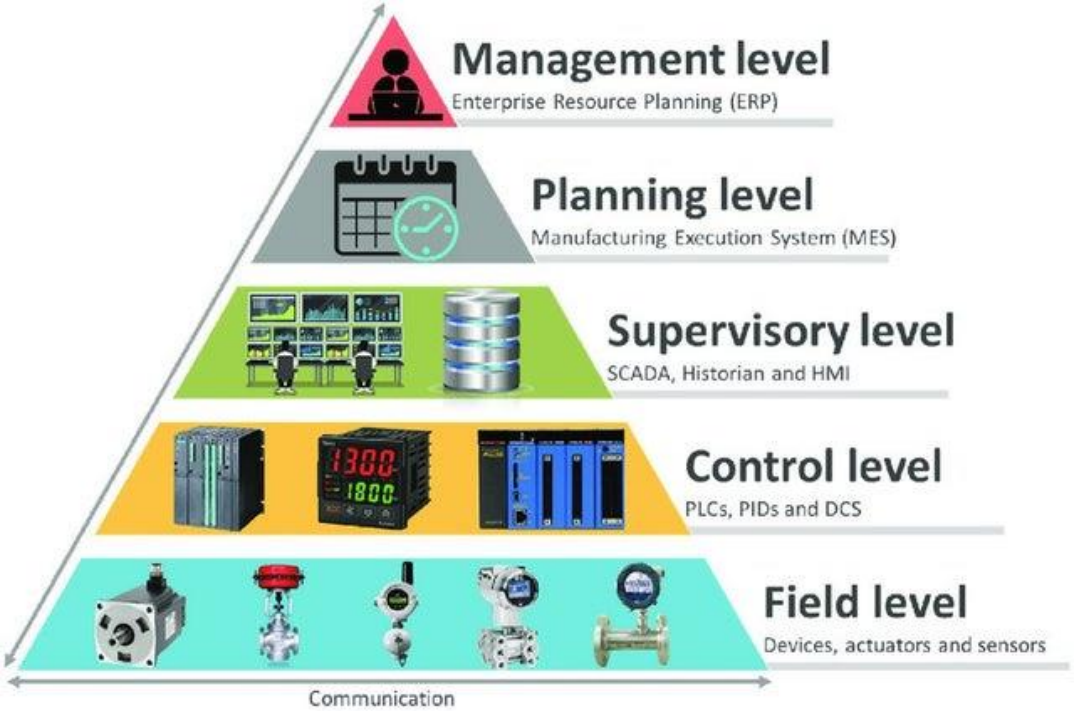


Figure 8. Industrial communications 5-level architecture

1. Field Level: Is the part of the industrial process where there are the sensor and actuators. This part is most close to the process. The typical signals a Time-to-live (TTL) (1-0, on-off, all-nothing, etc.) or analogue signals (temperature, pressure, etc.).
2. Control Level: This part is also close to the process, and the typical devices are the Programmable Logic Controllers (PLCs). In this part there is a program running that communicates with the field level and exchange the data with the sensors and actuators. The typical protocols are Modbus (TCP and RTU), Profinet, Profibus, Ethernet/IP, Ethercat, DNP3, etc., mainly depending on the PLC's manufacturer.
3. Supervisory Level: In this level the common devices are the HMI and the SCADA. The HMI is closer to the process. One SCADA can connect to many HMIs and/or PLCs. One HMI can connect to one/some PLCs. In the HMI and SCADA systems there is usually a human acting with them and the typical protocols are the same ones with those in control level, but nowadays they also include Open Platform Communications – Unified Architecture (OPC-UA), Message Queuing Telemetry Transport (MQTT) or restful Application Programming Interface (API).
4. Planning Level: Is the level where the Manufacturing Execution System (MES) is placed and this system organized control and monitor the manufacturing process if a factory.
5. Management Level: Here is where the Enterprise Resource Planning (ERP) of the company is located and the typical protocols (in planning level too) are the IT protocols.

In previous times, Operational Technology (OT) networks existed as segregated entities, maintaining no connection to both the Information Technology (IT) network and the broader Internet landscape

[12]. Interaction with OT devices was solely facilitated by OT personnel, with minimal external engagement. Furthermore, the foundational OT protocols were established in a preceding period, leading to a lack of emphasis on security considerations.

Nowadays with IoT and industry 4.0 the OT and IT networks are merging. Therefore, important security considerations need to be included to mitigate cybersecurity risk associated to the growing exposure of the industrial networks as stated in threat landscape surveys for industrial systems [13].

For each level of the industrial communication architecture, there are different protocols, signals and specification. Cybersecurity Industrial Standard mentioned in section 2.1 (mainly IEC-62443) enable recommendations on how to protect and secure the OT network. Figure 9 illustrates an example of a possible network architecture of an IACS network provided by Purdue Enterprise Reference Architecture (PERA) model according to the ISA-99 and IEC 62443 [10]. In this way IEC-62443-3-3 [6] has 7 FR, which come to mitigate the most common issues in the OT networks.

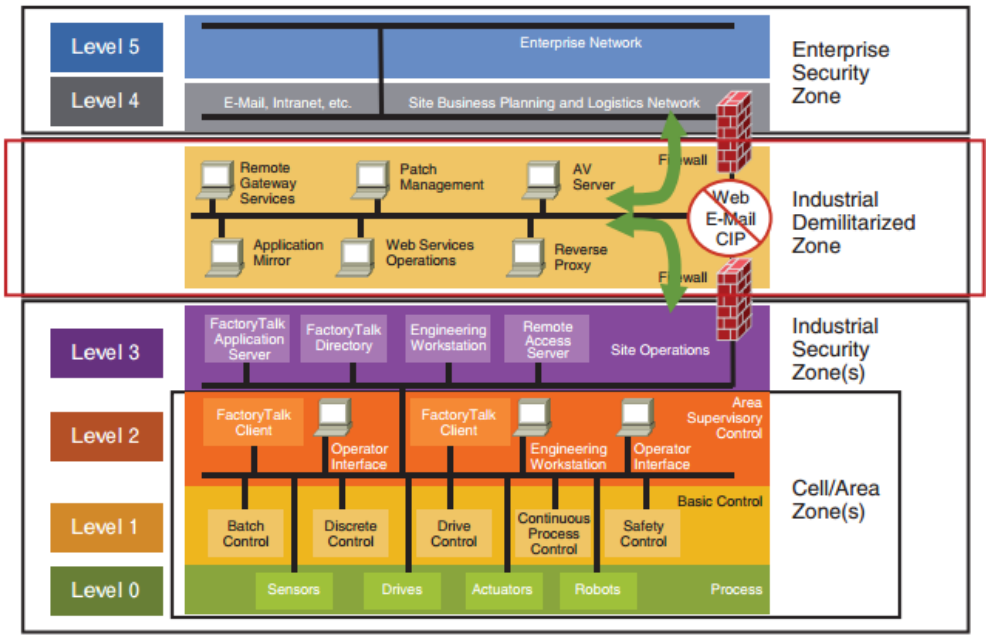


Figure 9. Purdue Enterprise Reference Architecture (PERA) model according to the ISA-99 [10]

2.3. Cybersecurity in IIoT

It is possible to compare a CPS to a large IIoT system or as a box with many IIoT systems inside that monitor and control an industrial process. This is why IIoT approach is also considered in the scope of this SoTA. IIoT is born from the needs of connectivity to transport the data from the field to higher instances. New elements, that were not present in industry, are now deployed and need to be integrated, not only with the existing network but also with new sensors, with new protocols and communication methods. This highlights a big need for reference architectures for IIoT to provide standardized approach. These initiatives aim to facilitate interoperability, simplify development, and ease implementation. Table 2 provides a brief overview of them.

Table 2. IIoT reference architectures [20]

Category	Initiative	Description	Status	URL
IoT reference architecture models	Reference Architecture Model Industrie 4.0 (RAMI 4.0)	A reference architecture for smart factories dedicated to IoT standards, which started in Germany and today is driven by all major companies and foundations in the relevant industry sectors.	Version 1 as of July 2015	www.zvei.org/en/association/specialist-divisions/automation/Pages/default.aspx
	Industrial Internet Reference Architecture (IIRA)	The Industrial Internet Consortium (founded by AT&T, Cisco, General Electric, IBM, and Intel) has delivered a reference architecture for broader consideration and discussion.	Version 1.7 as of June 2015	www.iiconsortium.org
	Internet of Things—Architecture (IoT-A)	The IoT-A delivered a detailed architecture and model from the functional and information perspectives. The project also performed a detailed analysis of system requirements.	The final architectural reference model for the IoT v.3.0 as of July 2013	www.iot-a.eu/public/public-documents/d1.5/view
	Standard for an Architectural Framework for the Internet of Things (IoT)	The IEEE P2413 project has a working group on the IoT's architectural framework, highlighting protection, security, privacy, and safety issues.	An ongoing activity, with no white papers or defining documents as of Sept. 2015	https://standards.ieee.org/develop/project/2413.html
	Arrowhead Framework	This initiative enables collaborative automation by open-networked embedded devices. It's a major EU project to deliver best practices for cooperative automation.	Ongoing updates and release of material by spring 2016	www.arrowhead.eu
Machine-to-machine (M2M) standards relevant to the IoT	European Telecommunications Standards Institute Technical Committee (ETSI TC) for M2M	The TC provides IoT communication standards.	Various available standards and drafts	www.etsi.org/technologies-clusters/technologies/m2m
	International Telecommunication Union Telecommunication Standardization Sector (ITU-T)	The ITU-T has coordination activities on aspects of identification systems for M2M.	Various available standards and drafts	www.itu.int/en/Pages/default.aspx
Further activities	European Research Cluster on the Internet of Things (IREC)	The IREC is involved in many IoTrelated issues, including connected objects, the Web of things, and the future of the Internet.	Ongoing updates	www.internet-of-things-research.eu
	Smart Appliances (SMART) study	This EU-funded study focused on semantic assets for smartappliance interoperability.	Smart-appliance reference ontology definition as of Mar. 2015	https://sites.google.com/site/smartappliancesproject/home or http://ontology.tno.nl/saref

Some IIoT refence architecture initiative to be mentioned are:

- Reference Architectural Model Industry (RAMI 4.0) [14]
- Industrial Internet Reference Architecture (IIRA) v1.9 [16]
- OpenFog reference Architecture (OpenFog RA) [17]
- IoT IEEE P2413 [18]
- Arrowhead Framework [19]

This way, industrial networks, typically disconnected from the Internet, began to need to interconnect their sensors and actuators to the IT systems, and two different worlds with very different specifications were connected. So, IIoT needs to secure the information from the sensor data sent to the application service (mainly located in the cloud) in a continuous way and in time. Therefore, cybersecurity is needed.

The following section show some initiatives and reference architectures aiming to reflect and include cybersecurity aspects in the architecture schemas.

2.3.1. OpenFog Reference Architecture

Fog Computing is a horizontal, system-level architecture that distributes computing, storage, control and networking functions closer to the users along a cloud-to-thing continuum. Fog computing

provides the missing link in the cloud-to-thing continuum. Fog architectures selectively move compute, storage, communication, control and decision making closer to the network edge where data is being generated in order to solve the limitations in current infrastructure to enable mission-critical, data-dense use cases. The OpenFog Consortium was formed on the principle that an open fog computing architecture is necessary in today's increasingly connected world. The OpenFog Reference Architecture (OpenFog RA) [17] is intended to help business leaders, software developers, silicon architects and system designers create and maintain the hardware, software and system elements necessary for fog computing.

Figure 10 presents an abstract architecture including perspectives, shown in grey vertical bars on the sides of the architectural description.

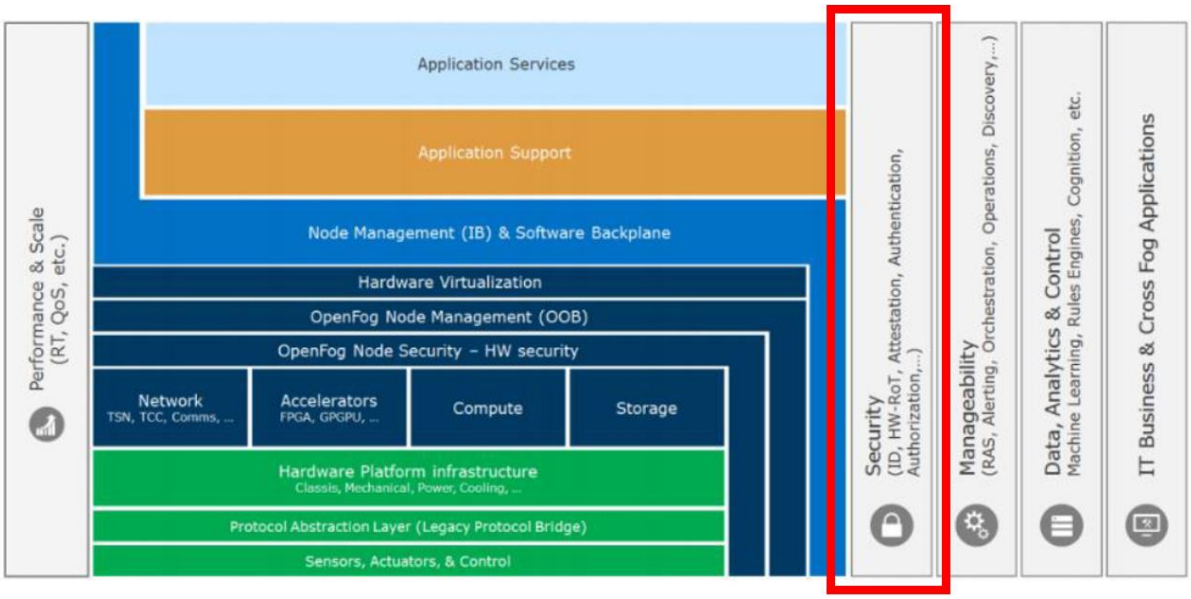


Figure 10. OpenFog IoT architecture description [17]

In terms of the Security perspective, end-to-end security is critical to the success of all fog computing deployment scenarios. If the underlying silicon is secure, but the upper layer software has security issues (and vice versa) the solution is not secure. Data integrity is a special aspect of security for devices that currently lack adequate security. This includes intentional and unintentional corruption.

2.3.2. IoT-A Reference Architecture

The IoT-A Reference Architecture [21] is designed as a reference for the generation of compliant IoT concrete architectures that are tailored to one's specific needs. It provides a Functional View diagram including the nine groups of the Functional Model, as shown in Figure 11:

- The Application Functionality Group (FG) and Device FG are out-of-scope of the IoT-A Reference Architecture and are coloured in yellow.
- Management FG and Security FG are transversal FGs and are coloured dark blue.

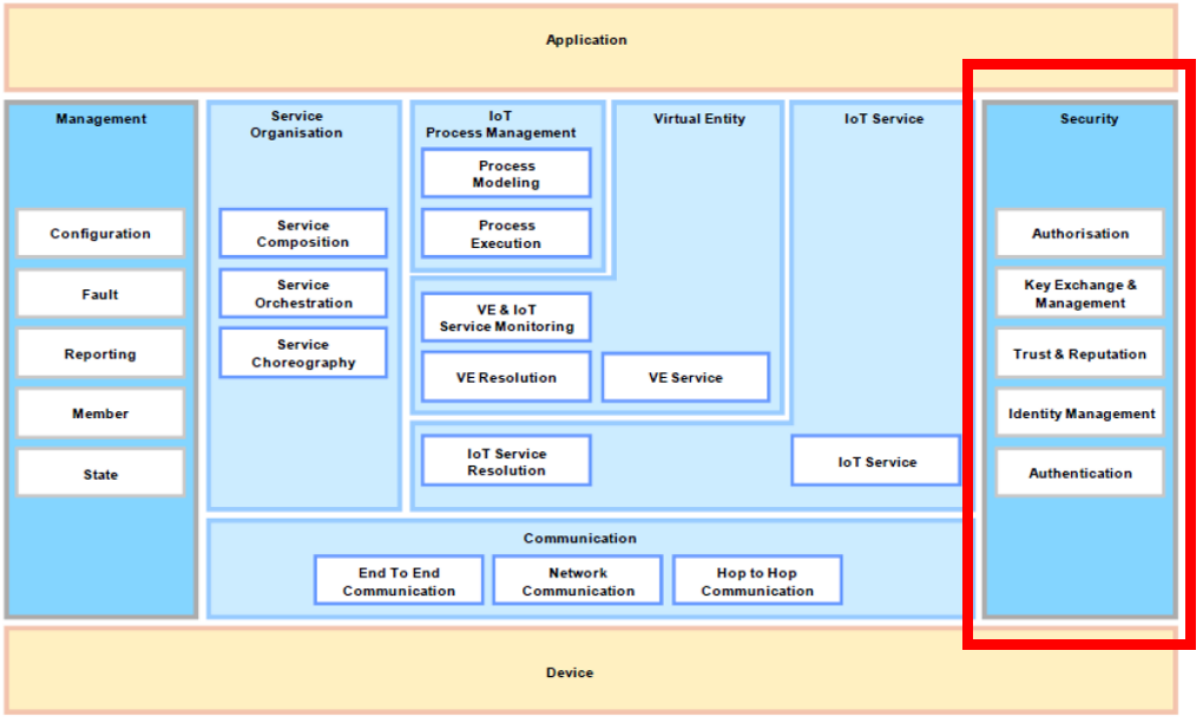


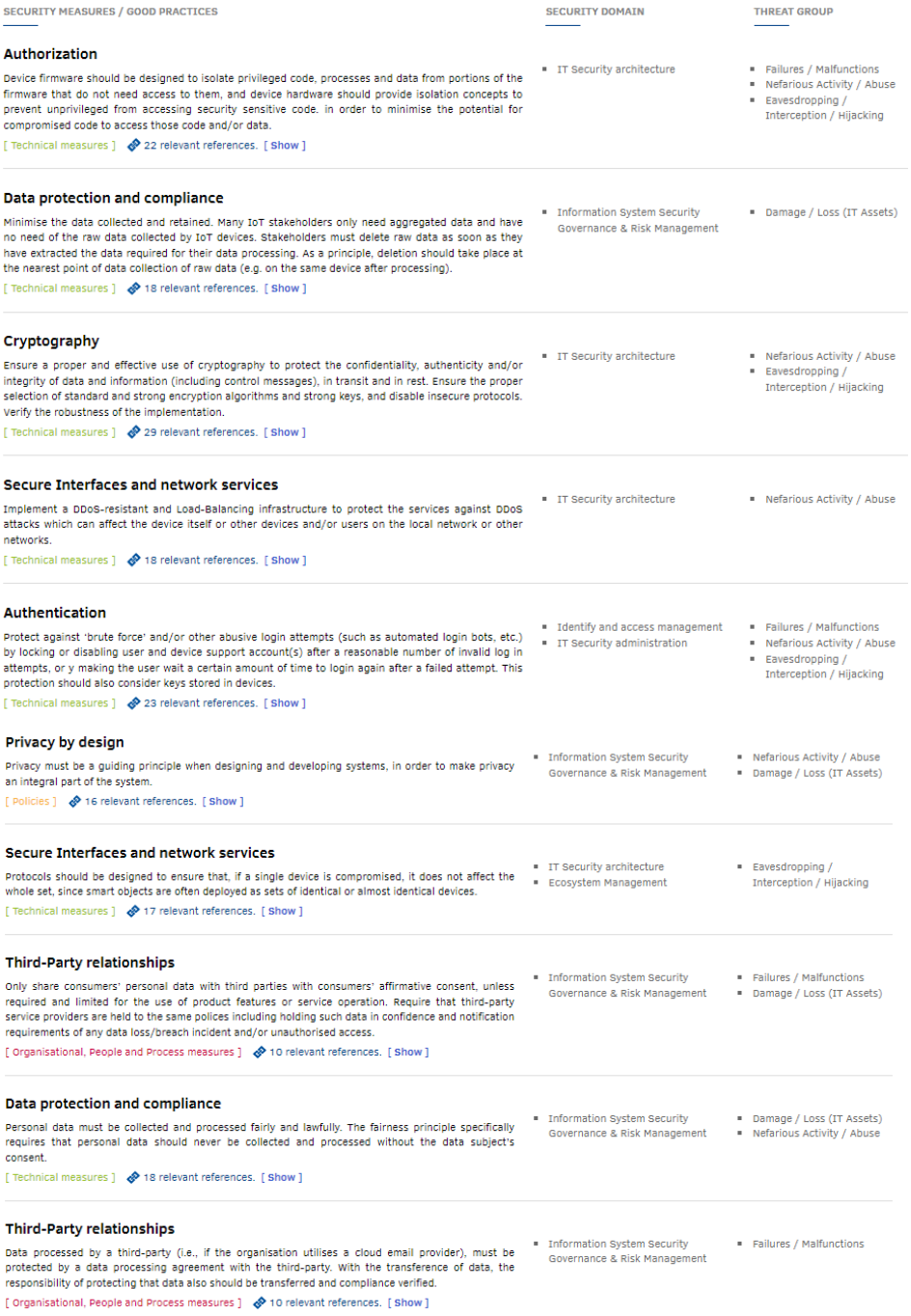
Figure 11. Functional-decomposition viewpoint of the IoT-A reference architecture [21]

The Security FG is responsible for ensuring the security and privacy of IoT-A-compliant systems. It consists of five functional components:

- Authorization: manages and enforces access control policies. It provides services to manage policies (CUD), as well as taking decisions and enforcing them regarding access rights of restricted resources.
- Key Exchange & Management: is used for setting up the necessary security keys between two communicating entities in an IoT system.
- Trust & Reputation: manages reputation scores of different interacting entities in an IoT system and calculates the service trust levels.
- Identity Management: manages the different identities of the involved Services or Users in an IoT system.
- Authentication: verifies the identity of a User and creates an assertion upon successful verification. verifies the identity of a User and creates an assertion upon successful verification.

2.3.3. ENISA good practices for IoT, Smart infrastructures and the I4.0

As expected, all the cybersecurity standards have almost the same key points to secure networks: ENISA's (European Union Agency for Cybersecurity) good practices for IoT and smart infrastructure [22], provides a consolidated web format baseline security measures and good practices as they are listed in ENISA's report "Baseline security recommendations for IoT" that was published in 2017 [81]. It includes "Good practices" for specific filters, such as Security Measures Category, Security Domains, Threat Groups or even specific Standards. This is presented in Figure 12.



SECURITY MEASURES / GOOD PRACTICES	SECURITY DOMAIN	THREAT GROUP
Authorization Device firmware should be designed to isolate privileged code, processes and data from portions of the firmware that do not need access to them, and device hardware should provide isolation concepts to prevent unprivileged from accessing security sensitive code. in order to minimise the potential for compromised code to access those code and/or data. [Technical measures] 22 relevant references. [Show]	▪ IT Security architecture	▪ Failures / Malfunctions ▪ Nefarious Activity / Abuse ▪ Eavesdropping / Interception / Hijacking
Data protection and compliance Minimise the data collected and retained. Many IoT stakeholders only need aggregated data and have no need of the raw data collected by IoT devices. Stakeholders must delete raw data as soon as they have extracted the data required for their data processing. As a principle, deletion should take place at the nearest point of data collection of raw data (e.g. on the same device after processing). [Technical measures] 18 relevant references. [Show]	▪ Information System Security Governance & Risk Management	▪ Damage / Loss (IT Assets)
Cryptography Ensure a proper and effective use of cryptography to protect the confidentiality, authenticity and/or integrity of data and information (including control messages), in transit and in rest. Ensure the proper selection of standard and strong encryption algorithms and strong keys, and disable insecure protocols. Verify the robustness of the implementation. [Technical measures] 29 relevant references. [Show]	▪ IT Security architecture	▪ Nefarious Activity / Abuse ▪ Eavesdropping / Interception / Hijacking
Secure Interfaces and network services Implement a DDoS-resistant and Load-Balancing infrastructure to protect the services against DDoS attacks which can affect the device itself or other devices and/or users on the local network or other networks. [Technical measures] 18 relevant references. [Show]	▪ IT Security architecture	▪ Nefarious Activity / Abuse
Authentication Protect against 'brute force' and/or other abusive login attempts (such as automated login bots, etc.) by locking or disabling user and device support account(s) after a reasonable number of invalid log in attempts, or y making the user wait a certain amount of time to login again after a failed attempt. This protection should also consider keys stored in devices. [Technical measures] 23 relevant references. [Show]	▪ Identify and access management ▪ IT Security administration	▪ Failures / Malfunctions ▪ Nefarious Activity / Abuse ▪ Eavesdropping / Interception / Hijacking
Privacy by design Privacy must be a guiding principle when designing and developing systems, in order to make privacy an integral part of the system. [Policies] 16 relevant references. [Show]	▪ Information System Security Governance & Risk Management	▪ Nefarious Activity / Abuse ▪ Damage / Loss (IT Assets)
Secure Interfaces and network services Protocols should be designed to ensure that, if a single device is compromised, it does not affect the whole set, since smart objects are often deployed as sets of identical or almost identical devices. [Technical measures] 17 relevant references. [Show]	▪ IT Security architecture ▪ Ecosystem Management	▪ Eavesdropping / Interception / Hijacking
Third-Party relationships Only share consumers' personal data with third parties with consumers' affirmative consent, unless required and limited for the use of product features or service operation. Require that third-party service providers are held to the same polices including holding such data in confidence and notification requirements of any data loss/breach incident and/or unauthorised access. [Organisational, People and Process measures] 10 relevant references. [Show]	▪ Information System Security Governance & Risk Management	▪ Failures / Malfunctions ▪ Damage / Loss (IT Assets)
Data protection and compliance Personal data must be collected and processed fairly and lawfully. The fairness principle specifically requires that personal data should never be collected and processed without the data subject's consent. [Technical measures] 18 relevant references. [Show]	▪ Information System Security Governance & Risk Management	▪ Damage / Loss (IT Assets) ▪ Nefarious Activity / Abuse
Third-Party relationships Data processed by a third-party (i.e., if the organisation utilises a cloud email provider), must be protected by a data processing agreement with the third-party. With the transference of data, the responsibility of protecting that data also should be transferred and compliance verified. [Organisational, People and Process measures] 10 relevant references. [Show]	▪ Information System Security Governance & Risk Management	▪ Failures / Malfunctions

Figure 12. ENISA's good practices for IoT and smart infrastructure [22]

More specifically, one of the key strengths of the ENISA Good Practices tool is its adaptability to various sectors and use cases. Whether it's healthcare, energy, transportation or any other domain, the tool offers practical insights and tailored guidance for securing IoT devices and infrastructure elements specific to each sector's needs. Additionally, the tool emphasizes the importance of collaboration and information sharing among stakeholders, fostering a community approach to cybersecurity in the IoT ecosystem. As IoT continues to reshape industries and connect more devices and systems, the ENISA tool serves as a vital resource to help organizations navigate the complexities of IoT security and build robust, resilient and trustworthy IoT environments that can thrive in the digital age. The aforementioned tool developed by ENISA is presented in **Appendix D ENISA Good practices for IoT and Smart Infrastructures Tool**.

2.3.4. NIST Cybersecurity IoT Program

The National Institute of Standards and Technology (NIST) is an agency of the United States Department of Commerce whose mission is to promote American innovation and industrial competitiveness. NIST has developed the NIST Cybersecurity IoT Program, a cybersecurity standard wide extended and implemented in USA. The main goal of the standard is to secure networks they have published several articles to protect not only IT network but also OT networks. For instance, NIST Cybersecurity IoT program [23] aims at fostering cybersecurity for devices and data in the IoT ecosystem, across industry sectors and at scale.

NIST's Cybersecurity for the Internet of Things (IoT) program supports the development and application of standards, guidelines, and related tools to improve the cybersecurity of connected devices, products, and the environments in which they are deployed. By collaborating with stakeholders across government, industry, international bodies, academia, and consumers, the program aims to cultivate trust and foster an environment that enables innovation on a global scale. Up to date this program has published several whitepapers mostly focused in baseline IoT device cybersecurity along with specifications for IoT device manufacturers.

2.3.5. IoT Security Maturity Model

The goal of a Security Maturity Model (SMM) [24] is to provide a path for Internet of Things (IoT) providers to know where they need to be, and how to invest in security mechanisms that meet their requirements without overinvesting in unnecessary security mechanisms. Figure 13 illustrates the structure of the SMM and the breakdown of security maturity domains.

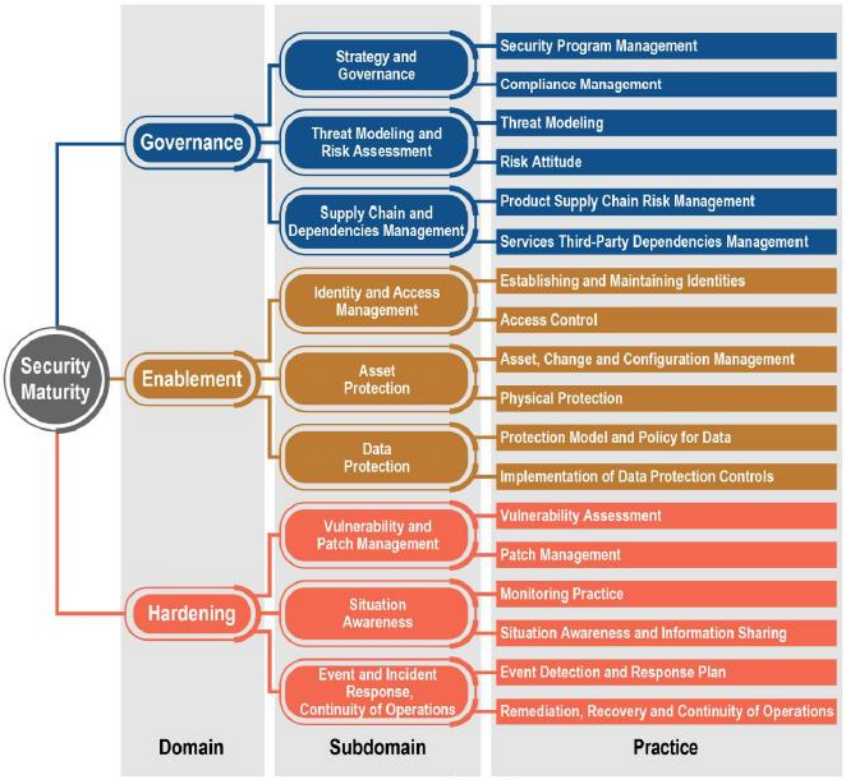


Figure 13. IoT Security Maturity Model- Security Maturity Domains [58]

Domains are the high-level views to capturing the key aspects of security maturity and determining the priorities of security maturity enhancement at the strategic level. At this level, the stakeholder determines the priorities of the direction in improving security.

This Security Maturity Level is organized in three domains: Governance, Enablement and Hardening. The Governance domain encompasses the establishment of a strategic framework that guides the organization's overall security strategy and ensures alignment with business goals and compliance requirements. Its purpose within a security maturity model is to provide a structured approach to managing and overseeing cybersecurity initiatives. By defining policies, procedures, and responsibilities, organizations can make informed decisions regarding risk management, resource allocation, and security investments. Governance also involves regular audits and assessments to evaluate the effectiveness of security measures and identify areas for improvement. Through strong governance, organizations can create a culture of security awareness and accountability across all levels.

The Enablement domain is focused on providing the necessary tools, resources, and technologies to support effective cybersecurity practices. Its purpose is to empower individuals and teams with the means to implement security measures efficiently. This domain encompasses training programs, security awareness campaigns, and the deployment of security solutions. By enabling employees to understand and adopt security best practices, organizations can reduce human error and enhance their overall security posture. Effective enablement equips the workforce with the knowledge and skills needed to identify and respond to security threats and incidents.

The Hardening domain involves the process of strengthening IT systems, networks, and applications to minimize vulnerabilities and potential attack vectors. Its purpose is to create a robust defense against cyber threats by reducing the attack surface. Hardening encompasses activities such as applying security patches, configuring systems according to security benchmarks, and implementing access controls. By hardening systems and infrastructure, organizations can significantly reduce the risk of successful attacks and unauthorized access. This domain plays a critical role in ensuring that technology assets are resilient in the face of evolving threats.

Domains have different key aspects to it, called subdomains. Subdomains reflect the basic means of obtaining these priorities at the tactical level. At this level, the stakeholder identifies the typical needs for addressing security concerns. Finally, practices define typical activities associated with domains and identified at the planning level. At this level, the stakeholder considers the purpose of specific security activities.

2.3.6. ISO/IEC TR 30141

ISO/IEC 30141 provides a reference architecture for IoT. Additionally, it offers brief directions to secure IoT networks. Due to their distributed nature and the diverse nature of the entities that form IoT networks, they present a very large attack surface which poses a significant challenge while trying to maintain security across the network. To overcome this challenge the use of an information security management system (ISMS) is advised. This system should be able to detect the risks faced by the network and implement sets of security controls that can be applied to the IoT system to address them. The development of the ISMS should be parallel with the rest of the system and it should be updated when parts of the IoT network change. Moreover, testing and validation of the security controls implemented by the ISMS should be done regularly to test their efficiency. Finally, the elements of the ISMS should be checked against known vulnerabilities or when exposed to security incidents.

2.3.7. ISO/IEC TR 30164

ISO/IEC TR 30164 deals with the security of IoT networks that utilize edge computing focusing on three aspects, secure design, secure communications, and redundancy/adaptability of the network. The following aspects are emphasized for the design phase of the network: Initially, the design of the network should consider foundational security principles that encompass securing information to ensure availability, integrity, and confidentiality. Additionally, the design should be secure meaning that it should ensure the secure operation of systems to prevent hijacking and vulnerabilities while maintaining availability against threats such as DDoS attacks. If threats exist that cannot be dealt by design, mechanisms should be deployed that detect, log and report attacks and other disruptive incidents. Security related functions should be flexible in terms of deployment and easy to scale. Security functions used in the edge of the network should be adapted to the specifics of the edge architecture e.g., consider limited resources.

Concerning the communications of the network, the operator should ensure that management and access to the entities using or being part of the network are subject to authorization and authentication. Moreover, entities can only communicate with other authorized entities. appropriate data protection principles should be implemented for personal data storage, processing, and transmission across networks.

Finally, concerning the redundancy/adaptability of the network, the system should be provisioned to continuously mitigate attacks within a certain period, while being able to tolerate function failures within a specified range and limit. During this time its basic functions run properly. Finally, it should ensure that the entire system can quickly recover from failure.

2.4. Cybersecurity in 5G architectures

With the advancement of information and communication technologies, fifth generation (5G) has become an emerging communication medium to support higher speed, lower latency, and massive connectivity to various devices by leveraging the evolution of 4G with the addition of new radio technology, service-based architecture, and cloud infrastructure. Additionally, 5G technology has been designed considering industrial use, and there are numerous benefits of companies using non-public 5G networks: they realize latency, scalability, availability, reliability, ubiquitous mobility, and fog computing, which are needed for critical massive IoT applications.

Nonetheless, the introduction of new technologies and advanced features in 5G communications gives rise to new security requirements and challenges. Figure 14 shows a categorization on security aspects for 5G.

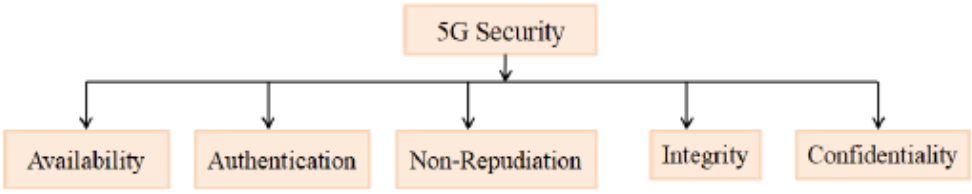


Figure 14. Categorization of security in 5G [25]

According to the figure above, 5G comes to ensure the following key points [25]:

- **Availability**: 5G networks—including the business support systems (BSS) providing such critical functions as charging and policy—must provide 99.999%, or "five nines," of data availability annually. This equates to just six minutes of unscheduled downtime per year.
- **Authentication**: Service-based architecture (SBA) has been proposed for the 5G core network. Accordingly, new entities and new service requests have also been defined in 5G. Some of the new entities relevant to 5G authentication are listed below.
    - The Security Anchor Function (SEAF) is in a serving network and is a "middleman" during the authentication process between a User Equipment (UE) and its home network. It can reject an authentication from the UE, but it relies on the UE's home network to accept the authentication.
    - The Authentication Server Function (AUSF) is in a home network and performs authentication with a UE. It makes the decision on UE authentication, but it relies on backend service for computing the authentication data and keying materials when 5G-AKA or EAP-AKA' is used.
    - Unified data management (UDM) is an entity that hosts functions related to data management, such as the Authentication Credential Repository and Processing Function (ARPF), which selects an authentication method based on subscriber identity and configured policy and computes the authentication data and keying materials for the AUSF if needed.
    - The Subscription Identifier De-concealing Function (SIDF) decrypts a Subscription Concealed Identifier (SUCI) to obtain its long-term identity, namely the Subscription Permanent Identifier (SUPI), e.g., the IMSI. In 5G, a subscriber long-term identity is always transmitted over the radio interfaces in an encrypted form. More specifically, a public key-based encryption is used to protect the SUPI. Therefore, only the SIDF has access to the private key associated with a public key distributed to UEs for encrypting their SUPIs.

A unified authentication framework has been defined to make 5G authentication both open, as depicted in Figure 15 (e.g., with the support of Extensible Authentication Protocol (EAP)) and access-network agnostic (e.g., supporting both 3GGP access networks and non-3GPP access networks such as Wi-Fi and cable networks).

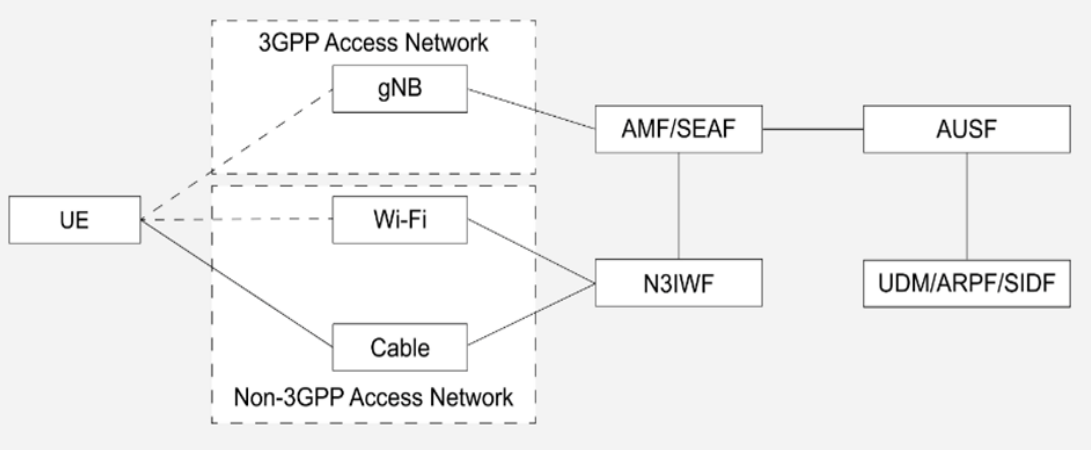


Figure 15. 5G Authentication Framework

- **Non-repudiation**: non-repudiation ensures that no party can deny that it sent or received a message via encryption and/or digital signatures or approved some information.
- **Integrity**: In 5G, integrity protection of the user plane (UP) between the device and the gNB, was introduced as a new feature. Like the encryption feature, the support of the integrity protection feature is mandatory on both the devices and the gNB while the use is optional and under the control of the operator.
- **Confidentiality**: In the 5G system, Subscription Concealed Identifier (SUCI) is a privacy preserving identifier containing the concealed SUPI.

These key points align with the Foundational Requirements of IEC-62443 and other mentioned Security in Industry related standard, presented in section 2.1.

5G can be combined with various technological solutions, as it is shown in the image below (Figure 16), but always keeping in mind the 5 security categories mentioned above.

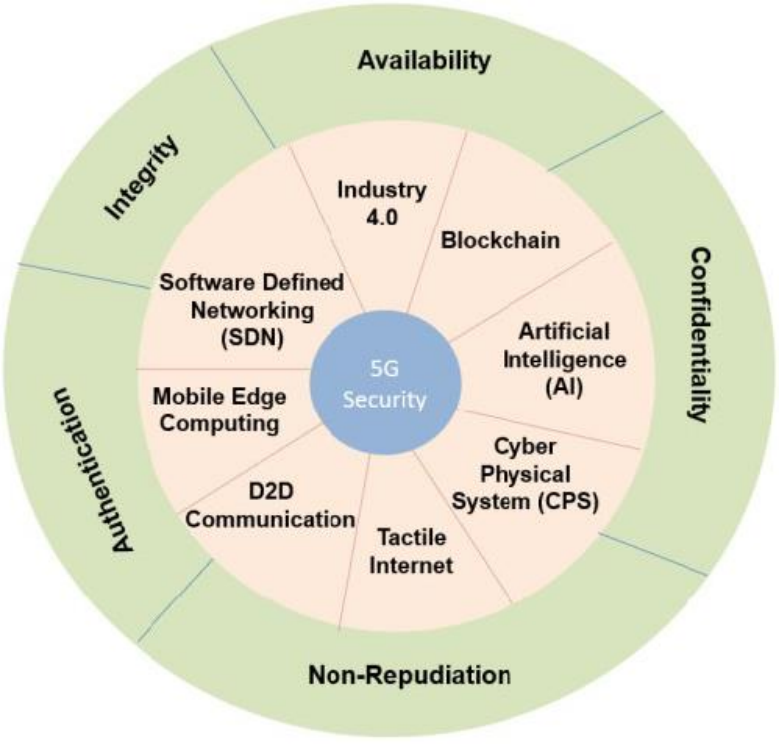


Figure 16. Technologies used and or related to security in 5G [25]

2.5. Cybersecurity in CPSoS

2.5.1. CPSoS

CPS are systems that integrate computing elements with the physical components and processes. The computing elements coordinate and communicate with sensors, which monitor cyber and physical indicators, and actuators, which modify the cyber and physical environment. CPSoS are connected CPSs. They are large complex systems where physical elements interact with and are controlled by many distributed and networked computing elements.

CPS is a fundamental enabler of Industry 4.0. Therefore, cybersecurity in these components need to be considered. Figure 17 shows a common attack surface on CPSs.

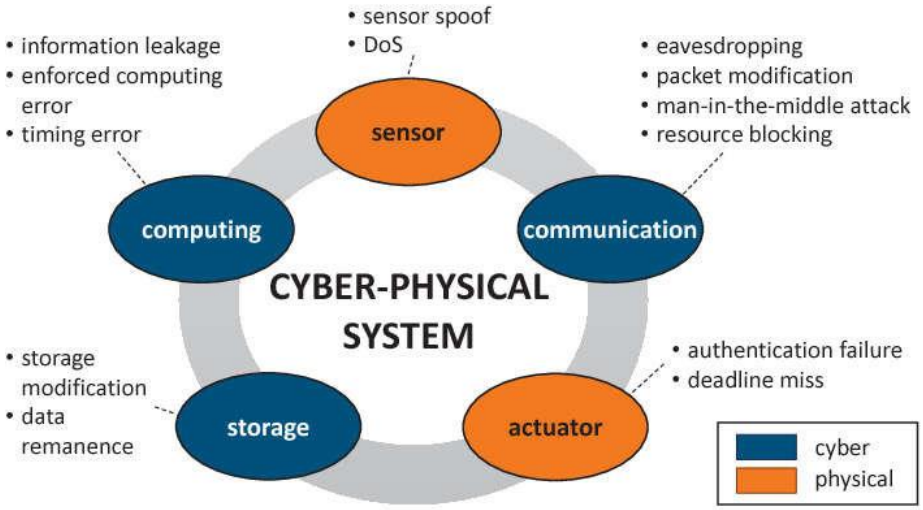


Figure 17. CPS attack surface

Cybersecurity providers' responsibility is to be aware of this attack surface and the related threat modelling and cyber kill chain and be prepared to react and mitigate this kind of attacks.

A cybersecure CPSoS is a complex and interconnected network of Cyber-Physical Systems designed to operate securely in a digital environment. Its primary goal is to protect the confidentiality, integrity, and availability of data and operations while operating in a connected and technology-driven ecosystem. Here are the main elements of a cybersecure CPSoS:

Security-Centric Design: A cybersecure CPSoS is designed from the ground up with cybersecurity as a foundational principle. Security considerations are integral to the system's architecture, components, and operations.

Secure Architecture: The CPSoS is built upon a secure architectural framework that includes mechanisms to protect against cyber threats. This architecture employs secure communication protocols, access controls, and data encryption to safeguard the system.

Encrypted Communication: All communication within the CPSoS, whether between individual systems or with external entities, is encrypted and authenticated. Encryption ensures data confidentiality, integrity, and protection against unauthorized access.

Access Control: Access to the CPSoS and its components is strictly controlled through robust access control mechanisms. Only authorized personnel or systems are granted access, and permissions are based on a need-to-know basis.

Continuous Monitoring: The CPSoS is subject to continuous monitoring for any signs of suspicious activity or security breaches. Intrusion detection systems, security information and event management tools, and anomaly detection are used to detect and respond to threats in real-time.

Patch Management: A comprehensive process for managing security updates and patches is established to address known vulnerabilities promptly. Regular updates of software and firmware components help maintain a secure posture.

Redundancy and Resilience: Redundant components and resilient design principles are integrated into the CPSoS to ensure system availability and operational continuity even in the face of cyberattacks or component failures.

Security Training: Personnel responsible for operating and maintaining the CPSoS receive training in cybersecurity best practices. There is a focus on fostering a culture of security awareness to mitigate human-related risks.

Incident Response Plan: A well-defined incident response plan is in place to guide actions in the event of a cybersecurity incident. This plan outlines procedures for investigating, mitigating, and recovering from security breaches.

Compliance with Standards: The CPSoS aligns with relevant cybersecurity standards and regulations or industry-specific standards, to ensure adherence to recognized security practices.

2.5.2. IEC61499 Standard

IEC 61499 is an international standard for the design of distributed control systems in industrial automation. It provides a framework for designing and implementing control applications that can be distributed across multiple devices and executed in a decentralized manner. The IEC61499 standard described the following key characteristics of such a distributed control system:

Function Blocks:

IEC 61499 introduces the concept of function blocks, which are modular units of control logic. Function blocks encapsulate specific control functions (Algorithm) with data inputs and outputs and event inputs and outputs which is encapsulated in a basic function blocks type, that can utilize an execution control chart (ECC) to control the execution of its algorithms. This basic function block types can be interconnected within a composite function block type. Both can be interconnected to create complex control applications, and each can be distributed to different devices or aggregated to subapplications:

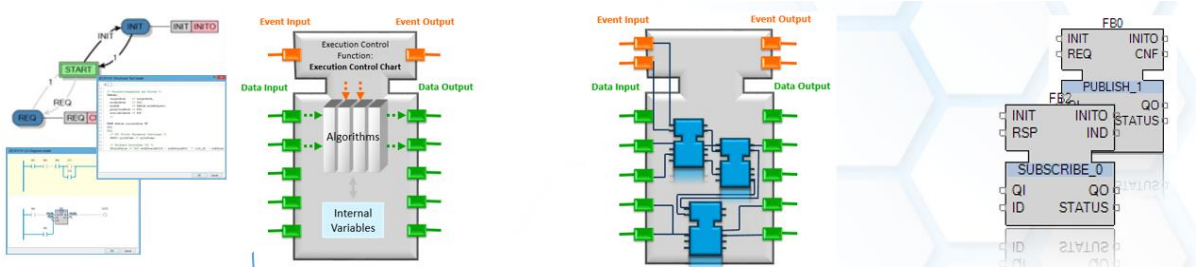


Figure 18. ECC, Basic FB (with ECC and algorithm), Composite (network of FBs) and SFB (additional communication)

Event-Driven:

Unlike traditional PLC programming with the ICE61131 standard, IEC 61499 is event-driven. Function blocks react to events and can trigger other events, allowing for more flexible in dynamic control systems, as well a data reduction between different devices or layers in comparison to cyclic systems.

Distributed Control and Communication:

IEC 61499 is a standard, able to design distributed control systems, where control tasks are distributed across multiple devices or controllers. This allows a greater scalability and flexibility in industrial automation systems.

The standard includes mechanisms for communication between function blocks and devices within a distributed control system. This ensures that information can be exchanged efficiently between different parts of the system. Service function blocks can extend the and simplify the communication to IT applications, when integrated in the composite function block network as standard communication method, which simplifies the usage of services interfaces over different protocols and semantics to couple real-time automation with enterprise applications.

Hierarchical Structure:

Control applications in IEC 61499 are organized in a hierarchical structure. The standard defines several levels, including the system, device, and resource levels, to manage the distribution of control tasks. Additionally, the nesting of basic functions blocks within composite functions blocks, nested in another composite function block, makes hierarchical design and communication possible, as well the distributed design, when deploying different function block instances to different devices.

Reusability:

IEC 61499 promotes the reusability of control logic by allowing function blocks to be developed and tested independently. These function blocks can be combined to software objects, which can be reused in multiple control applications, reducing the development and engineering time and effort. This fundamentally object-oriented design facilitates the re-use via software component libraries.

Portability:

Control applications developed according to IEC 61499 are designed to be portable across different hardware platforms and vendors, promoting interoperability in industrial automation systems. They are rather Application/Asset-centric than controller-centric:

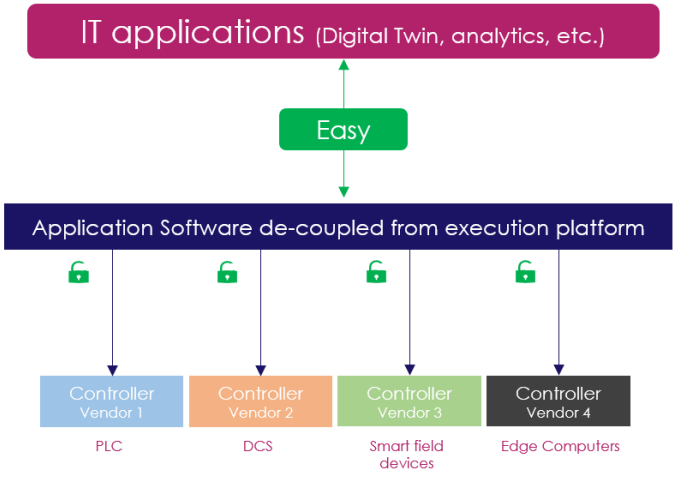


Figure 19. Portable Control Application Software and Enterprise Communication

In summary, IEC 61499 introduces the concept of function blocks and event-driven programming, allowing for greater flexibility, scalability, and reusability in control applications. This standard plays a crucial role in modernizing and enhancing industrial automation and control processes, because its key

characteristics are important for next generation automation systems, like the following figure (Figure. 20) shows:

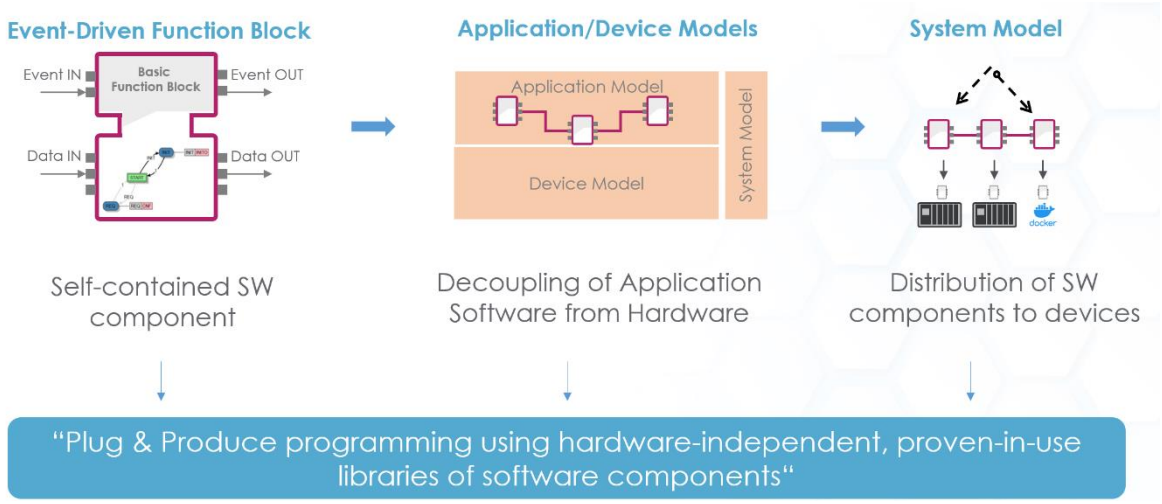


Figure. 20 Next Generation Automation System with the IEC61499 standard

2.5.3. CPSoS and IEC61499 and Cybersecurity

Regarding cybersecurity IEC 61499 does not explicitly address cybersecurity, anyway the effective implementation of IEC 61499 can be applied in a way that enhance the security of cyber-physical systems (CPSoS). Cybersecurity considerations when designing control systems based on the IEC 61499 standard, can be applied implementing cybersecurity best practices, and security controls such us: Access Control and Authorization, Secure Communication, Data Integrity and Authentication, Code Integrity and Software Updates, Event Logging and Monitoring, Redundancy and Fault Tolerance, Isolation and Segmentation, Secure Deployment and Configuration, Security Training and Awareness, Vendor Assessment, Threat Modelling and Incident Response Plan. Applying this security controls, based on recognized best practices, will significantly improve the security posture of control systems designed using IEC 61499. Additionally, it is worth to mention that it is important to integrate cybersecurity into the entire lifecycle of the control system, from design to operation and maintenance.

Another cybersecurity aspect that should be addressed when talking about CPSoS is the integration of new CPSs, based on IEC61499, into existing CPSoS. This integration should make existing CPS more cybersecure.

To support this, the IEC 61499 engineering environment and runtime (RT) can integrate security measures on design time:

- Setting up users and defining their rights, encrypted deploy to dedicated hardware (HW) devices, signed setup installation, and provide secure communication with other devices and tools on the IT level. This would enable only integrate new secure CPSs into existing CPSoS. One measure is the IEC 61499 runtime is Account Management, which means that each runtime system can be configured with users, passwords, and permissions. It can be selected what user has which rights on the runtime: some users can only deploy, some can watch, and others cannot even connect to the RT on some interfaces.

- Another measure is Certificate management: Certificates are used to encrypt connections and for authentication to know that someone who wants to communicate with the runtime is who he/she is presenting to be.
- For sure the communication between all entities involved in an automation system should be secured by leveraging the authentication and authorization of entities. This way a secure interaction between CPSs and other systems can be provided. Ciphered communication protocols can be leveraged to provide data privacy in a CPSoS. This aspect is in the IEC 61499 engineering environment supported on application level where different FBs can be used to encrypt and decrypt data sent from one device to another.
- Besides this, when an IEC 61499 application is compiled in the engineering environment, the compiler creates a binary file for each FB in the application that is being deployed to the RT and the FBs are signed and verified to prevent manipulation between the studio and the RT. The compiler signs each FB, and the RT reads and verifies these signatures.

2.6. Cybersecurity for Federated Learning

Federated Learning is a decentralized machine learning algorithm which aims to collaboratively train a model across multiple devices or edge nodes without the need to collect the data to a central location. The method has been introduced by Google as to train a shared machine learning model on the data of millions of clients while ensuring privacy at the same time. In FL, participating clients/nodes are responsible for training a model on the data they have, and exchange only the updates to the shared model with the central server. The central server is responsible to aggregate the updates collected by the clients and calculate the next version of the shared model.

A typical implementation of Federated Learning algorithm consists of the following phases: The first stage is Initialization, the second Client selection, next comes Local training and the final, fourth stage is called Aggregation. Stages 2 to 4 are iterative until the number of rounds defined have been completed or until the model has reached a specific performance. The complete pseudo-code of the FederatedAveraging [26] algorithm, presented in Table 3, is typical approach of FL based approaches.

Table 3. Algorithm for Federated Averaging

Algorithm 1 *FederatedAveraging*. The *K* clients are indexed by *k*; *B* is the local minibatch size, *E* is the number of local epochs, and *η* is the learning rate.

Server executes:

initialize w0

for each round t = 1, 2, . . . **do**

$m \leftarrow \max(C \cdot K, 1)$

$S_t \leftarrow$ (random set of m clients)

for each client $k \in S_t$ **in parallel do**

$w_{t+1}^k \leftarrow ClientUpdate(k, w_t)$

$m_t \leftarrow \sum_{k \in S_t} n_k$

$$w_{t+1} \leftarrow \sum_{k \in S_t} \frac{n_k}{m_t} w_{t+1}^k$$

ClientUpdate*(k,w)*: *// Run on client k*

$\mathcal{B} \leftarrow (split\ \mathcal{P}_k\ into\ batches\ of\ size\ B)$

for each local epoch *i* from 1 to *E* **do**

for batch $b \in \mathcal{B}$ **do**

$w \leftarrow w - \eta \nabla \ell(w; b)$

return *w* to server

Federated Learning unique features of collaborative training on distributed datasets and the use of models on the edge, bring forth various important advantages: the first and most important advantage is that it enhances data protection since raw data do not need to leave the device. This saves resources, distributed storage and processing of data significantly reduces bandwidth and energy consumption while the need of a large central server with extreme processing capabilities is not required. Models reside on the edge, inference is done on the edge without the need of communicating again with the central server and at same time allows for real-time continual learning.

Due to the above advantages FL has been widely adopted by scientific community and has been applied extensively in many fields such as finance, healthcare, smart cities and Internet of Things in a wide range of applications next word prediction [26] object detection [27], industrial IoT [28] etc.

2.6.1. Threats, attacks and defences

The adoption of FL by fields with particular sensitivity in data privacy and security, requires robust data protection and security mechanisms. To consider a Federated Learning system safe it should protect not only against adversarial clients but should also ensure that the data will remain private even against the central server.

In the context of cybersecurity there are several threat models associated with Federated Learning. Different threats occur in the various phases of FL by different adversaries. In the context of security Federated Learning can be divided into three different phases where each of them has different vulnerabilities and thus face different security and privacy threats [26].

The first phase involves Data and behaviour auditing: In this phase there are two possible threats. First the client data might be of low quality with inaccuracies in labels and features and secondly the client itself might be malicious or might have been compromised by adversaries.

Next follows the Training phase, in which the system utilizes the involved client's data and computational capabilities to train collaboratively the shared model. This gives the opportunity to adversaries that have compromised clients to manipulate the data, the model gradients and the parameters to attack the global model. Malicious actions in this phase can also come from the central server which might use the updates collected by the clients to deduce sensitive information about their training data. Lastly, due to the exchange of data between the server and the clients this phase is prone to eavesdropping attacks.

Finally, in the Predicting phase the trained model is shared and thus makes this phase prone to evasion and privacy inference attacks. Evasion aims to corrupt the model to produce false predictions while inference attacks aim to deduce information and reconstruct the data used for training. The entire process is schematically shown in Figure 21.

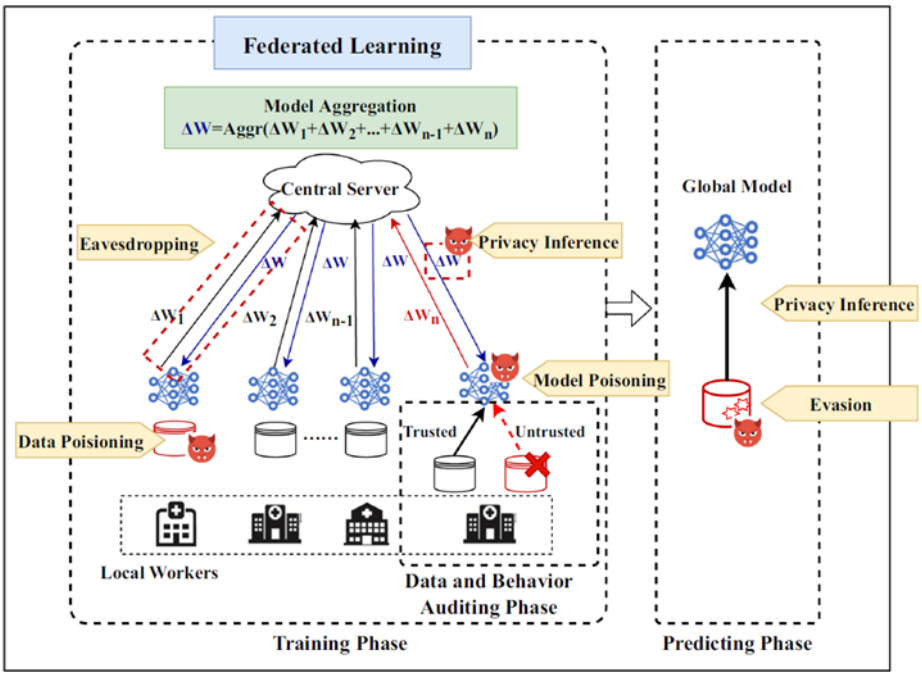


Figure 21. The multi-phases framework of FL including data and behaviour auditing, model training, model predicting along with various threats from [26]

2.6.1.1 Threats and attacks

- ***Poor data quality and malicious behaviour (Auditing phase)***

  Involved clients/nodes are susceptible to external adversaries which aim to exploit possible vulnerabilities to gain access or corrupt the data and the client's system. In addition, not all data are equal, meaning that some of the data collected by the clients might be of low quality or even poisoned by an adversary.

- ***Poisoning attacks (Auditing phase, Training phase)***

  Poisoning attacks pose a critical security concern for a Federated Learning system and aim to reduce the accuracy of the model, untargeted attacks, or to inject a backdoor, targeted attacks. Adversaries aim to influence client's training and result to a shared model that produces false predictions. Poisoning is achieved either by the introduction of malicious training samples, tampering of existing samples, modifying model parameters or by sending specific gradient updates to the central server that help achieve the adversarial goal.

  In addition, targeted and untargeted attacks both can be further divided based on into data poisoning and model poisoning. Usually, data poisoning occurs during the data collection, examples of such attacks include adding noise or flipping the labels of the data. Model poisoning on the other hand aims to tamper with model updates and leads the server to aggregate a corrupt

shared model. Since this attack requires access to model updates it is only feasible during the training phase of the algorithm.

- ***Privacy inference (Training and prediction phase)***

  Even though the main feature of Federated Learning is that preserves privacy by not requiring to send data to the central server studies have shown that the gradients and the shared model can be used to extract sensitive information about participating clients. Eavesdropping attacks are included in privacy inference since the adversary steal model weights and aims to extract meaningful information from them. Privacy inference attacks can be further categorized based on the goal of the adversary performing the attack into:

  - ***Membership inference attacks***, in which the adversary aims to identify if specific samples were used during training of the shared model.
  - ***Class representative inference attacks****, in which the adversary aims to infer prototypical samples representative of the ones used during training.*
  - ***Data properties inference attacks****, in which the adversary aims to certain properties that exist in the training data of other participants.*
  - ***Training data inference attacks / Sample reconstruction****, in which the adversary aims to fully reconstruct the actual data used during training of the shared model.*

  Privacy inference attacks make use of the model parameters (or gradients) and/or the output of the model, thus they can be applied both during training and prediction phase.

- ***Evasion (Prediction phase)***

  Evasion is a non-invasive attack aiming to construct adversarial examples that can break through the systems robustness and lead the model to false predictions. This type of attack requires only access to the final model and its outputs and thus exists during the prediction phase.

2.6.1.2 Defences

- **Defences against poisoning attacks**

  - **Untargeted attacks – Byzantine attacks defences**

    Robust Byzantine-resilient algorithms can converge even in a scenario where a large volume of adversarial attacks is involved. Such attempts to defend include AUROR [30] algorithm which assumes that most honest participants will have a similar distribution to the most important features of the model. Using this intuition clusters honest participants and discards updates from the outliers. Outliers are the participants that exceed a pre-defined threshold distance. Krum [31] measures the Euclidean distance between the collected updates and their mean and those with the greater distance are discarded. Bulyan [32] presents a similar approach but in this case also computes the trimmed media of the collected updates after the outliers have been removed. Robust Aggregation for Federated Learning [33] utilizes the geometric median to aggregate the updates. Other attempts focus on different mechanisms to filter out Byzantine participants like Su et al [34] which use the filtering procedure of Steinhardt. In Zeno [35] a score for each model update is calculated to show the performance gains for each update considering gradients with higher scores to come from honest

participants. While in contrast to previous methods in Byzantine-Robust Stochastic Aggregation (RSA) [36] a term is added to objective function in order robustify the aggregation.

- **Targeted attacks – Backdoor attacks defenses**
  Defenses for targeted attacks can be two-fold: methods to detect if a backdoor exists in a model or if a sample is a backdoor and methods to remove the backdoor.

  Detection
  These algorithms are based on the intuition that statistical differences exist between latent representations of backdoor triggers and benign samples [37][38][39].
  [Spectral signatures in backdoor attacks]
  [Targeted backdoor attacks on deep learning systems using data poisoning]

  Erasing
  Backdoor defense mechanism includes methods like in [40], where backdoors are mitigated by clipping the norm and adding noise to the updates. In FoolsGold [41] defends against Sybil attacks utilizing similarity methods to differentiate sybils to benign participants. In Lastly Certifiably Robust Federated Learning (CRFL) [42] framework has been presented to certifiably train robust FL models.

- **Defenses against privacy attacks**
  Defense against privacy inference attacks can be a daunting task in the context of Federated Learning due to the special characteristics of such systems like data heterogeneity, network connectivity etc. The three main categories of these defense mechanisms are the following:

  - **Homo-morphic encryption (HE)**
    These algorithms allow the computation to be done on encrypted data. An additive homomorphic scheme has been used by [43] to secure data through encryption against an adversary and participate in federated training. Homomorphic encryption is widely used in distributed learning settings [44].

  - **Secure Multiparty Computation (SMC)**
    Secure multiparty computation in [45] provides the protocol to be followed to implement computations between participants that don't trust each other. It was applied in the federated learning setting and ensured security even in the existence of malicious actors [46].

  - **Differential Privacy (DP)**
    Differential privacy is a mathematical concept which ensures that no useful information can be deduced about the existence of a specific sample in the training dataset used for training the model. Several applications in the Federated Learning setting have adopted differential privacy mechanisms to ensure privacy in several variations. The three main variations are centralized differential privacy (CDP) [47], local differential privacy [48],[49],[50] and distributed differential privacy (DDP) [51][52].

2.6.1.3 Challenges

It is evident that security and privacy preservation in Federated Learning systems is not easy, it is a complex problem which involves several challenges that require careful consideration and expertise to address it effectively. An open challenge identified by the literature is the use of large models and Byzantine-robust aggregation algorithm. The large size of a model allows adversaries to add small but effective changes and remain undetected. In [x] the idea of sharing less sensitive information is proposed to enhance robustness. Another unexplored area is the applicability and adaptability of attacks and defenses in the field of heterogeneous federated learning. Federated Learning aims to satisfy multiple and often contrary goals which include fast convergence, generalization, communication efficiency, privacy, robustness. Existing works are not able to effectively address all the aspects of FL, usually a trade-off between two or more goals is introduced. In addition, it is impossible to apply a defensive mechanism without incurring some type of cost to the system. So, two open challenges emerge, the need for better defense algorithm that address all the aspects of FL and the necessary tools to calculate utility and cost trade-offs as well as trade-offs between different aspects of the system. Lastly data and behavior auditing are usually overlooked in a Federated Learning system even though many attacks can be mitigated before compromising the training procedure. Data quality assessment methods and client trustworthiness measurements can be devised to defend against adversarial clients and servers.

The federated learning algorithm proposed in the context of the T4.3 should not only be asynchronous but also ensure security and privacy of the involved clients. Thus, privacy preservation and security methods will be integrated to the asynchronous FL algorithm to mitigate possible threats and contribute to the implementation of a robust and secure system able to learn an accurate model even in the presence of malicious actors. Details will be included in the relevant deliverable, D4.3 Asynchronous learning of predictive models of intelligent agents. From the literature in the field, it is evident that many threats are associated with data tampering or low data quality. The data auditing module proposed in the context of the T4.4 will also contribute towards the security and robustness of the federated learning system. Specific details will be included in the D4.4 Federated data infra & toolkit for data-driven model.

2.7. Security-by-Design approaches in systems, IOT, 5G and the industrial environment

A simple definition of security-by-design is provided in [53], which refers to security-by-design as building technology products in a way that “reasonably protects against malicious cyber actors successfully gaining access to devices, data, and connected infrastructure”. The same white paper also defines security-by-default, which refers to technology products that have defences against common threats and vulnerabilities out of the box, without additional costs to the end-user. The following section presents a review on two relevant whitepapers along with the ISO/IEC TR 30164 and the ISO/IEC TS 19249 standards. Appendix B Mapping of ISO/IEC TS 19249 Security-by-Design Principles to other standards" provides a mapping of ISO/IEC TS 19249 Security-by-Design Principles to 58 principles and recommendations of other standards and whitepapers, to allow the project participants to easily identify the approaches and principles relevant to their technologies and applications.

2.7.1. ISO/IEC TR 29148

ISO/IEC/IEEE 29148 [54] deals with systems and software engineering and more specific with requirements engineering. It contains a section for the definition of system security requirements, according to which system security requirements should include both the operational security requirements of the system examined along with any system requirements arising from the physical space where the system is operating. These security requirements should include factors that would protect the system from accidental or malicious access, use, modification, destruction, or unauthorized disclosure.

The standard provides the following requirements in this area as an example of system requirements: a) utilization of cryptographic techniques, b) logging security related incidents and related historical data, c) isolating functions to different modules of the system, d) restrict communications between only between the necessary parts of the system, e) critical variables of the system should be checked for their data integrity and f) data privacy should be assured.

2.7.2. CISA 2023

This short whitepaper [53] was published on April 2023 by multiple international cybersecurity related organisms. It provides definitions for Security by Design and by Default and offers a guide for technology manufacturers to ensure security of their products in the form of short list of suggestions.

The suggestions concerning security-by-design can be split to two categories. The first category involves security related decisions concerning the tools and hardware that will be used for the system and includes but is not limited to principles such as the use of:

- Memory safe programming languages
- Secure Hardware Foundation and Secure Software Components
- Web template frameworks with automatic escaping of user input
- Parameterized queries against databases.

The second category concerns security related decisions concerning the procedures and practices that will be followed during the creation of the system and includes but is not limited to principles such as:

- Performing static and dynamic application security testing
- Code reviewing
- Checking CVE completeness against know databases
- Creation of Software Bill of Materials (SBOM)
- Satisfaction of a predefined list of good security practices such as the Cyber Performance Goals [56]

Additionally, more points are suggested for secure-by-default /out of the box technologies and consequently systems:

- Elimination of default passwords
- Implementation of single sign on approach
- Provision secure logging

- Use of Software Authorization Profiles
- Prefer forward-looking security over backwards compatibility
- Track and reduce the so-called hardening guide size. A hardening guide is a document or set of instructions that provides comprehensive guidelines for enhancing the security of a computer system or a product etc.
- Consider the user experience consequences of security settings

Secure-by-default products all not all encompassing against threats and vulnerabilities. However, a secure configuration should be the default baseline and its' complexity should be an issue solved by the system, software or product engineer. Products and technologies designed under these principles should to make the end user aware that when a deviation from the default security settings occurs, the defence of the product is compromised unless additional compensating controls are implemented.

2.7.3. OWASP developer guide

The Open Worldwide Application Security Project (OWASP) produces the OWASP Developer guide [55] which includes twelve principles towards secure-by-design development. By adhering to the following security-by-design principles, organizations can foster a proactive and robust security posture within the industrial environment, protect critical assets and processes and effectively mitigate security risks and threats.

These principles can be split to three different categories i.e., principles that involve a) System Architecture and design, b) Access control and privilege management and c) Error and Exceptions handling.

There are seven principles that belong to the System Architecture and design category: The first principle of this category is called the *No Security Guarantee*. According to this principle there no application or system that is completely secure against all attacks. Consequently, the aim of security-by-design in not to design a foolproof system but a system that is hard to launch a successful attack. The next principle is called *Economy of Mechanism* and states that if there are multiple implementations then the simplest and most easily understood implementation should be chosen. The likelihood of vulnerabilities increases when a) the complexity of the software architectural design increases and b) when it is hard to follow or review the underlying code. Simplifying and making the software design and implementation easily comprehensible helps to decrease the software's attack surface.

The *Open Design security* principle emphasizes the separation of the design's implementation details from the design itself. This approach enables the design to remain open and transparent while keeping the implementation confidential if needed. It stands in contrast to the concept of security by obscurity, where software security relies on concealing the design itself. By employing the open design concept in software architecture, the review of the design does not lead to the compromise of the software's safeguards. This ensures that even if the design is openly accessible, the security measures remain intact.

The security principle of *Least Common Mechanisms* prohibits the sharing of mechanisms among users or processes with different privilege levels. This prevents potential vulnerabilities that may arise from such shared access. Psychological acceptability is a principle that aims to maximize the adoption and utilization of security functionality within software. It advocates that this achieved by ensuring that the security features are user-friendly and transparent. Security mechanisms should not excessively hinder resource access, as this might lead users to seek ways to bypass the mechanism and compromise

security.
According to the *Weakest Link* security principle, the resilience of software against hacker attempts depends heavily on safeguarding its most vulnerable components, such as the code, services, or interfaces.
Finally, the security principle of *Leveraging Existing Components* focuses on minimizing the attack surface and avoiding the introduction of new vulnerabilities. This is accomplished by promoting the reuse of proven software components, code, and functionality. Utilizing existing components, which have undergone testing and have available security patches, enhances overall security. Open-source components, benefiting from the contributions of many developers, are likely to be even more secure.

The four principles that belong to the Access control and privilege management category are Defence in Depth, Least Privilege, Separation of privilege and Complete Mediation. The first principle of this category also called *Layered Defence*, involves an architectural approach where is approach were single points of complete compromise are either eliminated or mitigated by incorporating of a series or multiple layers of security safeguards and risk-mitigation countermeasures. If a single layer of defence proves insufficient, having diverse defensive strategies in place allows subsequent layers of protection to come into play. This multi-layered approach ensures that if one defence is breached, another layer can step in to prevent a complete compromise. Moreover, if the second layer is also bypassed, the subsequent layer has the potential to thwart the exploit.

The principle of *Least Privilege* entails providing user or a process with the absolute minimum level of access rights required to carry out a specific task. Moreover, this access should only be granted for the exact duration necessary to complete the assigned operation. By adhering to this principle, the potential damage resulting from a system compromise is mitigated, as it restricts an attacker's ability to escalate privileges either horizontally or vertically. Achieving this principle requires establishing precise and appropriate granularity of privileges and permissions.

The third principle is referred to as *Separation of privilege* or *Separation of Duties*. It mandates a function design where that the accomplishment of a specific task necessitates the satisfaction of two or more distinct conditions. These conditions, when considered independently, are inadequate for completing the task on their own. The applications of this principle are numerous, encompassing scenarios such as curbing the potential harm that may arise from a malicious insider, as well as restraining an event of privilege escalation.

The *Complete Mediation* principle enforces checking for authorization (rights and privileges) upon every request for some object, ensuring that authority is not circumvented in subsequent requests of an object by a subject. This means, that all access requests by a subject for an object are always completely mediated every time.

Finally, the Error and Exceptions handling category only contains the *Fail-Safe* principle which aims to maintain confidentiality, integrity and availability of the system even when an error condition is detected. The occurrence of error conditions can stem from various sources, such as deliberate attacks or flaws in design and implementation. Regardless of the cause, it is imperative for the system or applications to prioritize a secure state over an unsafe one. For instance, unless explicit access is granted to a subject, it should be automatically denied access to the associated object by default. By adhering to this principle of failing safe, the software exhibits greater resiliency, enabling the system or application to swiftly recover from design or implementation issues.

2.7.4. ISO/IEC TR 19249

ISO/IEC TR 19249 [57] offers specific principles to create a secure architecture that allows as basis to enforce specific properties that a system is expected to effectively enforce and additionally has the ability to be robust against attacks that the system faces while operating in its intended operational environment. The architecture should have the ability to block attacks by design, while providing tools that detect any attacks not blocked and limit or mitigate the effects of such attacks. To achieve these aims, the standard provides five architectural and five design principles that should followed towards the creation of a secure architecture. The remainder of the subsection briefly presents these principles.

3.7.4.1 Architectural principles

The first architectural principle involves *Domain Separation*. In the context of the standards, a domain is a concept of grouping system components, data and applications into discreet entities which can be managed separately during the assignment of privileges and other security related attributes e.g., configuration. The application or system designer should separate the components of an application or a system with common security relevant attributes e.g., access to files, from other components with different security relevant attributes. The tasks inside each domain should be executed with the least privileges. Different domains should be isolated with each-other, while inter-domain interactions should be controlled and occur through well-defined interfaces. Such an approach simplifies error detection, limits error propagation, and enables the implementation of a defence-in-depth strategy.

The second principle, concerns *Layering* which is an architectural approach where the functions offered by an application are offered in hierarchical manner: One layer can used functions of the next lower layer and offers its' functions to the next higher layer. In a layered architecture the lowest layer provides very basic simple functions that are then used by the next higher layer to implement more complex functions, then used by the next layer to provide even more complex functions, and so on. Each layer provides access to lower layer functions provided by lower layer by abstracting them and does not allow a layer to bypass the functions of the next lower layer and use functions provided by layers lower in the hierarchy. This attribute allows each layer to implement its own security policy and protects functions implemented in lower layers from being tampered with or bypassed by functions implemented in higher layers.

The third principle entails the *Encapsulation* of the various objects of an application or system. Each object has specific function to access, manipulate and manage it which can be described as agent to control it. The functions assigned to this agent include the security related ones e.g., those responsible for access control, security audits, integrity protection or data encryption. These agents should be separate from untrusted objects or entities participating in the system, while mechanisms that ensure they cannot be bypassed or tampered with, should be enforced.

The fourth principle is *Redundancy* which involves the creation of an architecture that even in the case of errors allows the recovery of devices, communication links, functions and data even in the event of errors. It can be utilized as a mechanism that minimizes effects of attacks where redundant mechanisms are used as backup. Redundant systems allow for automatic recovery in the case of errors or flaws, provided that a functionality exists that is able to detect potential errors and flaws as fast as possible and determines which of system elements still operates correctly. Additionally, Redundancy can be utilized to improve the availability of elements of a system with limited resource availability or

high usage rate. To achieve this a management function that distributes the load among the redundant elements is needed. The distribution can be achieved by monitoring KPIs and trying to optimize the e.g., ensure that throughput is maximized or the maximum wait time is minimized.

The final principle is *Virtualization*, i.e. the emulation of a real or a logical device, application, processor, system on a different real or a logical device, processor, system. Virtualization allows a software version of component to replace anon-virtualized component. This software can then be either executed on the layer that provides the virtualization or abstract complex functionalities of real components. Virtualization enables for separation of components providing additional access control and the implementation of additional security functionalities e.g., encryption.

3.7.4.2 Design principles

The first design principle involves the assignment of *Least Privilege*. i.e., allowing an application, component or user the minimal principles that are required to perform the task they are assigned to. Application of the Least Privilege principle requires the definition of a set of privileges with different granularities to be assigned to the entities of a system or an application, either statically or dynamic, which allow or restrict access to data or functions. The set of privileges can then be used for fine-grained administrative roles, where each type of administrative action is bound to a dedicated privilege that can be assigned to users. Minimizing the privileges granted, supports the detection of errors or attempts of untrusted users and code attempts to access resources not normally available to that entity.

The second principle relates to the *Minimization of the Attack surface* of an application or a system. By attack surface, the standard refers to the set of interfaces and services that can be used as basis of an attack by untrusted and potentially malicious actors. The size of this set is correlated to the probability of an event of an attack. Minimizing the attack requires an in-depth analysis of the interfaces are part of the attack surface, the kind of attacks the operator can expect at those interfaces and the skills and motivation of the hostile acters can be expected to have. Such an analysis can then enable an informed effort to minimize the number of interfaces a potential attacker has access to, de-activating interfaces not needed, reduce the complexity of those interfaces and monitor their use to detect potential misuse or attack attempts.

The third design principle involves the use of *Centralized parameter validation*. Flawed or incomplete validation of system parameters can be the cause of multiple known vulnerabilities. Centralized parameter validation should ensure that the parameters to critical functions are always validated. This should be achieved using a common set of validation functions which allow for a single comprehensive analysis that ensures correctness and completeness. Centralized validation tools include firewalls and protocol validators.

The fourth principle concerns the *Offering of security services in a Generalized and Centralized approach*. Initially, the use of security functions provided by underlying systems or platforms e.g., the OS should be prioritized. The remaining gaps should be addressed by security related functions that are designed to be generalizable i.e., they can be reused by multiple components of an application or system. Such functions can be provided by a centralized component which reduces overall complexity and allows easier monitoring of said functions by the operator of the system. The standard offers the following examples of such services:

- user identification and authentication, user privilege management

- access control
- audit record collection and evaluation
- cryptographic services
- security monitoring and management

The final design principal concerns *Preparation for Error and Exception Handling*: Every system is expected to face errors or failures. Systems designed according to this principle, should have mechanisms that allow the detection of error or critical events. The results of the detection should either reported or used as input to mechanism that allow for the automation correction of the error or the mitigation of its' impact to the system. The standard proposes the following series of actions toward efficient error and exception handling:

a) Identify a list of possible errors and exceptions that require specific handling during the system/application requirements definition

b) Specify the functionalities required to detect the errors or exceptions included in the list

c) Specify the functions that will handle each error or exceptions

d) Specify the steps of the process required to either resume normal operation or to shut down dedicated functions or shut the whole system/product gracefully down after an error or an exception state occurs.

3. DevOps Methodology

This section discusses the DevOps methodology, its different phases and how security controls can be added to this methodology. The purpose of creating the DevSecOps methodology by moving security to the beginning of the software development phases and applying different controls throughout the software lifecycle is focused on secure software development. Additionally, an overview of DevSecOps in CPS is provided.

3.1. DevOps Definition

Historically, the lack of cooperation among the development and operations teams in software production often resulted in facing a lot of challenges along the software development lifecycle. Hence, the plan of deploying so many changes at once leads to very hard forensics processing on identifying what, where and why are located those bugs that crashes the new release available.

This is where DevOps came into play. The term coined by Patrick Debois, in October 2009 [60] is about fast, flexible development and provisioning of business processes, which by efficiently integrating development, delivery, and operations, facilitates a lean, fluid connection of these traditionally separated silos [61]. The most consolidated definition of DevOps [62] is: "DevOps is a collaborative and multidisciplinary effort within an organization to automate continuous delivery of new software versions, while guaranteeing their correctness and reliability".

DevOps integrates the two worlds of development and operations, using automated development, deployment, and infrastructure monitoring. It is an organizational shift in which, instead of distributed siloed groups performing functions separately, cross-functional teams work on continuous operational

feature deliveries. This approach helps to deliver value faster and continuously, reducing problems due to miscommunication between team members, and accelerating problem resolution.

The following are the 4 fundamental principles of DevOps methodology:

1. **Collaboration**: between project team roles.
2. **Everything as a Code**: all assets are versioned, scripted and shared where possible.
3. **Automation**: deployment, testing, provisioning any manual or human-error-prone-process.
4. **Monitoring**: any metric in the development or operational spaces that can inform, prioritize, direct and draw policy.

3.2. DevOps Phases

There are various phases in the DevOps lifecycle. The DevOps lifecycle refers to a continuous software development process that uses DevOps best practices throughout the lifecycle of the software. It is often presented in a continuous loop. Although there are several approaches aiming to identify which are the different DevOps stages or phases, those that are most frequently adopted in DevOps culture includes eight phases: Plan, Code, Build, Test, Release, Deploy, Operate, Monitor, as presented in Figure 22.

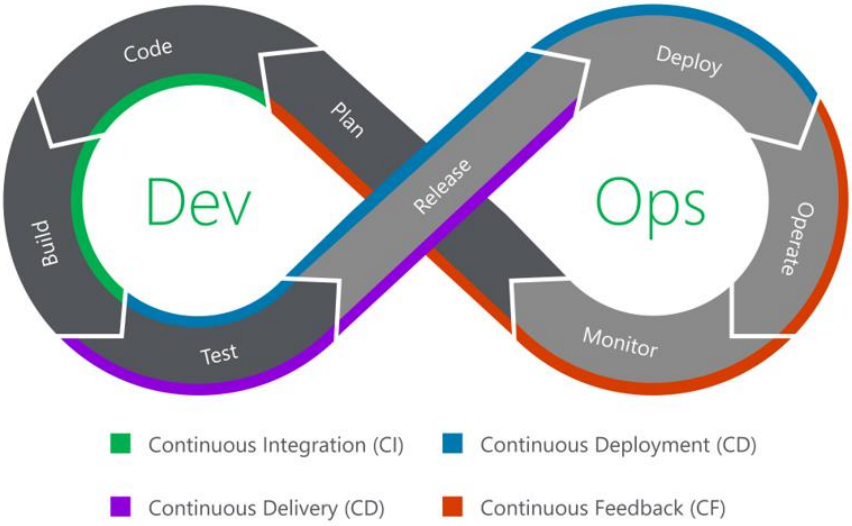


Figure 22. DevOps workflow [63]

A short description of the different phases [63] is described below:

- **Plan**: The Plan stage covers everything that happens before the developers start writing code, and it is mainly relate with the Product/Project Manager role. Requirements and feedback are gathered from stakeholders and/or customers and used to build a product roadmap to guide future development.
- **Code**: This is the phase where the developments start. In addition to the standard toolkit of a software developer, the DevOps team has a set of plugins installed in their development environments to aid the development process, including consistent code-styling and avoiding common security flaws. Resulting in developers good coding practice and in fewer failed builds.
- **Build**: Once a developer has finalized a task, the resulting code is committed to a shared code repository, typically through a pull request. Another developer then reviews these changes and once there are no issues, the pull-request is approved. Simultaneously, the pull request

triggers an automated process, which builds the codebase and runs a series of tests to identify any regressions. If the build fails, or any of the tests fail, the pull-request fails, and the developer is notified to resolve the issue.

- **Test**: Once a build succeeds, it is automatically deployed to a staging environment for deeper, out-of-band testing. Once the application is deployed to the test environment, a series of manual and automated tests are performed.
- **Release**: The Release phase is a milestone in a DevOps pipeline, as it is the point where a build is ready for deployment into the production environment. By this stage, each code change has passed a series of manual and automated tests, and the operations team can be confident that breaking issues and regressions are unlikely.
- **Deploy**: This stage is when a build is released into production. The new environment is built, and it sits alongside the existing production environment. When the new environment is ready, the hosting service points all new requests to the new environment. If at any point, an issue is found with the new build, it is just necessary to tell the hosting service to point requests back to the old environment.
- **Operate**: The new release is now live and being used by the customers. In this stage, the operations team should make sure that everything is running smoothly. It is recommended to build a way for the customers/stakeholders to provide feedback on their service.
- **Monitor**: The final phase of the DevOps cycle is to monitor the environment, sustained by the customer feedback, by collecting data and providing analytics on customer behaviour. All this information is fed back to the Product Manager and the development team to close the loop on the DevOps process. This should be considered as a DevOps continuous process.

Ideally, and with the goal of agile and rapid deployment, DevOps software shall move continually through the aforementioned eight DevOps stages in an infinity loop. In this sense, some of the previous defined stages are grouped within the so-called CI/CD concept. CI/CD are the foundational component of modern software DevOps development, as they involve the Code, Build, Test, Release and Deploy phases of the DevOps lifecycle, as shown in Figure 23.

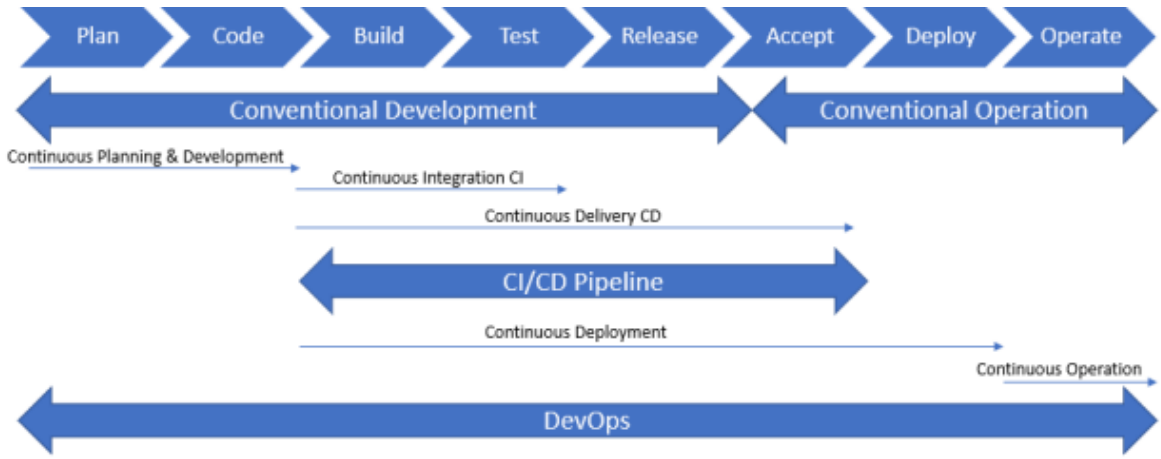


Figure 23. Continuous Integration, Continuous Development and Delivery

A breakdown of these terms and how they are related to the phases of the pipeline are described below.

- **Continuous Integration**: One of the biggest difficulties in coordinating a software development team is managing the collaboration of many developers on a single codebase. A shared code repository is key to solving this problem However, there can be issues when merging the changes made by multiple people on the same piece of code. Continuous integration aligns with the Code and Build phases of the DevOps pipeline. It generally refers to performing all of code tests, unit tests, and integration tests. By merging smaller changes more regularly, the issues become smaller and easier to manage, improving overall productivity.
- **Continuous Delivery**: It can be seen as an extension of Continuous Integration, which automates the process of deploying a new build into production. The goals of Continuous Delivery are (i) to perform automated testing on each new build in order to verify that builds are ready for release into production; (ii) to manage the automatic provisioning and configuration of deployment environments as well as testing of these environments for stability, performance, and security compliance; and (iii) to deploy a new release into production when approved and manually triggered by the organisation. As it can be seen in Figure 23, Continuous Delivery embraces the Test and Release phases of the pipeline, allowing organisations to manually trigger the release of new builds as regularly as they choose.
- **Continuous Deployment**: It is a more advanced version of Continuous Delivery. The goals are the same, but the manual step of approving new releases into production is now automated. It involves the Test, Release, and Deploy phases of the pipeline. In a Continuous Deployment model, each build which passes all the checks and balances of the pipeline are automatically deployed into the production environment.

3.3. Evolution from DevOps to DevSecOps and DevSecOps Methodology

In the past, the role of security was isolated to a specific team in the final stage of development, but those days are over. Now, in the collaborative framework of DevOps, security is a shared responsibility integrated from end to end. Security is so important that it led to coin the term “DevSecOps” to emphasize the need to build a security foundation into DevOps initiatives.

DevSecOps [64] means thinking about application and infrastructure security from the beginning and embedding DevOps with security controls providing continuous security assurance. DevSecOps is a natural extension of DevOps to include security-by-design and continuous security testing by automating some security controls in the DevOps workflow. Figure 24 presents how DevSecOps embeds security controls across the DevOps lifecycle phases.

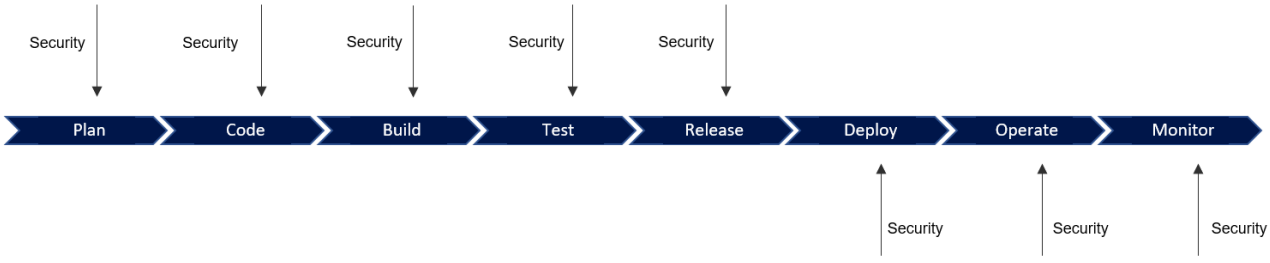


Figure 24. Security Controls in the DevSecOps workflow

The core concept of DevSecOps is that everyone is responsible for security. Management must take into consideration when defining requirements and developing schedules. Developers must incorporate it into every facet of code and specifications. Security must be tested by Quality Assurance

(QA) professionals in addition to functionality. Finally, operations teams must monitor software behaviour and respond quickly to problems.

3.3.1. DevSecOps Principles

Therefore, security awareness must be incorporated into each stage (Plan, Code, Build, Test, Release, Deploy, Operate, Monitor) [65].

- **Plan**: The planning phases involves collaboration, discussion, review, and a strategy for security analysis. Teams must conduct a security analysis and develop a schedule for security testing that specifies where, when, and how it will carry it out.
- **Code**: Developers can produce better secure code using DevSecOps technologies during the code phase. Code reviews, static code analysis, and pre-commit hooks are important code-phase security procedures. Every commit and merges automatically should start a security test or review when security technologies are directly integrated into developer's workflow.
- **Build**: In this step the primary objective of DevSecOps build tools is automated security analysis of the build output artifact. Static application software testing (SAST), unit testing, and software component analysis are crucial security procedures. Tools can implement into an existing CI/CD pipeline to automate these tests.
- **Test**: Dynamic application security testing (DAST) tools are used throughout the testing process to detect application flows such as authorization, user authentication, endpoints connected to APIs and SQL injection.
- **Release**: This stage focuses on protecting the runtime environment architecture by reviewing environment configuration values, including user access control, network firewall access, and personal data management. One of the main concerns of the release stage is the principle of least privilege (PoLP), it signifies that each program, process, and user need the minimum access to carry out its task and combines checking access tokens and API keys to limit access for the owners.
- **Deploy**: The security problems that only affect the live production system should be addressed during deployment. It is essential to carefully examine any configuration variations between the current production environment and the initial staging and development settings. The deploy stage is a good time for runtime verification tools to gather data from an active system to assess if it functions as intended.
- **Operation**: Operation teams should monitor vulnerabilities frequently. DevSecOps should use appropriate tool to protect the organization infrastructure from cyber threats.
- **Monitor**: A breach can be avoided if security is constantly being monitored for anomalies. It is essential to deploy a robust continuous monitoring tool that operates in real-time to maintain track of system performance and detect any exploits at an early stage.

The CI/CD philosophy also applies to DevSecOps methodology, by embedding security controls in this continuous loop. This is depicted thoroughly in Figure 25.

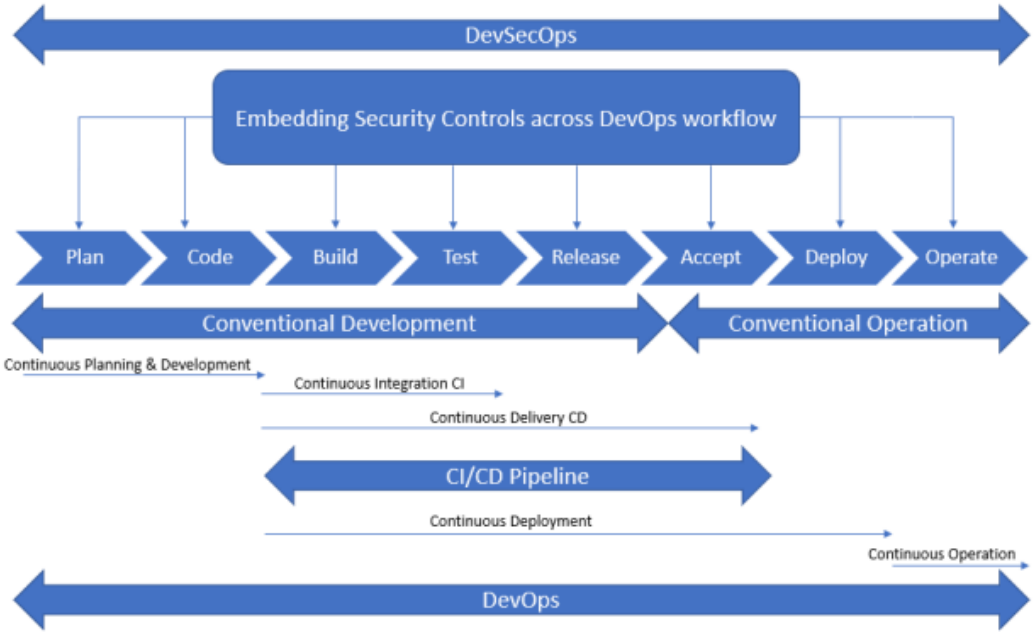


Figure 25. CI/CD in DevSecOps

3.3.2. DevSecOps Workflow

The DevSecOps principles are a set of guidelines for providing the foundation for creating different security controls in the DevSecOps continuous security model.

- **Culture Communication, Collaboration and Sharing**: The DevSecOps is a techno-cultural transformation that necessitates mind shift of the development, operations, and security people to collaborate, communicate and share information to deliver security ready applications with velocity and agility.
- **Automation**: Automation is the backbone of DevSecOps workflow implementation and enables the implementation of DevSecOps principles and practices.
- **Metrics, Measurements and Quality Assurance**: Metrics for performance and quality measurement for an automated delivery flow (i.e., agility, velocity, security, and quality). Shift Security Left: Shifting security to left advocates building security controls into the applications at earlier stages of the development cycle.
- **Security-by-Design (SbD)**: SbD is an approach to system implementation that focuses on minimizing the vulnerabilities and reducing the attack surface of the system through designing and building security controls at every stage of the system implementation.
- **Security-as-Code (SaC)**: SaC is about implementing security checks and controls into the workflow through codes.
- **Infrastructure-as-Code (IaC)** IaC treats infrastructure, both physical servers and virtual resources, as programmable unit and uses software development approach for their provisioning and configuration.
- **Compliance-as-Code (CaC)**: CaC advocates using code, to define, implement and validate security policy and controls in the workflow.

- **Adaptative Security**: An adaptive security system does not wait for incident to happen but anticipate before it can and act proactively to prevent system from any possible security breach.

3.3.3. DevSecOps Practises

The DevSecOps practices are the different activities executed along the workflow that activate security controls.

The list below describes DevSecOps Practices and target activities associated:

- Continuous Testing (CT):
  - SCA (Static Composition Analysis)
  - SAST (Static Application Security Testing)
  - Unit and integration testing
  - DAST (Vulnerability scan, PenTest, Exploit Test)
  - Acceptance, Smoke, Load and Performance Testing o IAST (Interactive Application Security Testing)
  - Infrastructure Configuration and Security Testing
- Continuous Planning, Design and Development (CPD)
  - Development Environment
  - Threat modelling and Security-by-Design
  - Source Version Control
  - Development Management
- Continuous Integration
  - Integration Automation
  - Build Automation
  - Artifact Repository
- Continuous Delivery
  - Configuration Management
  - Delivery Automation
- Continuous Deployment
  - Deployment Automation
- Continuous Operation
  - Logging, Analysis, Visualization & Notification
  - Continuous Monitoring
  - Intrusion Detection System (IDS), Intrusion Prevention System (IPS) & Security Information and Event Management
  - RASP (Runtime Application Self-Protection)
  - Infrastructure orchestration
  - Secret management
- Continuous Feedback
  - Collaboration & Communication Environment
  - Quality & Performance Measurements, Analytics, Trending & Alerting

Figure 26 also shows the leading technologies in the different categories of DevOps methodology. For instance, in testing, continuous integration, containers, cloud, and specially from the viewpoint of this document security.

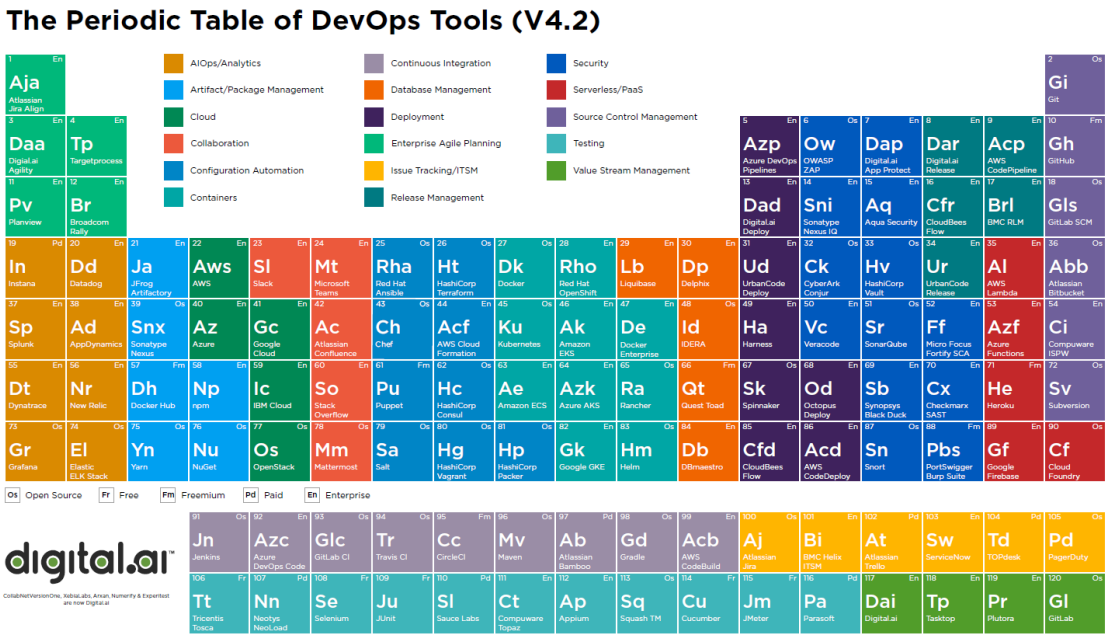


Figure 26. DevOps and DevSecOps leading technologies [66]

3.3.4. Threat modelling in DevSecOps

For providing security by design in the system, focus should be placed in the first step of the SW lifecycle and DevOps (DevSecOps) methodology, that is in the Plan phase [67]. To include security aspects and controls in this phase, threat modelling needs to be performed at this stage. This section describes what is threat modelling and identifies several methods for performing this process.

Threat modelling is a procedure for optimizing application, system or business process security by identifying objectives and vulnerabilities, and then defining countermeasures to prevent or mitigate the effects of threats to the system. There are a variety of methodologies that can be employed support cybersecurity teams in the threat modelling process.

There are several methods and models to perform the threat modelling process:

The Process for Attack Simulation and Threat Analysis (PASTA) model [70] is a risk-centric threat modelling framework first introduced in 2012. The PASTA threat model includes seven stages, as depicted in Figure 27, each with their own respective activities, with outputs that are aligned with business objectives, compliance standards and technical requirements, which makes it a model that is more strategic than it is tactical. To begin, organizations define their assets. Then each asset is walked through the seven-step process, incorporating feedback from operations, management, technology and development stakeholders. As the PASTA threat model incorporates business and impact analysis components, key organizational decision makers and staff from outside of the IT department are involved in the process. At the end of the process, a summary of threat options, severity scores and potential remediations are produced for each asset.

Figure 27. PASTA method for threat modelling [70]

The STRIDE threat modelling methodology [70], which aligns with Microsoft's default security and privacy initiative, Trustworthy Computing, and is designed to give software developers the tools needed to integrate security directly into the software design phase. The process begins with security professionals creating a data flow diagram that identifies the system's components, events, interactions and boundaries. The diagram is then overlaid with a general set of known threats using the threat types identified above. As deviations or issues are identified in the system when compared to the STRIDE model, developers can then refine the target system. This process will continue until threats are either addressed or an organization reaches its defined level of acceptable risk. Table 4 depicts some important aspects of the STRIDE threat modelling methodology.

Table 4. STRIDE Method for threat modelling [71]

	Threat	Property Violated	Threat Definition
S	Spoofing identity	Authentication	Pretending to be something or someone other than yourself
T	Tampering with data	Integrity	Modifying something on disk, network, memory or elsewhere
R	Repudiation	Non-repudiation	Claiming that you didn't do something or were not responsible; can be honest or false
I	Information disclosure	Confidentiality	Providing information to someone not authorized to access it
D	Denial of service	Availability	Exhausting resources needed to provide service
E	Elevation of privilege	Authorization	Allowing someone to do something they are not authorized to do

While each framework has their own slightly different naming convention and series of steps, the most prevalent threat models provide roughly the same logical flow and steps [71][72], as shown in Figure 28.

- **Form a team**. This team should include all stakeholders, including business owners, developers, network architects, security experts and C-level execs. A diverse team will generate a more holistic threat model.
- **Establish the scope**. Define and describe what the model covers. For example, is it focused on an application, a network or the application and the infrastructure it runs on? Create an inventory of all components and data included and map them to architecture and data flow diagrams. Each data type must be classified.
- **Determine likely threats**. For all components that are threat targets, determine where threats exist. This what-if exercise builds broad, technical and unexpected threat scenarios, including threat or attack trees to identify possible vulnerabilities or weaknesses that could lead to compromise or failure. Threat modelling tools can help automate and streamline this step.
- **Rank each threat**. Determine the level of risk each threat poses and rank them to prioritize risk mitigation A simple but effective approach is to multiply the damage potential of a threat by the likelihood of it occurring.
- **Implement mitigations**. Decide how to mitigate each threat or reduce the risk to an acceptable level. The choices are to avoid risk, transfer it, reduce it or accept it.
- **Document results**. Document all findings and actions, so future changes to the application, threat landscape and operating environment can be quickly assessed and the threat model updated.

Figure 28 Threat modelling steps [71]

3.4. Secure DevOps approach to Cyber Physical Systems

As defined earlier in section 1, CPSs are automated systems taking advantage of advanced technologies, which integrate physical world activities together with communication and computing systems. The main goal of CPSs is to interconnect multiple systems, in contradiction to conventional embedded devices, being developed primarily as separate units. CPSs collect data via internet-enabled devices like sensors, controllers or actuators. The information obtained is then utilized to develop solutions to real-world situations, in domains such as IoT, smart industry or autonomous cars. These are fast emerging areas, which utilize cutting-edge technology to address real-world difficulties at incredible speed and scale.

On the other hand, as in any system, there are threats and vulnerabilities associated with CPSs, that can be exploited by hostile entities. Attacking a system at the right time can cause massive damage depending on the system's target. Data collection can also be used by hackers to gather information for future attacks against systems or the general public. In addition, political conflicts between nations,

such as the recent US-China feud, can damage international relationships and lead to unintentional injuries or deaths on a larger scale. In order to prevent all of the above consequences, the best option is to analyse the security needs on the development and operation of complex CPSs across critical and regulated industries and determine the implications of applying a DevOps approach in a secure way to such environments.

Development, infrastructure and operations are the three major concepts that are intended to be unified under the DevOps approach for software engineering and IT administration. As the name indicates, it attempts to bring together these processes to provide a solid growth environment. In order to guarantee that their projects are bug-free and scalable, development teams collaborate closely with security and other teams. A more streamlined delivery of products, better solutions and less significant delays are the results of the common culture and openness around all project components.

A Secure DevOps approach to Cyber Physical Systems (CPS) is an essential consideration in the development and operation of these systems. CPS are systems that combine software, electronics, and physical components, and they are becoming increasingly prevalent in areas such as transportation, healthcare, and manufacturing. Due to their critical nature, securing CPS is of paramount importance. One key aspect of a Secure DevOps approach to CPS is the integration of security considerations throughout the entire development and operation process. This includes practices such as threat modelling, vulnerability assessments, and penetration testing during the development phase, and ongoing monitoring and incident response during operation. This helps to identify and mitigate potential security risks early in the development process, reducing the likelihood of successful attacks.

Even the implementation of security controls such as access controls and network segmentation can be part of the DevSecOps methodology[68]. Access controls help to ensure that only authorized personnel can access CPS systems, while network segmentation helps to isolate CPS systems from the rest of the network, reducing the potential attack surface. In addition to these technical controls, it is also important to have a comprehensive security policy in place, which outlines the roles and responsibilities of all parties involved in the development and operation of CPS systems. This includes guidelines for secure software development, incident response procedures, and regular security training for all personnel. Overall, a Secure DevOps approach to CPS is critical for ensuring the safety and reliability of these systems. By integrating security considerations throughout the development and operation process and implementing appropriate controls, organizations can reduce the risk of successful attacks and protect the systems that are critical to their operations.

3.5. DevSecOps in Zero-SWARM

This section explains why the DevOps approach, and more specifically the DevSecOps approach for a continuous model is suitable for the Zero-SWARM project, and where the methodology is further applied.

The proposed cybersecurity methodology for the project is the DevSecOps approach. This has been chosen because DevOps is a well-known industry standard for software development in a continuous, fluid and agile way. As explained before, DevSecOps approach is an evolution from DevOps to include security concerns and controls in all the phases of the SW development cycle, so that cybersecurity can be considered and included by design.

In Zero-SWARM the following tools and methodologies will be used in order to implement the designed DevSecOps methodology:

3.5.1. Threat modelling

At Zero-SWARM, it has been decided to use the PASTA threat model (following the steps presented in the Figure 27) to be able to analyse the possible vulnerabilities of the components that make up the architecture. By means of the defined steps, it is hoped to have a clear vision of which are the weak points related to cybersecurity and to improve these in the development life cycle of the new software components within the project.

Specifically, and with the aim of analysing the known and unknown vulnerabilities of the CPSoS present in the architecture, a specific cybersecurity threat test module will be developed in task T5.4, thus allowing for an exhaustive analysis of the different components that make up the Zero-SWARM architecture.

3.5.2. Tool for source version control and continuous planning design and development

Version control systems (VCS) allow tracking and managing changes to source code during the development phase of software production, so these systems play a major role in continuous planning design and development (CPD) practices. Git and SVN (Apache Subversion) are the most common open-source approaches, the first being distributed and the second centralised. Git will be used as the VCS standard in the Zero-SWARM project as it has been decided in the consortium.

Analysing the different options, we find GitHub and GitLab. GitHub offers a cloud-based service but has limited control over the repositories created and the free version is limited by participants. GitLab, in turn, is open source and although it is possible to use the service in the cloud by registering, the code can be installed on a server and a code repository can be created.

Since the project needs to have full control of the repository and access to unlimited users, it has been decided to install a GitLab server where all the code developed in the project will be stored.

Besides managing and storing versions of the developed source code, the Gitlab code repositories also include a registry for storing images of containers, packages and infrastructure definitions related to the development. Since the project will develop applications based on microservices, these images will be stored in the "Container registry" of each repository. In addition, GitLab's "Package registry" function will be used to store Helm packages (along with other code packages such as Maven, npm or PyPI).

3.5.3. Tools for build automation and continuous integration

As explained in the previous section, the project will use GitLab for version control of the code.

The CI/CD process in Gitlab is carried out by an executor, which runs a series of jobs listed in a YAML file (the *.gitlab-ci.yaml file*) and reports its final results in an easy-to-use dashboard.

The CI/CD process in Gitlab is called CI/CD pipelines, which are made up of jobs (defining the action, e.g. compile or test code) and stages (grouping a series of jobs and defining the exact time to execute them, e.g. stages containing job tests are executed after the stage that compiles the code). Jobs are the most important elements of a Gitlab pipeline as they are the ones that actually carry out the required executions. Within a stage there can be an unlimited number of jobs. Pipelines usually move on to the next stage if all the jobs in a stage are successful, otherwise the next stage is not executed, and the pipeline ends before the completion of all stages. A typical GitLab pipeline consists of four stages: build, test, stating and production.

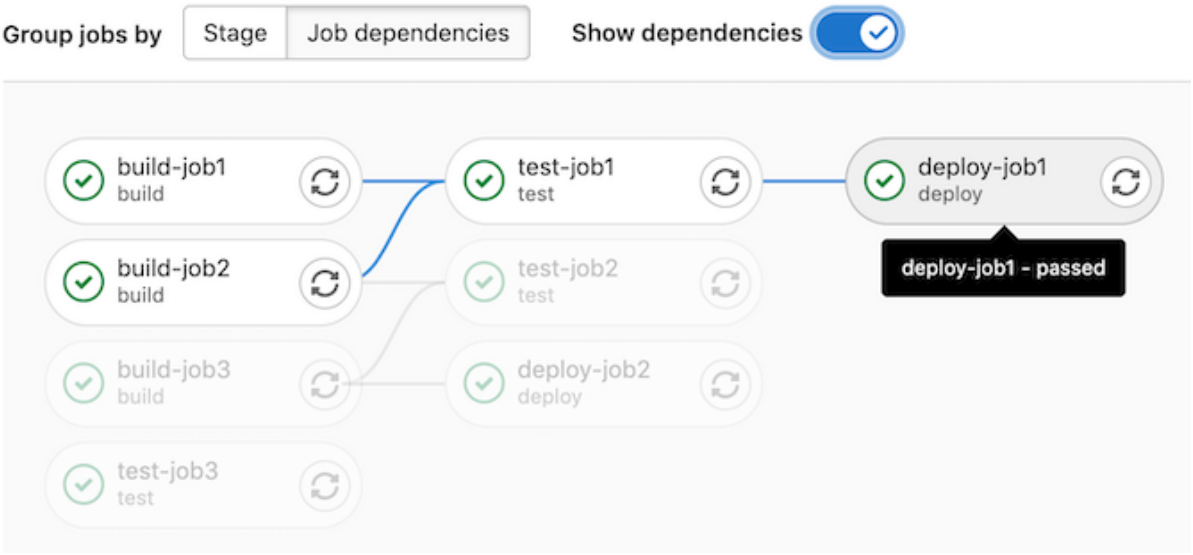


Figure 29. CI/CD pipeline in GitLab [69]

3.5.4. Tools for monitoring

Monitoring the status of software development during the different steps of the DevSecOps methodology is essential. For this purpose, on the one hand, the status of the CI/CD pipelines will be analyzed in the Gitlab dashboard.

In addition, it is necessary to analyse the security status once the code has been deployed in the production environment. In order to analyse this, in task T5.5 a cybersecurity pipeline will be developed and deployed, consisting of a SIEM and SOAR that will be in charge of analysing the security of the different components and networks and of providing a response to mitigate the attacks or threats detected in the shortest possible time, thus minimizing the damage they can cause.

4. Zero-SWARM Cybersecurity requirements

This section identifies and defines the requirements in terms of cybersecurity in the Zero-SWARM project environment, based on the study and analysis performed in the framework of WP2 through the identification of the project requirements (T2.1) and the design of the architecture (T2.2) from a viewpoint of cybersecurity and based on the research performed on the SoTA section of this document.

The following table describes the cybersecurity requirements for the Zero-SWARM project, that are also included in the D2.1 overall project requirement specification, under the "Requirements Engineering" area category. Therefore, for further information and rationale about the table format and methodology to derive into the final set of requirements please refer to "D2.1 Definition & Analysis of Trials, KPIs & GDPR Compliance" [86].

Table 5. Cybersecurity Requirements in Zero-SWARM

No	Area	Subarea	Priority (highest - 1, lowest - 3)	Overview	Description
40	Overall Security aspects	Cybersecurity & Security levels	1	The security level following IEC 62443 must be SL-2	Protection against intentional misuse by simple means with few resources, general skills and low motivation. The security available is

					the one provided and supported by OPC UA specifications.
41	Overall Security aspects	Cybersecurity & Security levels	3	The security level following IEC 62443 should be SL-3 and may be even SL-4	Protection against intentional misuse by sophisticated means with moderate resources, IACS-specific knowledge and moderate motivation
42	Overall Security aspects	DevSecOps	1	Git based Source Code Management tool must be supported.	A Git based Source Code Management tool must be installed. This repository will be used to upload the code of the different developments of the project
43	Overall Security aspects	DevSecOps	2	Deployment of a CI/CD pipeline should be supported.	In order to create an agile code development and integration in the project, a Continuous Integration / Continuous Delivery pipeline should be deployed
44	Overall Security aspects	DevSecOps	2	Testing security of code with static SAST tools and dynamic testing with DAST tools should be supported.	In order to create a security-by-design code, a Static/Dynamic code testing pipeline should be deployed in order to implement an automatic code testing in the DevOps phases
45	Overall Security aspects	Security Management	1	Security information and event management (SIEM) and Security Orchestration Automation and Response (SOAR) must be supported.	Security event monitoring in the architecture and automated response to detected incidents in order to minimise the potential harm from possible attacks.
46	Overall Security aspects	Cyber Security	2	Implementation of vertical cybersecurity services must be supported.	Cybersecurity services implemented in the architecture should provide: Confidentiality, Integrity, Availability, Authentication and Identification, Non-repudiation and Authenticity
47	Overall Security aspects	Cyber Security	1	The Anomaly Detection module shall be able to detect anomalies with high accuracy	In order to be able to detect anomalies accurately in multiple layers across a system, this module will perform deep packet inspection and behavioural analysis of the data flows.
48	Overall Security aspects	Cyber Security	1	Cyber threat countermeasure must be supported.	The countermeasure selection module shall provide a manual mode of functionality which shall allow the network operator to make decision on mitigation actions.
49	Overall Security aspects	Cyber Security	1	The anomaly detection & countermeasure selection modules shall be developed taking into account the IEC TR 62443-3-1 standard and other relevant works	These two modules need to be scalable and extensible. Which means they shall have the capability to function effectively in situations involving extensive Systems of Systems and possess the flexibility to effortlessly expand to accommodate new applications, such as the

					integration of additional data sources and cyber-security risks
50	Overall Security aspects	Cyber Security	1	The Hypothesis testing module shall use the manual mode provided by the countermeasure selection module to examine the outcomes of choosing different mitigation strategies	The Hypothesis testing module shall utilize the mitigation engine to test how different choices of the mitigation actions would affect the system
51	Overall Security aspects	Cyber Security	1	The Penetration testing module shall offer a series of attack tests that exploit known vulnerabilities of an industrial ecosystem	To ensure that concrete penetration tests will be produced for execution on the CPSoS at the development stage.

First, it must be highlighted that all the security requirements are marked as priority 1 due to their criticality in terms of impact to the system and the operation. Then, under this "Overall Security aspects" category it can be distinguished some more subareas, such as, "Cybersecurity & security Levels", "DevSecOps" and "Security Management".

"Cybersecurity & Security Levels" subcategory refers to the security requirements coming from the IEC-62443 [5] and IEC-62443-3-3 [6] Standards regarding Cybersecurity recommendations in Industrial systems and environments. "DevSecOps" subcategory refers to security requirements derived from the DevSecOps methodology introduces and explained in section 5 of this deliverable. "Security Management" subcategory refers to the security requirements linked to the overall cybersecurity aspects included in the Zero-SWARM OT/ICT architecture design, shown in Figure 30.

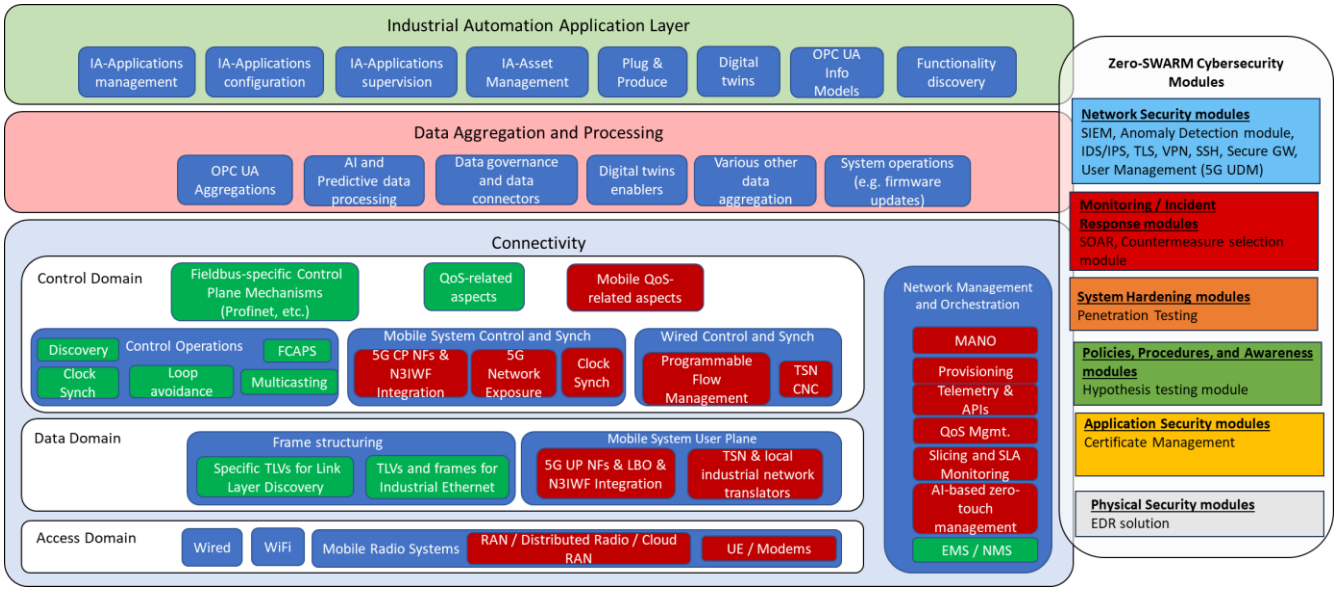


Figure 30. Zero-SWARM OT/ICT architecture [73] with Zero-SWARM Cybersecurity modules

In Figure 30, presents high level overview of the Zero-SWARM OT/ICT architecture along with the cybersecurity related modules and functionalities that will be used in the project (either developed or existing). It should be noted that the reference cybersecurity architecture of Zero-SWARM along with the details concerning the proposed Cybersecurity clusters is available on section 0.

- The anomaly detection tool will monitor and analyse cross layer and machine-to-X communications to detect anomalous traffic that might indicate adverse actions against the system, by behavioural analysis of the data flows. The tool will utilize deep learning AI and it will perform near-real time. Additionally, suspicious traffic will be further analysed to try and classify it to a specific attack type. The tool will perform analysis in multiple levels, such as the layers of the OPC-UA protocol, as well as the Network and Transport layers. The tool will be deployed strategically, across RAMI 4.0 layers and its agents will be distributed along each layer. This results in the segregation of every layer and safeguards from network scanning/probing-based attacks.
- The countermeasure selection mechanism offers a lightweight mechanism that automatically selects appropriate mitigation actions in an optimal way to countermeasure attacks faced by the network. This is achieved by using a novel Artificial Intelligence mechanism based on a Deep Neural Architecture called Pointer Networks to automatically select appropriate mitigations from a predefined action list to countermeasure any threats faced by the network while optimizing security-related KPIs. The tool will be able to receive threats detected in multiple levels, such as the layers of the OPC-UA protocol, as well as the network and transport layers.
- The Hypothesis Testing tool will allow the system operator to examine how the application of different countermeasures will affect the CPSoS. This is achieved by comparing the effect of different mitigation strategies based on the static KPIs values and their statistical difference.
- Transport Layer Security (TLS) is a cryptographic protocol ensuring secure communication over networks, safeguarding data through encryption, authentication, and integrity checks. It establishes a protected channel between parties, commonly used for securing internet data transmission.
- Secure Shell (SSH) is a cryptographic network protocol offering secure access to remote systems, enabling encrypted data exchange, remote login, and command execution. SSH ensures data privacy, authentication, and integrity during communication, commonly used for secure remote server management.
- A Virtual Private Network (VPN) establishes a secure, encrypted connection over a public network, ensuring privacy and anonymity for users. By routing internet traffic through a remote server, VPNs protect data from eavesdropping and provide access to restricted content or networks.
- Certificate management mechanisms encompass various processes and tools used to handle digital certificates securely and effectively. These mechanisms ensure the proper issuance, distribution, renewal, and revocation of certificates, as well as the protection of associated private keys.
- A security gateway is a network device or software that safeguards networks by controlling and monitoring incoming and outgoing traffic, enforcing security policies, and protecting against threats and unauthorized access. It serves as a barrier between network segments, often equipped with firewall, antivirus, intrusion detection, and VPN capabilities to ensure network security.

- Security Incident and Event Response (SIEM): The SIEM is responsible for monitoring and logging cybersecurity events on the network. These events can be collected from IDSs or agents that are in charge of monitoring traffic at different points in the networks. The events that reach the SIEM from different sources are parsed, normalised and visualised in the SIEM dashboard.
- Security Orchestration and Automated Response (SOAR): The SOAR is the responsible of the orchestration and the implementation of the cybersecurity response. The SOAR is in charge of parsing, correlating and analysing the alerts coming from other components such as the SIEM. These alerts, once analysed, can be raised to a cybersecurity case in order to provide a response and mitigate them. These responses can be manual, through the interaction of an operator with the system, or automatic.

- The Penetration Testing module is designed to conduct thorough testing, analysis and reporting within industrial automation and control systems. It will implement the IEC-62443 security standard and incorporate machine learning methods to automate various stages of the penetration testing process. This innovation empowers industries to enhance their cybersecurity readiness, protecting critical operations from potential malicious attacks and disruptions on CPSoS. Furthermore, it ensures the uninterrupted functionality of components that utilize communication protocols like MQTT, Modbus and OPC-UA.
- IDS/IPS: The IDS is the component that collects traffic from different points of the network in order to detect possible cyber-security intrusions. In addition, some IDSs are capable of implementing an orchestrated response (via SOAR) with the intention of mitigating detected cybersecurity events.

- User Management (5G UDM): The UDM manages data for access authorization, user registration, and data network profiles.

- EDR solution: Endpoint detection and response (EDR) solutions designed to automatically protect endpoint devices against threats and cyberattacks.

These modules will be further described and developed in the scope of “WP5 Standards-based toolkits for the life-cycle management of real-time CPSoS”, more concretely the framework of “T5.4 Ad-Hoc penetration and hypothesis testing plugins” and “T5.5 Anomaly detection and countermeasure selection modules”.

Finally, Table 6 and Table 7, present a mapping of the Security-by-design principles of ISO/IEC TR 19249 of related to the modules developed by or that will used in the project. ISO/IEC TS 19249 Security-by-Design Principles are high-level recommendations. To help with their application, Appendix B provides a mapping of to these principles to 58 principles and recommendations of other standards and whitepapers that provide a finer level of granularity.

Table 6. Security-by-design principles related to by modules offered by Zero-SWARM

Module Name	
Anomaly detection module	**Layering:** Adapt the security functions to the specific architecture of edge computing (ISO/IEC TR 30164), Defence in Depth (OWASP2023)

	Domain separation: Restrict communications between some areas of the programme (ISO/IEC/IEEE 29148)
	Use of least privilege: Use Control (IEC 62443-3-3)
	Centralized general security services: Detect attacks and incidents (ISO/IEC TR 30164), Record and report attacks and incidents (ISO/IEC TR 30164), Timely Response to Events (IEC 62443-3-3)
	Preparing for error and exception handling: Keep specific log or history data sets (ISO/IEC/IEEE 29148), Timely Response to Events (IEC 62443-3-3)
Countermeasure selection module	**Layering:** Adapt the security functions to the specific architecture of edge computing (ISO/IEC TR 30164), Defence in Depth (OWASP2023)
	Domain separation: Restrict communications between some areas of the programme (ISO/IEC/IEEE 29148)
	Use of least privilege: Use Control (IEC 62443-3-3)
	Centralized general security services: Record and report attacks and incidents (ISO/IEC TR 30164), Timely Response to Events (IEC 62443-3-3)
	Preparing for error and exception handling: Keep specific log or history data sets (ISO/IEC/IEEE 29148), Timely Response to Events (IEC 62443-3-3)
SIEM module for cybersecurity awareness and monitoring	**Centralized general security services:** Provide secure logging (CISA2023), Centrally managed, system wide audit trail (IEC 62443-3-3), Audit log accessibility (IEC-62443-3-3)
	Preparing for error and exception handling: Keep specific log or history data sets (ISO/IEC/IEEE 29148)
SOAR module for cybersecurity incident detection and response	**Preparing for error and exception handling:** Keep specific log or history data sets (ISO/IEC/IEEE 29148), Timely Response to Events (IEC 62443-3-3)
IPS/IDS for intrusion detection and prevention	**Preparing for error and exception handling:** Timely Response to Events (IEC 62443-3-3), Continuous monitoring (IEC 62443-3-3)
Penetration testing module	**Centralized parameter validation:** Perform Static and dynamic application security testing (CISA2023), Code reviewing (CISA2023), CVE completeness check (CISA2023), System Integrity (IEC 62443-3-3),
	Attack surface minimization: Weakest Link (OWASP2023)

Hypothesis testing Tool	**Preparing for error and exception handling:** Keep specific log or history data sets (ISO/IEC/IEEE 29148), Consider the user experience consequences of security settings (CISA2023)
	Centralized parameter validation: Perform Static and dynamic application security testing (CISA2023)

Table 7. Security-by-design principles related to by modules used in ZeroSwarm

Module Name	Security-by-design principles related to the module
TLS	**Centralized general security services:** Utilize certain cryptographic techniques (ISO/IEC/IEEE 29148)
	Attack surface minimization: Leveraging Existing Components (OWASP2023)
VPN	**Centralized general security services:** Utilize certain cryptographic techniques (ISO/IEC/IEEE 29148)
	Domain separation: Restrict communications between some areas of the programme (ISO/IEC/IEEE 29148)
	Centralized general security services: Assure data privacy (ISO/IEC/IEEE 29148)
	Attack surface minimization: Leveraging Existing Components (OWASP2023)
SSH	**Centralized general security services:** Utilize certain cryptographic techniques (ISO/IEC/IEEE 29148)
	Attack surface minimization: Leveraging Existing Components (OWASP2023)
Certificate management mechanisms	**Centralized general security services:** Use of information security management system (ISO/IEC 30141)
	Attack surface minimization: Leveraging Existing Components (OWASP2023)
Security Gateway	**Centralized general security services:** Identification and Authentication Control (IEC 62443-3-3)
	Domain separation: Restrict communications between some areas of the programme (ISO/IEC/IEEE 29148), Restricted Data Flow (IEC 62443-3-3)
	Use of least privilege: Use Control (IEC 62443-3-3)

	Attack surface minimization: Leveraging Existing Components (OWASP2023)
5G UDM	**Centralized general security services:** Identification and Authentication Control (IEC 62443-3-3)
	Use of least privilege: Use Control (IEC 62443-3-3)
	Centralized general security services: Use of information security management system (ISO/IEC 30141)
	Attack surface minimization: Leveraging Existing Components (OWASP2023)
EDR Solution	**Preparing for error and exception handling:** Timely Response to Events (IEC 62443-3-3), Continuous monitoring (IEC 62443-3-3)

5. Zero-SWARM Cybersecurity templates

This section defines a set of security templates for the project, based on the study and analysis performed in the framework of WP2 through the identification of the project requirements (T2.1) and the design of the architecture (T2.2) from a viewpoint of cybersecurity and based on the research performed on the SoTA section of this document.

These templates are aimed to be used by project partners in the implementation and integration phases of the project (for instance, during WP6 Integration, Demonstration and Validation), considering cybersecurity in the technical specifications and design.

As mentioned in the SoTA, Zero-SWARM is relying on the IEC-62443 International set of Standards [5] to address the need to design cybersecurity robustness and resilience into industrial automation and control systems (IACS), covering both organizational and technical aspects of security over the life cycle of systems. According to the IEC-62443 general organization, part IEC-62443-3-3 [6] refers to System Security Requirements and Security Levels. Therefore, the security templates and guidelines defined here are highly related to this part of the standard, to verify that the project complies with its recommendations.

5.1. Cybersecurity template description

IEC 62443-3-3 provides detailed technical control System Requirements (SRs) associated with the 7 Foundational Requirements (FRs):

- Identification and Authentication Control (IAC)
- Use Control (UC)
- System Integrity (SI)
- Data Confidentiality (DC)
- Restricted Data Flow (RDF)
- Timely Response to Events (TRE)
- Resource Availability (RA)

These requirements are intended to be used for the definition of the appropriate security capabilities at system level. For an organisation to be aligned with standard IEC 62443-3-3, it is the organisation

itself, which decides what security levels to be implemented in each context. Security Levels (SL) are characterised according to the following criteria:

- **SL 0**: does not require security specifications or protections.
- **SL 1**: requires protection against unintended incidents.
- **SL 2**: requires protection against intentional incidents, perpetrated with simple means, few resources, basic knowledge, and low motivation.
- **SL 3**: requires protection against intentional incidents, perpetrated with advanced means, sufficient resources, average knowledge, and medium motivation.
- **SL 4**: requires protection against intentional incidents, perpetrated with very advanced means, major resources, advanced knowledge, and high motivation.

This Security Level are specifically adapted and described in the framework of each particular FR.

Now, for each of the identified FRs the corresponding SRs are described, and the selected SL to be implemented is established. The following table, Table 8, presents the general template.

Table 8. General cybersecurity template

Control	SL
FR 1 – Foundational Requirement name	
SR 1.1 – Security Requirement 1 name	
SR 1.2 – Security Requirement 2 name	
…	
SR 1.N – Security Requirement 2 name	

In the following subsections the cybersecurity templates for the 7 FRs are presented, where the SL for each SR should be established.

5.2. Identification and Authentication Control (IAC)

Identify and authenticate all users (humans, software processes and devices) before allowing them to access to the control system. Table 9 outlines the cybersecurity template for Identification and Authentication Control.

- **SL 1** - Identify and authenticate all users (humans, software processes and devices) by mechanism which protect against casual or coincidental access by unauthenticated entities.
- **SL 2** - Identify and authenticate all users (humans, software processes and devices) by mechanism which protect against intentional unauthenticated access by entities using simple means with low resources, generic skills and low motivation.
- **SL 3** - Identify and authenticate all users (humans, software processes and devices) by mechanism which protect against intentional unauthenticated access by entities using sophisticated means with moderate resources, IACS specific skills and moderate motivation.
- **SL 4** - Identify and authenticate all users (humans, software processes and devices) by mechanism which protect against intentional unauthenticated access by entities using sophisticated means with extended resources, IACS specific skills and high motivation.

Table 9. Cybersecurity template for Identification and Authentication Control

Control	SL

FR 1 – Identification and authentication control	
SR 1.1 – Human user identification and authentication	
SR 1.2 – Software process and device identification and authentication	
SR 1.3 – Account management	
SR 1.4 – Identifier management	
SR 1.5 – Authenticator management	
SR 1.6 – Wireless access management	
SR 1.7 – Strength of password-based authentication	
SR 1.8 – Public key infrastructure (PKI) certificates	
SR 1.9 – Strength of public key authentication	
SR 1.10 – Authenticator feedback	
SR 1.11 – Unsuccessful login attempts	
SR 1.12 – System use notification	
SR 1.13 – Access via untrusted networks	

5.3. User Control (UC)

Enforce the assigned privileges of an authenticated user (human, software process or device) to perform the requested action on the IACS and monitor the use of these privileges. Table 10, outlines the cybersecurity template for User Control.

- **SL 1** - Restrict use of the IACS according to specified privileges to protect against casual or coincidental misuse.
- **SL 2** - Restrict use of the IACS according to specified privileges to protect against circumvention by entities using simple means with low resources, generics skills and low motivation.
- **SL 3** - Restrict use of the IACS according to specified privileges to protect against circumvention by entities using sophisticated means with moderate resources, IACS specific skills and moderate motivation.
- **SL 4** - Restrict use of the IACS according to specified privileges to protect against circumvention by entities using sophisticated means with extended resources, IACS specific skills and high motivation.

Table 10. Cybersecurity template for User Control

Control	**SL**
FR 2 – User control	
SR 2.1 – Authorization enforcement	
SR 2.2 – Wireless use control	
SR 2.3 – Use control for portable and mobile devices	
SR 2.4 – Mobile code	
SR 2.5 – Session lock	
SR 2.6 – Remote session termination	
SR 2.7 – Concurrent session control	
SR 2.8 – Auditable events	
SR 2.9 – Audit storage capacity	
SR 2.10 – Response to audit processing failures	
SR 2.11 – Timestamps	

SR 2.12 – Non-repudiation	

5.4. System Integrity (SI)

Ensure the integrity of the IACS to prevent unauthorized manipulation. Table 11 outlines the cybersecurity template for System Integrity.

- **SL 1** - Protect the integrity of the IACS against casual or coincidental manipulation.
- **SL 2** - Protect the integrity of the IACS against manipulation by someone using simple means with low resources, generics skills and low motivation.
- **SL 3** - Protect the integrity of the IACS against manipulation by someone using sophisticated means with moderate resources, IACS specific skills and moderate motivation.
- **SL 4** - Protect the integrity of the IACS against manipulation by someone using sophisticated means with extended resources, IACS specific skills and high motivation.

Table 11. Cybersecurity template for System Integrity

Control	SL
FR 3 – System integrity	
SR 3.1 – Communication integrity	
SR 3.2 – Malicious code protection	
SR 3.3 – Security functionality verification	
SR 3.4 – Software and information integrity	
SR 3.5 – Input validation	
SR 3.6 – Deterministic output	
SR 3.7 – Error handling	
SR 3.8 – Session integrity	
SR 3.9 – Protection of audit information	

5.5. Data Confidentiality (DC)

Ensure the confidentiality of information on communication channels and in data repositories to prevent unauthorized disclosure. Table 12 outlines the cybersecurity template for Data Confidentiality.

- **SL 1** - Prevent the unauthorized disclosure of information via eavesdropping or casual exposure.
- **SL 2** - Prevent the unauthorized disclosure of information to an entity actively searching for it using simple means with low resources, generic skills and low motivation.
- **SL 3** - Prevent the unauthorized disclosure of information to an entity actively searching for it using sophisticated means with moderate resources, IACS specific skills and moderate motivation.
- **SL 4** - Prevent the unauthorized disclosure of information to an entity actively searching for it using sophisticated means with extended resources, IACS specific skills and high motivation.

Table 12. Cybersecurity template for Data Confidentiality

Control	SL

FR 4 – Data confidentiality	
SR 4.1 – Information confidentiality	
SR 4.2 – Information persistence	
SR 4.3 – Use of cryptography	

5.6. Restricted Data Flow (RDF)

Segment the control system via zones and conduits to limit the unnecessary flow of data. Table 13 outlines the cybersecurity template fir Restricted Data Flow.

- **SL 1** - Prevent the casual or coincidental circumvention of zone and conduit segmentation.
- **SL 2** - Prevent the intended circumvention of zone and conduit segmentation by entities using simple means with low resources, generic skills and low motivation.
- **SL 3** - Prevent the intended circumvention of zone and conduit segmentation by entities using sophisticated means with moderate resources, IACS specific skills and moderate motivation.
- **SL 4** - Prevent the intended circumvention of zone and conduit segmentation by entities using sophisticated means with extended resources, IACS specific skills and high motivation.

Table 13. Cybersecurity template for Restricted Data Flow

Control	SL
FR 5 – Restricted data flow	
SR 5.1 – Network segmentation	
SR 5.2 – Zone boundary protection	
SR 5.3 – General purpose person-to-person communication restrictions	
SR 5.4 – Application partitioning	

5.7. Timely Response to Events (TRE)

Respond to security violations by notifying the proper authority, reporting needed evidence of the violation and taking timely corrective action when incidents are discovered. Table 14 outlines the cybersecurity template for Timely Response to Events.

- **SL 1** - Monitor the operation of the IACS and respond to incidents when they are discovered by collecting and providing the forensic evidence when queried.
- **SL 2** - Monitor the operation of the IACS and respond to incidents when they are discovered by actively collecting and periodically reporting forensic evidence.
- **SL 3** - Monitor the operation of the IACS and respond to incidents when they are discovered by actively collecting and pushing forensic evidence to proper authority.
- **SL 4** - Monitor the operation of the IACS and respond to incidents when they are discovered by actively collecting and pushing forensic evidence to proper authority in near real-time.

Table 14. Cybersecurity template for Timely Response to Events

Control	SL
FR 6 – Timely response to events	
SR 6.1 – Audit log accessibility	

SR 6.2 – Continuous monitoring	

5.8. Resource Availability (RA)

Ensure the availability of the control system against the degradation or denial of essential services. Table 15 outlines the cybersecurity template for Resource Availability.

- **SL 1** - Ensure that the control system operates reliably under normal production conditions and prevent DoS situations caused by the casual or coincidental actions of an entity.
- **SL 2** - Ensure that the control system operates reliably under normal and abnormal production conditions and prevent DoS situations by entities using simple means with low resources, generic skills and low motivation.
- **SL 3** - Ensure that the control system operates reliably under normal and abnormal production conditions and prevent DoS situations by entities using sophisticated means with moderate resources, IACS specific skills and moderate motivation.
- **SL 4** - Ensure that the control system operates reliably under normal and abnormal production conditions and prevent DoS situations by entities using sophisticated means with extended resources, IACS specific skills and high motivation.

Table 15. Cybersecurity template for Resource Availability

Control	SL
FR 7 – Resource availability	
SR 7.1 – Denial of service protection	
SR 7.2 – Resource management	
SR 7.3 – Control system backup	
SR 7.4 – Control system recovery and reconstitution	
SR 7.5 – Emergency power	
SR 7.6 – Network and security configuration settings	
SR 7.7 – Least functionality	
SR 7.8 – Control system component inventory	

6. Zero-SWARM reference cybersecurity architecture

The following section contains the reference cybersecurity architecture proposed by the Zero-SWARM project. Initially, a recap concerning the security levels of IEC-62443 along with the Foundational and System security requirements of the same standard. These act as the basis of the Cybersecurity architecture presented in section 6.3.

6.1. Security Levels and IEC-62443 cybersecurity template assessment description

Section 2.1 presents the most widely implemented industrial cybersecurity standards in Europe. CPS are systems that integrate computing elements with the physical components and processes. CPSoS, which are connected CPSs, are large complex systems where physical elements interact with and are controlled by many distributed and networked computing elements. In this perspective, the CPSoS

should comply with IEC-62443 standard and the Security Architecture must be adapted to obtain a functional Architecture.

Referring the IEC-62443, one of the key conceptions of industrial communication network and system design is directly relate with the by term Security Level (SL). Each SL, ranges from 1 to 4 and indicates the strength of the system security, so as higher the SL the more security is implemented. In Section 5, the definitions for the various levels of SL are presented along with the Foundational and System security requirements of IEC-6244, along with some requirement enhancements.

Table 16. Cybersecurity template

SRs and Res	SL 1	SL 2	SL 3	SL 4
FR 1 – Identification and authentication control				
SR 1.1 – Human user identification and authentication	X	X	X	X
SR 1.1 RE 1 — Unique identification and authentication		X	X	X
SR 1.1 RE 2 — Multifactor authentication for untrusted networks			X	X
SR 1.1 RE 3 — Multifactor authentication for all networks				X
SR 1.2 – Software process and device identification and authentication		X	X	X
SR 1.2 RE 1 — Unique identification and authentication			X	X
SR 1.3 – Account management	X	X	X	X
SR 1.3 RE 1 — Unified account management			X	X
SR 1.4 – Identifier management	X	X	X	X
SR 1.5 – Authenticator management	X	X	X	X
SR 1.5 RE 1 — Hardware security for software process identity credentials			X	X
SR 1.6 – Wireless access management	X	X	X	X
SR 1.6 RE 1 — Unique identification and authentication		X	X	X
SR 1.7 – Strength of password-based authentication	X	X	X	X
SR 1.7 RE 1 — Password generation and lifetime restrictions for human users			X	X
SR 1.7 RE 2 — Password lifetime restrictions for all users				X
SR 1.8 – Public key infrastructure (PKI) certificates		X	X	X
SR 1.9 – Strength of public key authentication		X	X	X
SR 1.9 RE 1 – Hardware security for public key authentication			X	X
SR 1.10 – Authenticator feedback	X	X	X	X
SR 1.11 – Unsuccessful login attempts	X	X	X	X
SR 1.12 – System use notification	X	X	X	X
SR 1.13 – Access via untrusted networks	X	X	X	X
SR 1.13 RE 1– Explicit access request approval		X	X	X
FR 2 – Use control				
SR 2.1 – Authorization enforcement	X	X	X	X
SR 2.1 RE 1 – Authorization enforcement for all users		X	X	X
SR 2.1 RE 2 – Permission mapping to roles		X	X	X
SR 2.1 RE 3 – Supervisor override			X	X
SR 2.1 RE 4 – Dual approval				X
SR 2.2 – Wireless use control	X	X	X	X
SR 2.2 RE 1 – Identify and report unauthorized wireless devices			X	X
SR 2.3 – Use control for portable and mobile devices	X	X	X	X

SR 2.3 RE 1 – Enforcement of security status of portable and mobile devices			X	X
SR 2.4 – Mobile code	X	X	X	X
SR 2.4 RE 1 – Mobile code integrity check			X	X
SR 2.5 – Session lock	X	X	X	X
SR 2.6 – Remote session termination		X	X	X
SR 2.7 – Concurrent session control			X	X
SR 2.8 – Auditable events	X	X	X	X
SR 2.8 RE 1 – Centrally managed, system-wide audit trail			X	X
SR 2.9 – Audit storage capacity	X	X	X	X
SR 2.9 RE 1 – Warn when audit record storage capacity threshold reached		X	X	X
SR 2.10 – Response to audit processing failures	X	X	X	X
SR 2.11 – Timestamps		X	X	X
SR 2.11 RE 1 – Internal time synchronization			X	X
SR 2.11 RE 2 – Protection of time source integrity				X
SR 2.12 – Non-repudiation			X	X
SR 2.12 RE 1 – Non-repudiation for all users				X
FR 3 – System integrity				
SR 3.1 – Communication integrity	X	X	X	X
SR 3.1 RE 1 – Cryptographic integrity protection			X	X
SR 3.2 – Malicious code protection	X	X	X	X
SR 3.2 RE 1 – Malicious code protection on entry and exit points		X	X	X
SR 3.2 RE 2 – Central management and reporting for malicious code protection			X	X
SR 3.3 – Security functionality verification	X	X	X	X
SR 3.3 RE 1 – Automated mechanisms for security functionality verification			X	X
SR 3.3 RE 2 – Security functionality verification during normal operation				X
SR 3.4 – Software and information integrity	X	X	X	X
SR 3.4 RE 1 – Automated notification about integrity violations			X	X
SR 3.5 – Input validation	X	X	X	X
SR 3.6 – Deterministic output	X	X	X	X
SR 3.7 – Error handling		X	X	X
SR 3.8 – Session integrity		X	X	X
SR 3.8 RE 1 – Invalidation of session IDs after session termination			X	X
SR 3.8 RE 2 – Unique session ID generation			X	X
SR 3.8 RE 3 – Randomness of session IDs				X
SR 3.9 – Protection of audit information		X	X	X
SR 3.9 RE 1 – Audit records on write-once media				X
FR 4 – Data confidentiality				
SR 4.1 – Information confidentiality	X	X	X	X
SR 4.1 RE 1 – Protection of confidentiality at rest or in transit via untrusted networks		X	X	X
SR 4.1 RE 2 – Protection of confidentiality across zone boundaries				X
SR 4.2 – Information persistence		X	X	X
SR 4.2 RE 1 – Purging of shared memory resources			X	X
SR 4.3 – Use of cryptography	X	X	X	X
FR 5 – Restricted data flow				
SR 5.1 – Network segmentation	X	X	X	X
SR 5.1 RE 1 – Physical network segmentation		X	X	X

SR 5.1 RE 2 – Independence from non-control system networks			X	X
SR 5.1 RE 3 – Logical and physical isolation of critical networks				X
SR 5.2 – Zone boundary protection	X	X	X	X
SR 5.2 RE 1 – Deny by default, allow by exception		X	X	X
SR 5.2 RE 2 – Island mode			X	X
SR 5.2 RE 3 – Fail close			X	X
SR 5.3 – General purpose person-to-person communication restrictions	X	X	X	X
SR 5.3 RE 1 – Prohibit all general-purpose person-to-person communications			X	X
SR 5.4 – Application partitioning	X	X	X	X
FR 6 – Timely response to events				
SR 6.1 – Audit log accessibility	X	X	X	X
SR 6.1 RE 1 – Programmatic access to audit logs			X	X
SR 6.2 – Continuous monitoring		X	X	X
FR 7 – Resource availability				
SR 7.1 – Denial of service protection	X	X	X	X
SR 7.1 RE 1 – Manage communication loads		X	X	X
SR 7.1 RE 2 – Limit DoS effects to other systems or networks			X	X
SR 7.2 – Resource management	X	X	X	X
SR 7.3 – Control system backup	X	X	X	X
SR 7.3 RE 1 – Backup verification		X	X	X
SR 7.3 RE 2 – Backup automation			X	X
SR 7.4 – Control system recovery and reconstitution	X	X	X	X
SR 7.5 – Emergency power	X	X	X	X
SR 7.6 – Network and security configuration settings	X	X	X	X
SR 7.6 RE 1 – Machine-readable reporting of current security settings			X	X
SR 7.7 – Least functionality	X	X	X	X
SR 7.8 – Control system component inventory		X	X	X

The Cybersecurity template (Table 16) shows which Systems Requirements (SR) and Requirement Enhancement (RE) are recommended to add as a feature of the system, depending on the Security
As described in Zero-SWARM D2.1 related cybersecurity requirements, adherence to SL-2 is designated as obligatory (requirement 33 noted as highest priority). The corresponding SL requirements for completing the cybersecurity template for IEC-62443-3-3 are visually indicated in green within the previous table. Additionally, in the event of eventually needing to increase and achieve higher security levels, compliance with SL3 and SL4 or even 4 of IEC-62443 could be also required. It is worth mentioning that SL-3 and SL4 security levels will also be pursued as low-priority requirements, (as per requirement 34 of Zero-SWARM D2.1). This will increase significatively the compliance with highest security levels and compliance of IEC-62443-3-3. These SL are represented in the table with an orange coloration. The actual SL achieved in the project testbeds and demonstrations will be presented in D6.3 “Integration, validation, specification of the trial demonstrations.v2” due in M30.

6.2. Security by design in Zero-SWARM

IEC 62443-1-1 introduces the defence in depth approach to cybersecurity: Instead of depending solely on a single security measure, it acknowledges that safeguarding an automated industrial facility necessitates the adoption of multiple cybersecurity measures, with each measure contributing to a

defensive layer. If an attacker manages to breach the initial layer, they would subsequently need to overcome subsequent layers one by one until they can reach their final target, alleviating the problem of a single-point-of failure system.

In the context of the IEC 62443 standard, the "defence in depth" approach can be split to six clusters, containing various functionalities and procedures, briefly explained below:

1. **Policies, Procedures, and Awareness**: This involves implementing and maintaining security policies and procedures, as well as conducting regular security awareness training.
2. **Physical Security**: This includes measures to prevent physical tampering or damage to the CPS.
3. **Network Security**: This includes measures such as firewalls, intrusion detection systems, and secure network architectures to protect against network-based attacks.
4. **System Hardening**: This involves reducing system vulnerabilities by applying patches, disabling unnecessary services, and implementing secure configurations.
5. **Application Security**: This includes securing custom-built and commercial software applications used in the IACS, as well as their underlying databases.
6. **Monitoring and Incident Response**: This involves monitoring the IACS for potential security incidents, responding to detected incidents, and conducting post-incident analysis to prevent future incidents.

These layers of defence provide multiple barriers to prevent an attacker from compromising the IACS, ensuring that even if one layer is breached, the overall system remains protected. This reduces the risk of a successful attack and helps to ensure the integrity, availability, and confidentiality of data and services in a CPS. Multiple aspects of these layers are covered by the Zero-SWARM Cybersecurity templates based on the seven foundational requirements of IEC-62443 and the System Requirements derived from them, presented in section 5. This is shown in Table 17.

Table 17. Defense in depth layer mapping to IEC 62443 Foundational and system requirements

Defence-in-Depth Layer	IEC 62443 Foundational Requirements	IEC 62443 System Requirements
Policies, Procedures, and Awareness	Identification and authentication control (FR1)	Account management, System use notification, Access via untrusted networks
	User Control (FR2)	Auditable events, Non-repudiation, Session lock, Timestamps, Response to audit processing failures, Concurrent session control, Wireless use control, Use control for portable and mobile devices, Mobile code, Remote session termination
	System integrity (FP3)	Malicious Code Protection, Security functionality verification, Software and information integrity, Input validation, Deterministic output, Error handling
	Data Confidentiality (FR4)	Information confidentiality, Information persistence
	Resource Availability (FR7)	Resource management, Control system backup, Control system recovery and reconstitution, Control system backup, Control system recovery and reconstitution, Network and security configuration settings, Least functionality, Control system component inventory
Physical Security	Identification and authentication control (FR1)	Human user identification and authentication, Software process and device identification and authentication
	User Control (FR2)	Non-repudiation, Use control for portable and mobile devices

	System integrity (FR3)	Software and information integrity, Deterministic output, Error handling
	Resource Availability (FR7)	Resource management, Emergency power, Control system component inventory
Network Security	Identification and authentication control (FR1)	Human user identification and authentication, Software process and device identification and authentication, Wireless access management, Strength of password-based authentication, Authenticator feedback, Unsuccessful login attempts, Access via untrusted networks
	User Control (FR2)	Session lock, Authorization enforcement, Non-repudiation, Concurrent session control, Wireless use control, Mobile code, Remote session termination
	System integrity (FR3)	Communication integrity, Security functionality verification, Software and information integrity, Deterministic output, Error handling, Session integrity
	Data Confidentiality (FR4)	Information confidentiality, Use of cryptography
	Restricted Data Flow (FR5)	Network segmentation, Zone boundary protection, General purpose person-to-person communication restrictions
	Resource Availability (FR7)	Denial of service protection, Resource management, Network and security configuration settings, Least functionality, Control system component inventory
System Hardening	Identification and authentication control (FR1)	Public key infrastructure (PKI) certificates, Strength of public key authentication,
	User Control(FR2)	Concurrent session control, Mobile code
	System Integrity (FR3)	Communication integrity, Malicious Code Protection, Security functionality verification, Software and information integrity, Input validation, Deterministic output
	Restricted Data Flow (FR5)	General purpose person-to-person communication restrictions
	Resource Availability (FR7)	Resource management, Network and security configuration settings, Least functionality, Control system component inventory
Application Security	Identification and authentication control (FR1)	Identifier management, Authenticator management
	User Control (FR2)	Session lock, Authorization enforcement, Non-repudiation, Concurrent session control, Mobile code, Remote session termination, Malicious Code Protection
	System integrity (FR3)	Communication integrity, Security functionality verification. Software and information integrity, Input validation, Deterministic output, Error handling, Session integrity
	Data Confidentiality (FR4)	Information confidentiality, Use of cryptography
	Restricted Data Flow (FR5)	Application partitioning, General purpose person-to-person communication restrictions, Application partitioning
	Resource Availability (FR7)	Resource management, Least functionality, Control system component inventory
Monitoring and Incident Response	User Control (FR2)	Timestamps, Audit storage capacity, Non-repudiation, Response to audit processing failures

	System integrity (FR3)	Malicious Code Protection, Security functionality verification, Software and information integrity, Input validation, Deterministic output, Error handling, Protection of audit information
	Data Confidentiality (FR4)	Information persistence
	Timely Response to events (FR6)	Audit log accessibility, Continuous monitoring
	Resource Availability (FR7)	Resource management, Control system backup, Control system recovery and reconstitution, Emergency power, Control system backup, Control system recovery and reconstitution, Control system component inventory

ISO/IEC TR 19249 [57] offers five architectural and five design principles to create a secure architecture that allows as basis to enforce specific properties that a system is expected to effectively enforce and additionally has the ability to be robust against attacks that the system faces while operating in its intended operational environment, as presented in section 2.7.1. In section 4, Security-by-design principles related to by modules offered or will be used by Zero-SWARM were presented. Here we map them to Zero-SWARM tasks and architectural approaches as shown in Table 18.

Table 18. ISO/IEC TR 19249 architectural and design principles mapped to Zero-SWARM modules and approaches

	Principle	Zero-SWARM module or approach covering principle
Architectural	Domain Separation	Zero-SWARM Architecture utilizes 6 domains: User Domain, Physical Entity Domain, Sensing & Controlling Domain, Operations & Management Domain, External access & Interchange Domain and finally the Applications & Services Domain. These are extensively presented in D2.2 [73].
	Layering	Layering is defined as the functions offered by an application are offered in hierarchical manner i.e. one layer can use functions of the next lower layer and offers its' functions to the next higher layer. The clusters used in the Zero-Swarm reference architecture, presented in the following section can be used in such a manner.
	Encapsulation	Separation of concerns between entities and domains is achieved at network level. This is presented in D2.2 [73]. This is verified the Zero-SWARM cybersecurity templates and more specifically: SR 5.1 – Network segmentation
	Redundancy	The redundancy principle is partially covered by the cybersecurity template for Resource Availability (Table 15) and more specifically SR 7.2 – Resource management, SR 7.3 – Control system backup, SR 7.4 – Control system recovery and reconstitution, SR 7.5 – Emergency power
	Virtualization	Task T4.4 "Federated transparent, flexible, and trustable data infrastructure and DevOps tools for continuous data-driven models" will research and propose relevant IT solutions utilizing Container orchestration platforms (Kubernetes, Docker).
Design	Least Privilege	This design principle is covered by the Zero-SWARM cybersecurity templates and more specifically: SR 1.13 – Access via untrusted networks, SR 2.2 – Wireless use control, SR 2.3 – Use control for portable and mobile devices, SR 5.3 – General purpose person-to-person communication restrictions, SR 7.7 – Least functionality

	Minimization of the Attack surface	The penetration testing module will produce a list of the system vulnerabilities so they can be addressed, and the Attack Surface will be minimized. Moreover, this design principle is covered by the Zero-SWARM cybersecurity templates and more specifically: SR 1.7 – Strength of public key authentication, SR 2.7 – Concurrent session control, SR 2.4 – Mobile code, SR 3.2 – Malicious Code Protection, SR 5.3 – General purpose person-to-person communication restrictions, SR 7.2 – Resource management, SR 7.7 – Least functionality,
	Offering of security services in a Generalized and Centralized approach	Zero-SWARM will offer multiple security services in a Generalized and Centralized approach such as Secure Gateways, Authentication and Access Control, Anomaly/Intrusion detection Systems, SIEM. SOAR etc. These are briefly presented in section 4.
	Centralized parameter validation	This design principle is covered by the Zero-SWARM cybersecurity templates and more specifically by: SR3.3 Security functionality verification, SR 3.4 - Software and information integrity, SR 3.5 - Input validation, SR 7.6 – Network and security configuration settings
	Preparation for Error and Exception Handling	This design principle is covered by the Zero-SWARM cybersecurity templates and more specifically: SR 2.10 – Response to audit processing failures, SR 3.7 – Error handling

6.3. Zero-SWARM Cybersecurity Reference Architecture

As previously described in section 2.3 and considering reference architectures for IIoT, and also best practices in cybersecurity described in different standardization approaches, cyber security services are transversal or vertical services associated to these architectures as described for example in Section 2 describing reference architectures and in particular in sections 2.3.1 and 2.3.2, which describe OpenFog Reference Architecture and the IoT A Reference Architecture.

A cybersecurity architecture will be the foundation for a CPS and CPSoS defense against cyber threats, and it will enhance significatively the protection against cyberthreats of all components of its IT infrastructure are protected. Environments that are secured by a cyber security architecture include:

- Cloud
- Networks
- IoT devices
- Endpoints
- Mobile devices

Pre-emptive threat prevention technology is the key to a modern cyber security architecture blocking sophisticated attacks before damage can be inflicted. An organization needs to be able to predict and block unknown malware, as well as known malware, to deliver consistent protection across the entire IT infrastructure.

Zero-SWARM proposes a reference architecture, that utilizes upon the functionalities and procedures shown in Figure 31. This approach is based on the needs of the defense-in-depth approach described in 6.2 and can comply with the ISO/IEC TR 19249 principles.

The Zero-Swarm cybersecurity clusters can be considered subgroups of the six IEC 62443 defense-in-depths clusters presented in section 7.2:

The **Policies, Procedures, and Awareness** cluster has two subgroups:

- User Training and Awareness (Human Layer): This layer contains tools and functionalities that allow for User cybersecurity training, End-User and Operator security awareness.
- Backup and Disaster Recovery (Resilience Layer): This layer contains the tools and procedures that handle the backup and policies along with any disaster discover plan.

The **Physical Security** cluster is covered by Endpoint Security (Device Layer) which contains functionalities such as Endpoint Detection and Response Solutions (EDR), Anti-Malware and Anti-virus solutions for the devices of the CPS. Measures that to prevent physical tampering or damage to the CPS are not considered since they are out of scope in the cybersecurity context.

The Network Security cluster has four subgroups:

- Perimeter Defense (Outer Layer): This layer contains functionalities such as firewalls for external communications, VPNs, IDS, SIEM etc.
- Network Security (Internal Layer): This layer contains functionalities that handle network segmentation, firewalls for internal communications etc.
- Data Encryption (Data Layer): This layer contains functionalities that handle data encryption.
- Cloud Layer: This layer deals with such aspects as cloud Identity and Access Management (IAM policies) or cloud security controls

System Hardening cluster is covered by the Security Patch Management (Hygiene Layer) which handles regular patch management and includes tools that handle vulnerability discovery and assessment.

Application Security cluster needs are covered by the Authentication and Access Control or Identity Layer. The Identity and Access Management layer provides identity, authentication, and authorization services for both external and internal entities, systems (server-clients), users or applications for accessing in a secure way to Zero-Swarm's systems, data, and resources.

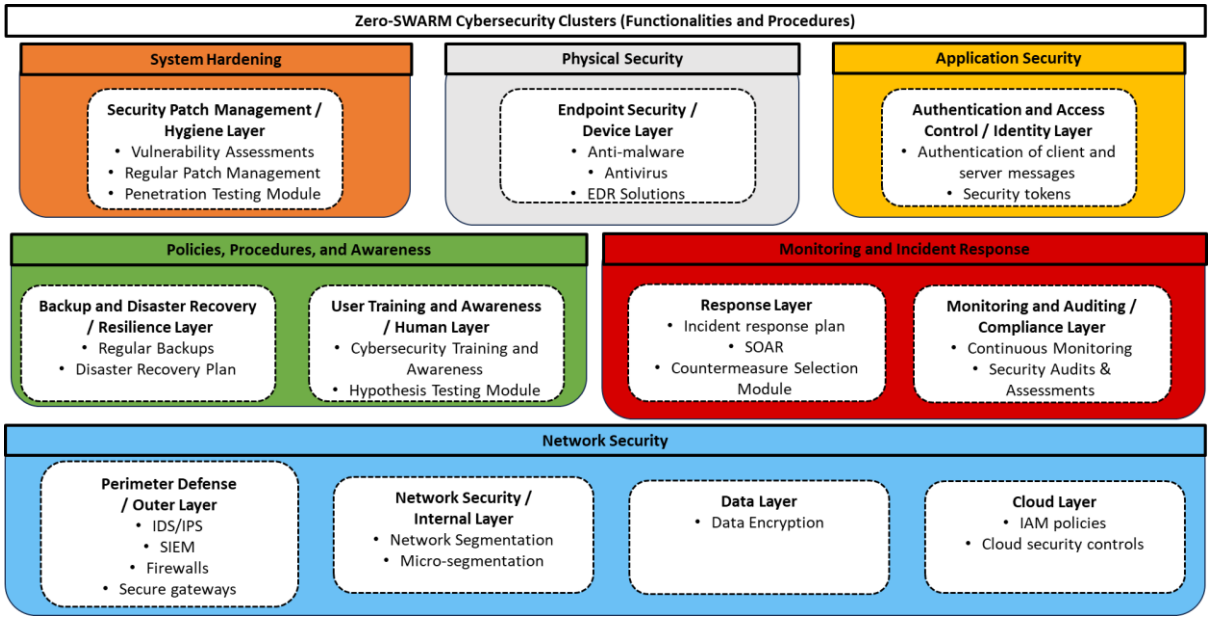


Figure 31. Zero-SWARM cybersecurity clusters

Finally, the Monitoring and Incident Response cluster needs are covered by two subgroups:

- Security Information and Event Management (SIEM): The SIEM system collects and analyses security event logs from various sources. SIEM system correlates data to detect and respond to security incidents in real-time.

- Monitoring and Auditing (Compliance Layer): This layer provides continuous cybersecurity monitoring along with functionalities to perform cybersecurity assessments and audits.

The proposed Defense-in-depth cybersecurity architecture has multiple layers of protection based on six clusters of functionalities and procedures, from the network perimeter to the human layer, in line with ISO/IEC TS 19249 and cybersecurity best practices. It also emphasizes continuous monitoring, incident response, and resilience in the event of security incidents or disasters integrating SIEM and SOAR systems in the deployment view of the architecture.

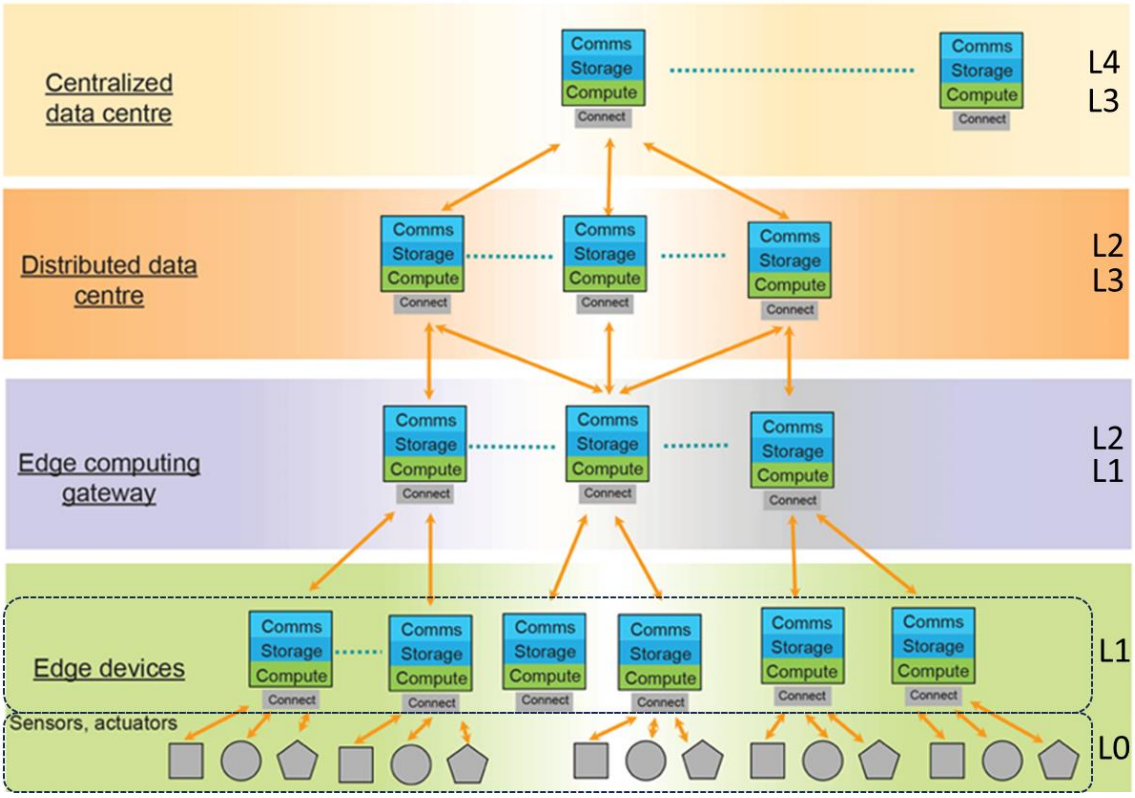


Figure 32. CPSoS deployment view / integration with responding IEC 62443 reference levels [73]

The proposed clusters are mapped to the Zero-SWARM architectural views introduced in deliverable D2.2. As mentioned in section 2.5, CPSoS are described as large complex systems where physical elements interact with and are controlled by many distributed and networked computing elements and human users. As far as the functional architecture described in zero-SWARM project the following CPSoS deployment view was introduced in deliverable D2.2, section 2.5 “Deployment view” inspired by the IEC 30164 standard [84], which describes the common concepts, terminologies, characteristics, use cases and technologies of edge computing for IoT systems applications. The proposed architectural view also can be applicable to CPSoS as it suitable for scenarios where services are deployed centrally, and the traffic volume is high such as smart manufacturing [84]. This view can be easily aligned the five levels defined by IEC 62443-1-1:

1. Level 0 (L0) is the process level, which deals with components (devices, sensors) that directly control or measure a domain specific process.
2. Level 1 (L1) manages local or basic control, e.g. controllers, I/O or Filebus communication
3. Level 2 (L2) involves Supervisory control e.g. distributed or local control systems
4. Level 3 (L3) concerns Operations/System management i.e., devices or solutions that add various functionalities or handle cybersecurity but are not critical to operate the facility in the context of L0 operations
5. L4 involves Enterprise systems e.g., Office computers and business-related systems

Levels 0 to 2 are considered trusted; level 3 is considered secure while level 4 is considered untrusted.

The Cybersecurity architecture must protect the different CPSoS deployment described and the communications between them. Each of the four different tiers can utilize or benefit from different layers, however multiple clusters are used across the different tiers as shown in Figure 33. It should be noted that the various functionalities are grouped in the same cluster based on their purpose, e.g. the Application Security cluster includes all the functions that handle authentication, and authorization services. This does not mean that a single function is used to handle secure authentication across the system.

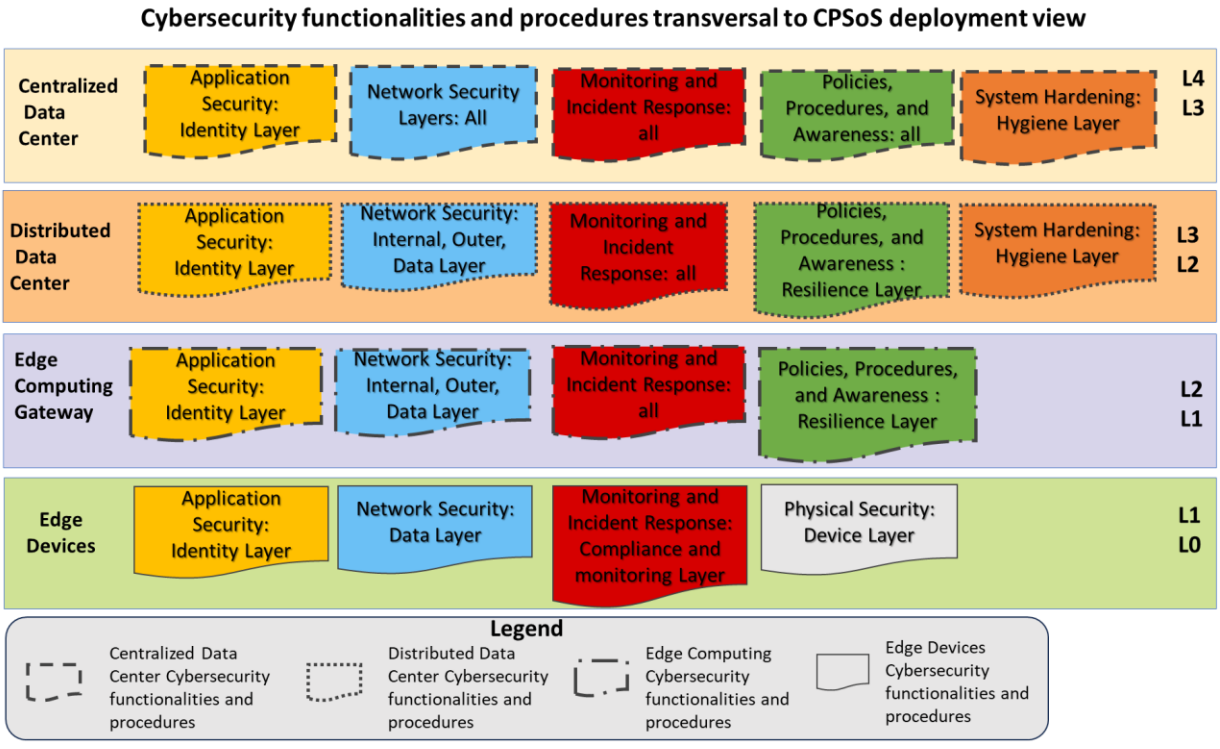


Figure 33. Cybersecurity layers' transversal to CPSoS deployment view

Most of the layers can be mapped to the initial version of the Zero-Swarm network view introduced in D2.2, based on ISO/IEC 30141. This view focuses on the internal layers of the connectivity aspect, namely, the access, data and control domains, along with the network management and orchestration. The result is shown in Figure 34.

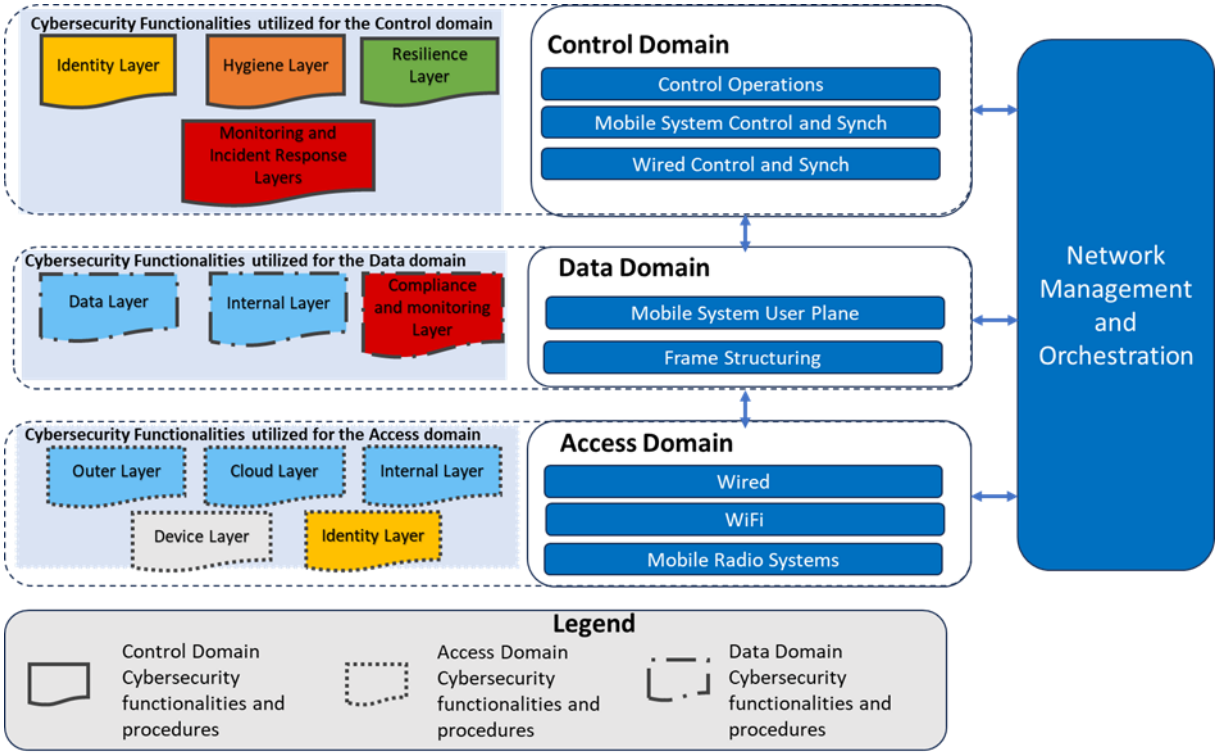


Figure 34. Zero-SWARM cybersecurity layers mapped to the initial Zero-SWARM network architectural view

Finally, the layers can be also mapped to the domain-level separation of concerns view shown in D2.2: Figure 35 depicts the position of the clusters to the cross-domain capabilities of ISO/IEC 30164. Figure 36 depicts the Zero-SWARM domain-level layers in a higher level of granularity than ISO/IEC 30164: the Zero-Swarm Industrial Automation Application Layer and the Data Aggregation and Processing Layer defined in D2.2. Both Figure 34 and Figure 36 will be updated when the final versions of the various views of projects' architecture are available and will be presented in in D6.3 "Integration, validation, specification of the trial demonstrations.v2" due in M30.

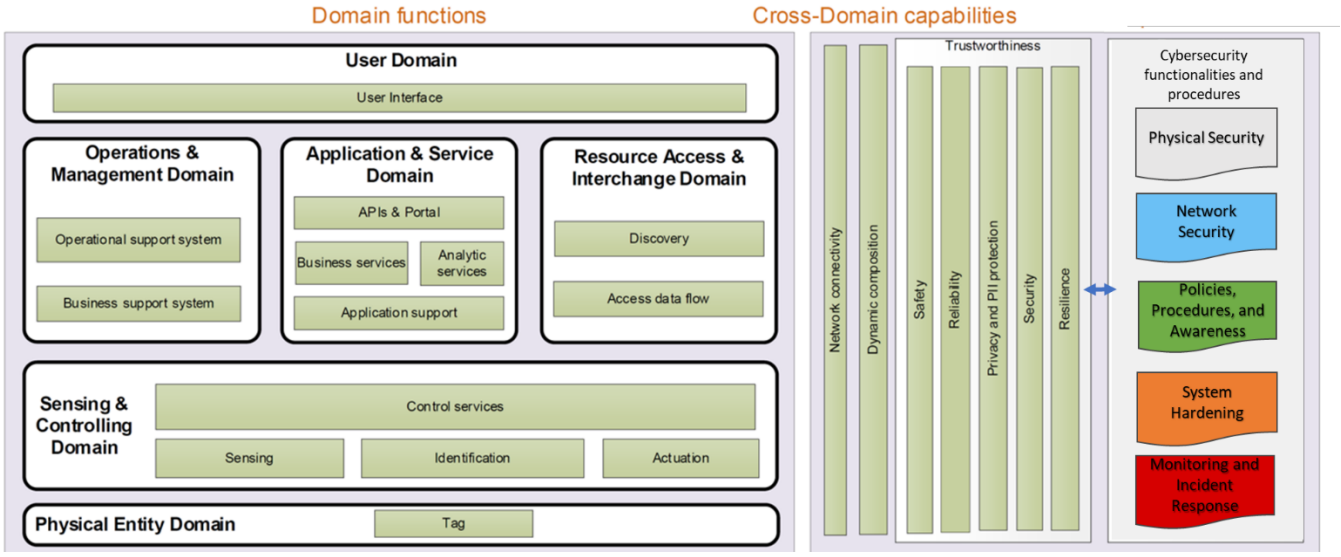


Figure 35. Zero-SWARM cybersecurity clusters belong to Cross-Domain capabilities providing "Trustworthiness" in the ISO/IEC 30164 domain-level separation of concerns view

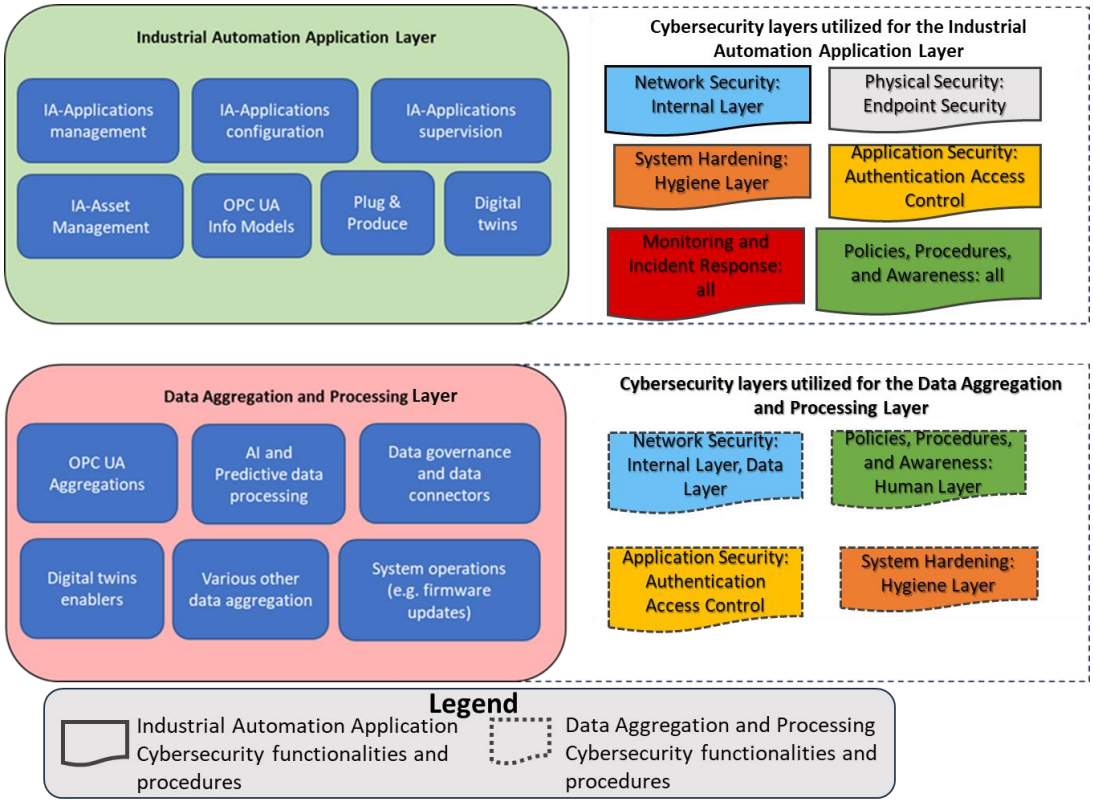


Figure 36. Zero-SWARM cybersecurity clusters mapped to the initial Zero-SWARM domain-level layers.

6.3.1. Example application of Zero-SWARM Cybersecurity Functionality and Procedures Clusters to a Zero-SWARM trial Architecture

Deliverable D5.1 "Distributed automation and information management" [85], presents a deployment view of the architecture of the CPS that will be used in the South Node trials 1 and 2. The proposed architecture, as seen in Figure 37, consists of a subset of the deployment view presented in section 6.3: It is 3-layer architecture, composed of the Cloud or IT layer, the Edge-Gateway or Edge Layer and the Edge Devices or OT layer. More details on this architecture are available on can be found in [85].

In Figure 37, we present an example of applying the Zero-SWARM Cybersecurity clusters to the trial architecture. A cluster can be applied to multiple layers of the system, for e.g., the penetration testing (Hygiene Layer), meaning that penetration testing could be easily applied to the network, the applications, the embedded systems, etc. However, this does not mean that the same penetration testing functionalities will be applied to the Edge and the OT Layer: This is denoted in the figures by different outlines used in for the cybersecurity clusters. Authentication (i.e., an Identity Layer) and Encryption (i.e., a Data Layer) are used throughout the communications of the system. Concerning Network security, an Internal Layer i.e., network segmentation is utilized in the Edge-Gateways and in the Edge-Devices Layer, while functionalities such as IDS, SIEM, firewalls (Outer Layers) are utilized in the Edge-Gateways layer. The Cloud is protected by cloud security controls, IAM policies etc. offered by the Cloud Layer. Finally, the functionalities concerning a) Backup, User Training etc. (Resilience and Human Layers), and b) Cybersecurity Countermeasures and Security audits (Response and Compliance Layers), reside in the cloud.

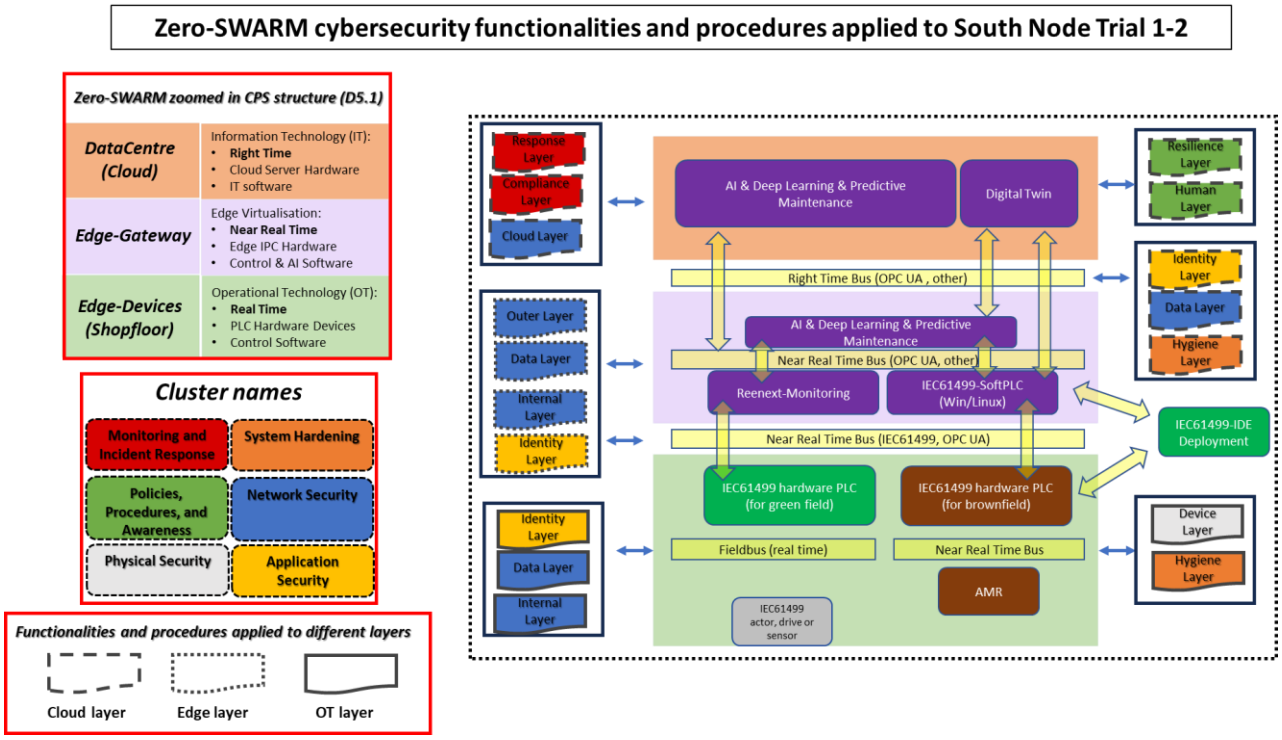


Figure 37. Zero-SWARM cybersecurity functionalities and procedures applied to South Node Trial 1-2

7. Conclusions

Deliverable D2.3 tackles the objective of Task T2.3 to introduce Cybersecurity implementation templates and a methodological approach to follow throughout the development of all Zero-SWARM components. By defining these guidelines during the design phase, the implementation of more secure components in the development phases is enabled. To reach the desired outcome of this task, several activities were preceded such as the analysis of the requirements formed in T2.1 and the architectural designs, presented in T2.2, from a cybersecurity perspective. Moreover, a state-of-the-art analysis was outlined, depicting the significance of the adoption of cybersecurity in the Industrial environment, together with a presentation of a DevOps and DevSecOps methodologies and the importance of adopting a secure DevOps approach in the Zero-SWARM project.

D2.3 provides a reference cybersecurity architecture i.e., a template solution for an architecture for the domain of IIoT. To achieve this, the five levels found in the IEC 62443 reference model along with the defence-in-depth, as described in IEC 62443-1-1 are utilized, along with architectural and design principles defined in ISO/IEC TR 19249. This architecture combined with the Zero-SWARM Cybersecurity templates defined in section 5, will help the project to follow a consistent approach for the planning, implementation, and deployment of the CPSoS under examination utilizing the guidelines of relevant standards and industry best practices. It should be noted that, while a first example of applying the reference architecture is presented in section 6.3.1, the final versions of the cybersecurity views and their application on the architectures of the trials will be presented in in D6.3 “Integration, validation, specification of the trial demonstrations.v2” due in M30.

References

[1] N. Jazdi, "Cyber physical systems in the context of Industry 4.0," *2014 IEEE International Conference on Automation, Quality and Testing, Robotics*, Cluj-Napoca, Romania, 2014, pp. 1-4, doi: 10.1109/AQTR.2014.6857843.

[2] Engell, S., Paulen, R., Reniers, M.A., Sonntag, C., Thompson, H. (2015). Core Research and Innovation Areas in Cyber-Physical Systems of Systems. In: Mousavi, M., Berger, C. (eds) Cyber Physical Systems. Design, Modeling, and Evaluation. CyPhy 2015. Lecture Notes in Computer Science(), vol 9361. Springer, Cham. https://doi.org/10.1007/978-3-319-25141-7_4

[3] Fei Tao, Qinglin Qi, Lihui Wang, A.Y.C. Nee, Digital Twins and Cyber–Physical Systems toward Smart Manufacturing and Industry 4.0: Correlation and Comparison, Engineering, Volume 5, Issue 4, 2019, Pages 653-661, ISSN 2095-8099, https://doi.org/10.1016/j.eng.2019.01.014.

[4] Cybersecurity & Infastructure Security Agency, MITRE, "Best Practices for MITRE ATT&CK Mapping",2021

[5] IEC-62443 International Standard for the Security of Industrial automation control systems

[6] IEC-62443-3-3 System security requirements and security levels

[7] DIN SPEC 27020:2020-03 requirements and Reference architecture of a security Gateway for the exchange of industry data and service

[8] ISA99 Industrial Automation & Control System Security

[9] NIST SP 800-53 Security and Privacy Controls for Information Systems and Organizations

[10] Pascal Ackerman, 'Industrial cybersecurity', 2021, PACKT, ISBN 9781800202092

[11] SANS institute, Best Practices for Secure ICS Architectures https://www.sans.org/blog/introduction-to-ics-security-part-2/

[12] McKinsey and Company, What are Industry 4.0, the Fourth Industrial Revolution, and 4IR? , 2022 https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-are-industry-4-0-the-fourth-industrial-revolution-and-4ir

[13] Kaspersky ECT, 'Threat landscape for industrial automation systems. Statistics for H2 2022' , 2023, https://ics-cert.kaspersky.com/publications/reports/2023/03/06/threat-landscape-for-industrial-automation-systems-statistics-for-h2-2022

[14] IEC-61499 Standard for distributed Automation

[15] Karsten Schweichhart, 2018, Reference Architectural Model Industry 4.0 (RAMI 4.0), https://ec.europa.eu/futurium/en/system/files/ged/a2-schweichhart-reference_architectural_model_industrie_4.0_rami_4.0.pdf

[16] Industrial Internet Reference Architecture v1.9 Jun 2019; https://www.iiconsortium.org/pdf/IIRA-v1.9.pdf

[17] OpenFog Reference Architecture, February 2017, http://site.ieee.org/denver-com/files/2017/06/OpenFog_Reference_Architecture_2_09_17-FINAL-1.pdf

[18] IoT IEEE P2413; https://standards.ieee.org/standard/2413-2019.html

[19] Arrowhead Framework; https://www.arrowhead.eu/eclipse-arrowhead/this-is-it/architecture/

[20] M. Weyrich, C. Ebert, "Reference architectures for the internet of things", IEEE Softw., 33 (1) (2016), pp. 112-116, https://www.researchgate.net/publication/288855901_Reference_Architectures_for_the_Internet_of_Things

[21] Bauer, M.; Boussard, M.; Bui, N.; Loof, J.D.; Magerkurth, C.; Meissner, S.; Nettsträter, A.; Stefa, J.; Thoma, M.; Walewski, J.W. IoT Reference Architecture. In Enabling Things to Talk; Functional-

decomposition viewpoint of the IoT-A reference architecture. Source: IoT-A, Springer: Berlin/Heidelberg, Germany, 2013; pp. 163–211, https://link.springer.com/chapter/10.1007/978-3-642-40403-0_8

[22] ENISA good practices for IoT and Smart infrastructures; https://www.enisa.europa.eu/topics/iot-and-smart-infrastructures/iot/good-practices-for-iot-and-smart-infrastructures-tool/results#IoT

[23] NIST Cybersecurity for IoT Program: https://www.nist.gov/itl/applied-cybersecurity/nist-cybersecurity-iot-program

[24] Sandy Carielli (Entrust Datacard), Ekaterina Rudina (Kaspersky Lab), Hamed Soroush (RTI) and Ron Zahavi (Microsoft), 2018, "IoT Security Maturity Model: Description and Intended Use", https://www.iiconsortium.org/pdf/SMM_Description_and_Intended_Use_2018-04-09.pdf

[25] Park, J.H.; Rathore, S.; Singh, S.K.; Salim, M.M.; Azzaoui, A.E.; Kim, T.W.; Pan, Y.; Park, J.H. A Comprehensive Survey on Core Technologies and Services for 5G Security: Taxonomies, Issues, and Solutions. Hum.-Centric Comput. Inf. Sci. 2021, 11, 3, http://hcisj.com/articles/?HCIS202111003

[26] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data." arXiv, Jan. 26, 2023. doi: 10.48550/arXiv.1602.05629.

[27] Y. Liu *et al.*, "FedVision: An Online Visual Object Detection Platform Powered by Federated Learning." arXiv, Jan. 17, 2020. doi: 10.48550/arXiv.2001.06202.

[28] J. Li, L. Lyu, X. Liu, X. Zhang, and X. Lyu, "FLEAM: A Federated Learning Empowered Architecture to Mitigate DDoS in Industrial IoT." arXiv, Dec. 10, 2021. doi: 10.48550/arXiv.2012.06150.

[29] Liu, Pengrui, Xiangrui Xu, and Wei Wang. "Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives." *Cybersecurity* 5.1 (2022): 1-19.

[30] S. Shen, S. Tople, and P. Saxena, "Auror: defending against poisoning attacks in collaborative deep learning systems," in Proceedings of the 32nd Annual Conference on Computer Security Applications. ACM, 2016, pp. 508–519.

[31] P. Blanchard, R. Guerraoui, J. Stainer et al., "Machine learning with adversaries: Byzantine tolerant gradient descent," in NeurIPS, 2017, pp. 119–129.

[32] E. M. E. Mhamdi, R. Guerraoui, and S. Rouault, "The hidden vulnerability of distributed learning in byzantium," CoRR, arXiv:1802.07927, 2018.

[33] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," arXiv preprint arXiv:1912.13445, 2019.

[34] L. Su and J. Xu, "Securing distributed machine learning in high dimensions," CoRR, arXiv:1804.10140, 2018.

[35] C. Xie, S. Koyejo, and I. Gupta, "Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance," in Proc. of ICML, 2019, pp. 6893–6901.

[36] L. Li, W. Xu, T. Chen, G. B. Giannakis, and Q. Ling, "RSA: byzantinerobust stochastic aggregation methods for distributed learning from heterogeneous datasets," in Proc. of AAAI, 2019, pp. 1544–1551.

[37] B. Chen, W. Carvalho, N. Baracaldo, H. Ludwig, B. Edwards, T. Lee, I. Molloy, and B. Srivastava, "Detecting backdoor attacks on deep neural networks by activation clustering," CoRR, arXiv:1811.03728, 2018.

[38] B. Tran, J. Li, and A. Madry, "Spectral signatures in backdoor attacks," in NeurIPS, 2018, pp. 8000–8010.

[39] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, “Targeted backdoor attacks on deep learning systems using data poisoning,” CoRR, arXiv:1712.05526, 2017.
[40] Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan, “Can you really backdoor federated learning?” arXiv preprint arXiv:1911.07963, 2019
[41] C. Fung, C. J. Yoon, and I. Beschastnikh, “The limitations of federated learning in sybil settings,” in 23rd International Symposium on Research in Attacks, Intrusions and Defenses ({RAID} 2020), 2020, pp. 301–316.
[42] C. Xie, M. Chen, P.-Y. Chen, and B. Li, “Crfl: Certifiably robust federated learning against backdoor attacks,” arXiv preprint arXiv:2106.08283, 2021.
[43] S. Hardy, W. Henecka, H. Ivey-Law, R. Nock, G. Patrini, G. Smith, and B. Thorne, “Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption,” CoRR, arXiv:1711.10677, 2017.
[44] Y. Aono, T. Hayashi, L. Wang, S. Moriai et al., “Privacy-preserving deep learning via additively homomorphic encryption,” IEEE Transactions on Information Forensics and Security, vol. 13, no. 5, pp. 1333– 1345, 2018.
[45] A. C. Yao, “Protocols for secure computations,” in SFCS, 1982, pp. 160–164.
[46] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, “Practical secure aggregation for privacy-preserving machine learning,” in CCS, 2017, pp. 1175–1191.
[47] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, “Learning differentially private recurrent language models,” in ICLR, 2018.
[48] L. Sun, J. Qian, X. Chen, and P. S. Yu, “Ldp-fl: Practical private aggregation in federated learning with local differential privacy,” arXiv preprint arXiv:2007.15789, 2020.
[49] A. Bhowmick, J. Duchi, J. Freudiger, G. Kapoor, and R. Rogers, “Protection against reconstruction and its applications in private federated learning,” CoRR, arXiv:1812.00984, 2018.
[50] L. Sun and L. Lyu, “Federated model distillation with noise-free differential privacy,” arXiv preprint arXiv:2009.05537, 2020.
[51] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, and R. Zhang, “A hybrid approach to privacy-preserving federated learning,” CoRR, arXiv:1812.03224, 2018.
[52] L. Lyu, “Lightweight crypto-assisted distributed differential privacy for privacy-preserving distributed learning,” in 2020 International Joint Conference on Neural Networks (IJCNN). IEEE, 2020, pp. 1–8.
[53] NSA, FBI, ACSC, NCSC-UK, CCCS, BSI, NCSC-NL, CERT NZ, NCSC-NZ , Shifting the Balance of Cybersecurity Risk: Principles and Approaches for Security-by-Design and –Default, 2023
[54] ISO/IEC/IEEE “International Standard - Systems and software engineering -- Life cycle processes -- Requirements engineering” , 2018
[55] OWASP, "Development Guide v3.0 (draft)", Open Worldwide Application Security Project, 2023
[56] Cybersecurity & Infastructure Security Agency, "Performance Goals, Version 1.0.1", March 2023
[57] Ref ISO/IEC TS 19249, “Information technology — Security techniques — Catalogue of architectural and design principles for secure products, systems and applications”, 2017
[58] Jon Geater (Jitsuin), Frederick Hirsch (Upham Security), Detlev Richter (TÜV SÜD), Michael Robkin (Six By Six), Ron Zahavi (Microsoft), 2022, “IoT Security Maturity Model Digital Twin Profile”, An Industry IoT Consortium and Digital Twin Consortium Whitepaper. https://www.digitaltwinconsortium.org/wp-content/uploads/sites/3/2022/06/SMM-Digital-Twin-Profile-2022-06-20.pdf

[59] Onik, Md Mehedi Hassan & KIM, Chul-Soo & Yang, Jinhong, 2019, "Personal Data Privacy Challenges of the Fourth Industrial Revolution", 635-638, 10.23919/ICACT.2019.8701932
[60] The Incredible True Story of How DevOps Gots Its Name – 2014; https://newrelic.com/blog/nerd-life/devops-name
[61] C. Ebert, G. Gallardo, J. Hernantes, and N. Serrano. 2016. DevOps. IEEE Software 33, 3 (2016), 94–100. Code: A54
[62] L. Leite, C. Rocha, F. Kon, D. Milojicic, P. Meirelles, "A survey of DevOps concepts and challenges", ACM Computing Surveys, Vol. 52, No. 6, Article 127. Nov. 2019
[63] Jakob Pennington, 2019, "The Eight Phases of a DevOps Pipeline", https://medium.com/taptuit/the-eight-phases-of-a-devops-pipeline-fda53ec9bba
[64] Rakesh Kumar, Rinkaj Goyal, "Modeling continuous security: A conceptual model for automated DevSecOps using open-source software over cloud (ADOC)", Computers & Security, Volume 97, 2020, 101967, ISSN 0167-4048, https://doi.org/10.1016/j.cose.2020.101967
[65] What are the Phases of DevSecOps?; Veritis; https://www.veritis.com/blog/what-are-the-phases-of-devsecops/#02
[66] DevOps periodic table; https://devopslatam.com/tabla-periodica-de-herramientas-devops/
[67] Cloud4c, A Complete Guide to Security by Design with DevSecOpsA Complete Guide toSecurity by Design withDevSecOps , 2022, https://www.cloud4c.com/lat/blogs/a-complete-guide-to-devsecops
[68] C. Jansen, "Developing and Operating Industrial Security Services to Mitigate Risks of Digitalization", IFAC-PapersOnLine, Volume 49, Issue 29, 2016, Pages 133-137, ISSN 2405-8963, https://doi.org/10.1016/j.ifacol.2016.11.076.
[69] Github documentation, https://docs.gitlab.com/ee/ci/pipelines/ , 2022
[70] Nataliya Shevchenko, 2018, "Threat Modeling: 12 Available Methods" https://insights.sei.cmu.edu/blog/threat-modeling-12-available-methods/
[71] Michael Cobb, 2021, "Definition of threat modelling" https://www.techtarget.com/searchsecurity/definition/threat-modeling
[72] Ramya Mohanakrishnan, 2021, "What Is Threat Modeling? Definition, Process, Examples, and Best Practices", https://www.spiceworks.com/it-security/network-security/articles/what-is-threat-modeling-definition-process-examples-and-best-practices/
[73] Zero-SWARM - D2.2 Eco-designed architecture, specifications & benchmarking, 2023, Grant Agreement: 101057083
[74] Vogel-Heuser, B., Diedrich, C., Fay, A., Jeschke, S., Kowalewski, S., Wollschlaeger, M. & Göhner, P. 2014. Challenges for Software Engineering in Automation. Journal of Software Engineering and Applications, 7(5), pp.440-451. DOI:10.4236/jsea.2014.75041.
[75] Vyatkin, V. 2011. IEC 61499 as Enabler of Distributed and Intelligent Automation: State-of-the-Art Review. IEEE Transactions on Industrial Informatics, 7(4), pp.768–781. DOI: 10.1109/TII.2011.2166785
[76] Yan, J. & Vyatkin, V.V. 2011. Distributed execution and cyber-physical design of Baggage Handling automation with IEC 61499. In 9th IEEE International Conference on Industrial Informatics (INDIN'2011), pp.573-578. 26-29 July 2011. DOI: 10.1109/INDIN.2011.6034942
[77] Rodrigues, N., Oliveira, E. & Leitão, P. 2014. Self-organization Combining Incentives and Risk Management for a Dynamic Service-Oriented Multi-agent System. in L. Camarinha-Matos et al. eds., Technological Innovation for Collective Awareness Systems. Springer Berlin Heidelberg, pp. 101–108. DOI: 10.1007/978-3-642-54734-8_12

[78] Thomas, U., Finkemeyer, B., Kröger, T. & Wahl. F. M. 2003. Error-tolerant execution of complex robot tasks based on skill primitives. In Proc. of the IEEE International Conference on Robotics and Automation (ICRA'03), volume 3, pp. 3069 - 3075. DOI: 10.1109/ROBOT.2003.1242062

[79] Silva da, R.M., Junqueira, F., dos Santos Filho, D.J. & Miyagi, P.E. 2012. Design of Reconfigurable and Collaborative Control System for Productive Systems. In ABCM Symposium Series in Mechatronics, Vol. 5., pp. 813-822.

[80] Cybersecurity facts for energy sector; http://scm.oas.org/pdfs/2021/CP44161Castro.pdf

[81] Baseline security recommendations for IoT, ENISA, 2017; https://www.enisa.europa.eu/publications/baseline-security-recommendations-for-iot/@@download/fullReport

[82] Jesse Ku, 2021, How to ensure OT cybersecurity, https://www.plantengineering.com/articles/how-to-ensure-ot-cybersecurity/

[83] Saif Shariff, 2020, What is ISA/IEC 62443? https://orignix.com/what-are-the-provisions-of-isa-iec-62443/

[84] ISO/IEC TR 30164 Internet of things (IoT) — Edge computing, 2020

[85] Zero-SWARM - Deliverable D5.1 "Distributed automation and information management", 2023

[86] Zero-SWARM - D2.1 Definition & Analysis of Trials, KPIs & GDPR Compliance, revision 1, 2023, Grant Agreement: 101057083

Appendix A List of IEC 62443 documents

The following section contains a list of IEC 62443 documents.

General

This group includes documents that address topics that are common to the entire series:

- 62443-1-1 introduces the concepts and models used throughout the series. The intended audience includes anyone wishing to become familiar with the fundamental concepts that form the basis for the series.
- 62443-1-2 is a master glossary of terms and abbreviations used throughout the series.
- 62443-1-3 describes a series of quantitative metrics derived from the foundational requirements, system requirements, and other guidance material in the standards.
- 62443-1-4 provides a more detailed description of the underlying lifecycle for Industrial Automation & Control Systems (IACS) security, as well as several use cases that illustrate various applications.

Policies and Procedures

Documents in this group focus on the policies and procedures associated with IACS security:

- 62443-2-1 describes what is required to define and implement an effective IACS cybersecurity management system. The intended audience includes end users and asset owners who have responsibility for the design and implementation of such a program.
- 62443-2-2 provides a methodology for evaluating the level of protection provided by an operational IACS against cybersecurity threats and how to apply what is required by 62443-2-1.
- 62443-2-3 provides guidance on patch management for IACS. The intended audience includes anyone who has responsibility for the design and implementation of a patch management discipline.
- 62443-2-4 specifies requirements for suppliers of IACS systems and related components. The principal audience include suppliers of control systems solutions. This standard was developed by IEC TC65 WG10.
- 62443-2-5 provides guidance on what is required to operate an effective IACS cybersecurity management system. The intended audience includes end users and asset owners who have responsibility for the operation of such a program.

System Requirements

The documents in the third group address requirements at the system level:

- 62443-3-1 describes the application of various security technologies to an IACS environment. The intended audience includes anyone who wishes to learn more about the applicability of specific technologies in a control systems environment.
- 62443-3-2 addresses security risk assessment and system design for IACS. This standard is primarily directed at asset owners or end users.

- 62443-3-3 provides the foundations for assessing the security levels provided by an automation system. The principal audience include suppliers of control systems, system integrators, and asset owners.

Component Requirements

The fourth and final group includes documents that provide information about the more specific and detailed requirements associated with the development of IACS products:

- 62443-4-1 describes the derived requirements that are applicable to the development of products. The principal audience include suppliers of control systems products and of components included in control systems solutions.
- 62443-4-2 contains sets of derived requirements that provide a detailed mapping of the system requirements to subsystems and components of the system under consideration. The principal audience include suppliers of components embedded in control systems solutions.
- IEC 62443 defines seven (7) Foundation Requirements (FR) that are a basis for the industry most common issues in Cybersecurity. Hereafter the obstacles and solutions for each FR are described.

Appendix B Mapping of ISO/IEC TS 19249 Security-by-Design Principles to other standards

The following table contains a mapping of ISO/IEC TS 19249 Security-by-Design Principles to other standards, to allow the project participant to easily the approaches and principles relevant to their technologies and applications.

ISO/IEC TS 19249 Principle	Similar Approaches and principles
Domain separation	Restrict communications between some areas of the programme (ISO/IEC/IEEE 29148)
Domain separation	Restricted Data Flow (IEC 62443-3-3)
Layering	Assign certain functions to different modules (ISO/IEC/IEEE 29148),
Layering	Defence in Depth (OWASP2023)
Layering	Adapt the security functions to the specific architecture of edge computing (ISO/IEC TR 30164)
Encapsulation	Least Common Mechanisms (OWASP2023)
Encapsulation	Ensure that entities only communicate with other authorized entities and that networks are appropriately protected (ISO/IEC TR 30164)
Redundancy	Ensure quick system recovery from failure (ISO/IEC TR 30164)
Redundancy	Resource Availability (IEC 62443-3-3)
Virtualization	N/A
Use of least privilege	Least Privilege (OWASP2023)
Use of least privilege	Separation of privilege (OWASP2023)
Use of least privilege	Complete Mediation (OWASP2023)
Use of least privilege	Eliminate default passwords (CISA2023)
Use of least privilege	Ensure that access to or management of entities in the system is subject to authentication and authorization (ISO/IEC TR 30164)
Use of least privilege	Use Control (IEC 62443-3-3)
Attack surface minimization	Track and reduce “hardening guide” size (CISA2023)
Attack surface minimization	Web template frameworks with automatic escaping of user input (CISA2023)
Attack surface minimization	Use Parameterized queries (CISA2023)
Attack surface minimization	Use Memory safe programming languages

Attack surface minimization	Secure Hardware Foundation and Secure Software Components (CISA2023)
Attack surface minimization	No Security Guarantee (OWASP2023)
Attack surface minimization	Weakest Link (OWASP2023)
Attack surface minimization	Leveraging Existing Components (OWASP2023)
Attack surface minimization	Secure systems to ensure that they operate to design, that they cannot be hijacked, that they have no vulnerabilities, that they are available (ISO/IEC TR 30164)
Centralized parameter validation	Use Software Authorization Profile (CISA2023)
Centralized parameter validation	Perform Static and dynamic application security testing (CISA2023)
Centralized parameter validation	Code reviewing (CISA2023)
Centralized parameter validation	CVE completeness check (CISA2023)
Centralized parameter validation	Check data integrity for critical variables (ISO/IEC/IEEE 29148)
Centralized parameter validation	System Integrity (IEC 62443-3-3)
Centralized general security services	Implement single sign on (CISA2023)
Centralized general security services	Provide secure logging (CISA2023)
Centralized general security services	Assure data privacy (ISO/IEC/IEEE 29148)
Centralized general security services	Utilize certain cryptographic techniques (ISO/IEC/IEEE 29148)
Centralized general security services	Use of information security management system (ISO/IEC 30141)
Centralized general security services	Detect attacks and incidents (ISO/IEC TR 30164)
Centralized general security services	Record and report attacks and incidents (ISO/IEC TR 30164)
Centralized general security services	Provision the system to continuously mitigate attacks within a certain period of time (ISO/IEC TR 30164)
Centralized general security services	Identification and Authentication Control (IEC 62443-3-3)
Centralized general security services	Economy of Mechanism (OWASP2023)
Centralized general security services	Timely Response to Events (IEC 62443-3-3)
Centralized general security services	Centrally managed, system wide audit trail (IEC 62443-3-3)
Centralized general security services	Audit log accessibility (IEC-62443-3-3)
Preparing for error and exception handling	Consider the user experience consequences of security settings (CISA2023)

Preparing for error and exception handling	Keep specific log or history data sets (ISO/IEC/IEEE 29148)
Preparing for error and exception handling	Ensure quick system recovery from failure (ISO/IEC TR 30164)
Preparing for error and exception handling	Provision the system to tolerate function failures within a specified range and limit while basic functions run properly (ISO/IEC TR 30164)
Preparing for error and exception handling	Fail Safe (OWASP2023)
Preparing for error and exception handling	Timely Response to Events (IEC 62443-3-3)
Other design principles not covered in ISO/IEC TS 19249	Forward-looking security over backwards compatibility (CISA2023)
Other design principles not covered in ISO/IEC TS 19249	Create Software Bill of Materials (CISA2023)
Other design principles not covered in ISO/IEC TS 19249	Satisfy Cyber Performance Goals (CISA2023)
Other design principles not covered in ISO/IEC TS 19249	Apply all appropriate data protection principles where personal data is involved, when stored or processed on an entity, or when transmitted on networks between entities (ISO/IEC TR 30164)
Other design principles not covered in ISO/IEC TS 19249	Secure information to ensure its availability its integrity and its confidentiality (ISO/IEC TR 30164)
Other design principles not covered in ISO/IEC TS 19249	Data Confidentiality (IEC 62443-3-3)
Other design principles not covered in ISO/IEC TS 19249	Design security functions that can be flexibly deployed and expanded (IEC 62443-3-3)
Other design principles not covered in ISO/IEC TS 19249	Open Design (OWASP2023)

Appendix C Mapping of OPC UA functionalities and components to IEC-62443-4-2

The following table contains a mapping of OPC UA functionalities and components to IEC-62443-4-2 Component Requirements and Requirement enhancements.

ISA-62443-4-2 SL2 CRs and REs	OPC UA Profile/ Facet/Conformance Unit (CU)
CR 1.1: Human user identification and authentication	IssuedIdentityToken
	JSON Web Token (JWT), JWT UserTokenPolicy
	Security User JWT IssuedToken, Security User JWT Token Policy, OPC UA Authority Profile
RE (1): Unique identification and Authentication	IssuedIdentityToken
	JSON Web Token (JWT), JWT UserTokenPolicy
	Security User JWT IssuedToken, Security User JWT Token Policy, OPC UA Authority Profile
	User Token JWT Server Facet, User Token JWT Client Facet
CR 1.2: Software process and device identification and authentication	ApplicationAuthentication, X.509 v3 Security Certificates
	ApplicationInstance Security Certificate
	EndpointDescription, EndpointUrl, Hostname (Device)
	Security Default ApplicationInstance Security Certificate, Global Security Certificate Management Server Facet
CR 1.4: Identifier management	UserIdentityToken, UserTokenPolicy
	Security User JWT IssuedToken, Security User JWT Token Policy, OPC UA Authority Profile
	User Token JWT Server Facet, User Token JWT Client Facet
CR 1.5: Authenticator management	UserIdentityToken, UserTokenPolicy
	Security User JWT IssuedToken, Security User JWT Token Policy, OPC UA Authority Profile
	User Token JWT Server Facet, User Token JWT Client Facet
CR 1.8: Security certificates	Security Certificates, TrustLists (CertificateStore), OPC UA Security Services
	Obtaining, validating, and installing Security Certificate services
	Security Certificates
	Security Administration, Global Security Certificate Management
	Security Certificate Management Overview
CR 1.9: Strength of public key-based authentication	Cryptographic Keys
	Trusted Security Certificates
	Security Profiles: Basic256_Limits, SecurityPolicy [B] – Basic256Sha256
CR 1.14: Strength of symmetric	Symmetric Encryption
	SymmetricEncryptionAlgorithm

key-based authentication	Global Service Key Credential Pull/Push Facets, KeyCredential Service Server Facet, KeyCredential Service Client Facet
	SecuritKeyService (SKS), SymmetricEncryptionAlgorithm
CR 2.1: Authorization enforcement	UserAuthorization
	Authorization Services, IssuedIdentityToken
	AuthorizationService, JSON Web Token (JWT)
	User Token – JWT Server Facet, User Token – JWT Client Facet
RE (1): Authorization enforcement for all users (humans, software processes, and devices)	UserAuthorization
	Authorization Services, IssuedIdentityToken
	AuthorizationService, JSON Web Token (JWT)
	User Token – JWT Server Facet, User Token – JWT Client Facet
RE (2): Permission mapping to roles	Roles, JWT, and User Roles
	User Authorization, Role Type
	RolePermissions
	User Role Management Server/Client Facets
CR 2.8: Auditable events	Auditability, Auditing, Audit Event Management
	Auditing
	AuditSecurityEventType
	Auditing Server Facet, Auditing Client Facet, Best Practice – Audit Events
CR 2.11: Timestamps	Message replay, Timestamps,SecureChannelID
	TimestampsToReturn
	AuditEventType
	Auditing Server Facet
RE (1): Time synchronization	Cryptographic Keys (time validity of security profile)
	SourceTimestamp, VersionTime, Redundant Server Set Requirements
	Time Synchronization
	Security Time Synchronization
CR 2.12: Non-repudiation	Message alteration, Server Profiling,
	System Hijacking, Repudiation, Audit
	Event Management
	Signing, GetEndpoints, SecureChannel, Auditing, Proof of Possession, UserTokenPolicy (user), SecurityPolicy

CR 2.12: Non-repudiation	Message alteration, Server Profiling, System Hijacking, Repudiation, Audit Event Management
	Signing, GetEndpoints, SecureChannel, Auditing, Proof of Possession,
	User Token – JWT Server/Client Facets, Auditing Server Facet, Auditing Client Facet, Best Practice – Audit Events
CR 3.1: Communication integrity	Secure Channel – OpenSecureChannel
	Secure Channel Service Set
	Secure Channel, SecurityProtocol
	Security Policy Required, Security
	Policy [A] & [B]
RE (1): Communication authentication	Secure Channel – OpenSecureChannel
	Secure Channel Service Set
	Secure Channel
	Security Policy Required, Security
CR 3.3: Security functionality verification	Identity Provider, SecurityKeyService, Secure Channel, TLS
	OpenSecureChannel, CreateSession, Write
	OPC UA Secure Conversation (UASC), Verifying Message Security, Token Policy, Bad_SecureChannel
	User Token – JWT Server/Client facets, Security Policy [A] & [B]
CR 3.4: Software and information integrity	ApplicationInstance Security Certificate
	SoftwareCertificates
	ApplicationInstance Security Certificate, X.509 v3
	Security ApplicationInstance Security Certificate, Global Security Certificate Management Server/Client Profiles
CR 3.7: Error handling	Request/Response Service
	SessionDiagnosticsObjectType
	MessageChunks, Error Handling, Error Message, CloseSecureChannel
	Security Policy Required, Security Policy [A] & [B]
CR 3.8: Session integrity	Secure Channel, Session ID
	Session Service Set, Creating a Session, Auditing Session Service, SessionAutenticationToken
	Session Services Facets, Standard UA Client 2017 Profile, Base Server Behavior Facet
CR 4.1: Information confidentiality	Confidentiality, Confidentiality, Eavesdropping, Client/Server, PubSub, Confidentiality
	SecureChannel Service Set
	OPC UA HTTPS, WebSockets (Security)
	Security Policy Required, Security Policy [A] & [B]
CR 4.3: Use of cryptography	Asymmetric Cryptography, Cryptography, Symmetric Cryptography, SecurityPolicies, Random Number Generation, Security Certificate Management
	GetEndpoints, OpenSecureChannel

	Security Handshake, Security Certificates, AccessTokens, Security Header, Deriving Keys (Table 49)
	AccessToken Request Client Facet, Security User Access Control Base Profile, Best Practice – Random Numbers, Global Discovery and Security Certificate Management 2017 Server, Global Security Certificate Management Client 2017 Profile
CR 4.3: Use of cryptography	Asymmetric Cryptography, Cryptography, Symmetric Cryptography, SecurityPolicies, Random Number Generation, Security Certificate Management
	GetEndpoints, OpenSecureChannel
	Security Handshake, Security Certificates, AccessTokens, Security Header, Deriving Keys (Table 49)
	AccessToken Request Client Facet, Security User Access Control Base Profile, Best Practice – Random Numbers, Global Discovery and Security Certificate Management 2017 Server, Global Security Certificate Management Client 2017 Profile
CR 5.1: Network segmentation	Network Segmentation, OpenSecureChannel
	Transport Layer – LS, Communication Layer – Secure Channel, Application Layer – Session for Auth
	Standard UA Client 2017 Profile, Base Server Behavior Facet
CR 6.2: Continuous monitoring	Monitor Items, GetMonitoredItems Method, SetMonitoringMode. Subscription Server Facet, Standard UA Client 2017 Profile, Standard DataChange Subscription 2017 Server Facet
CR 7.1: Denial of service protection	Application Crashes, Fuzz Testing, Certification
	CreateSession, OpenSecureChannel, AuthenticationToken
	Session Services Facets, Standard UA Client 2017 Profile, Base Server Behavior Facet
RE (1): Manage communication load from component	Message flooding, GetEndpoints, OpenSecureChannel
	CreateSession, OpenSecureChannel, AuthenticationToken
	Session Services Facets, Standard UA Client 2017 Profile, Base Server Behavior Facet
CR 7.2: Resource management	Resource exhaustion, ClientAuthentication, ServerAuditing, OpenSecureChannel
	CreateSession, OpenSecureChannel, AuthenticationToken
	Session Services Facets, Standard UA Client 2017 Profile, Base Server Behavior Facet

Appendix D ENISA Good practices for IoT and Smart Infrastructures Tool

The following table presents only a part of ENISA's Good practices for IoT and Smart Infrastructures tool. It includes only the security-by-design and privacy-by-design security domains, presenting a description for each security domain together with a reference title.

Security domain	Description	Reference title
Security by design	GP-PS-02: Address cybersecurity through embedded features of endpoints rather than only on the network level, if it is possible considering constraints such as limited computing power. Embed cybersecurity in automation systems by introducing fail-safe and fail-secure mechanisms from design.	Security Guidance for Early Adopters of the Internet of Things Industry 4.0: Secure by design Automotive Cybersecurity Best Practices - Executive Summary The Malicious Use of Artificial Intelligence: Forecasting, Prevention, and Mitigation IIC Endpoint Security Best Practices Smarter Security for Manufacturing in The Industry 4.0 Era: Industry 4.0 Cyber Resilience for the Manufacturing of the Future Internet of Things Security Guidelines v1.2 Industrial Internet of Things Volume G4: Security Framework IoT Security White Paper 2017 GSMA CLP.11 IoT Security Guidelines Overview Document
Security by design	GP-PS-04: Perform risk and threat analysis involving cybersecurity experts from the very early stages of the design process of the device to find out which security features will be necessary. The analysis should include possible and tailored use cases that the device may encounter. It is recommended to develop threat modelling for the IIoT systems and attack trees to consider resilience to	Summary NISTIR 8183: Cybersecurity Framework Manufacturing Profile IIC Endpoint Security Best Practices NIST SP 800 82r2: Guide to Industrial Control Systems (ICS) Security Smarter Security for Manufacturing in The Industry 4.0 Era: Industry 4.0 Cyber Resilience for the Manufacturing of the Future

	various attack scenarios. Cybersecurity experts should be involved in the process to provide insights on threats and risks that the control systems are facing based on the experience and knowledge of current threat and risk landscape.	Internet of Things Security Guidelines v1.2 ISO/IEC 27002:2013 Information technology -- Security techniques -- Code of practice for information security controls IoT Security White Paper 2017 GSMA CLP.11 IoT Security Guidelines Overview Document IoT Security Maturity Model: Description and Intended Use
Security by design	GP-PS-05: In each design document include a chapter addressing security of all information and control systems in industrial environment. The functional and/or technical specification should at least include information on security measures used, including but not limited to: a) system architecture b) access control c) interfaces and communication security d) policy enforcement e) mobile security f) cloud security g) backup/disaster recovery	Security Guidance for Early Adopters of the Internet of Things Automotive Cybersecurity Best Practices - Executive Summary ETSI GR QSC 004 V1.1.1 (2017-03) Quantum Safe Cryptography; Quantum-Safe threat assessment Connected Consumer Products. Best Practice Guidelines NIST SP 800 53r4: Security and Privacy Controls for Federal Information Systems and Organizations ANSI/ISA-95 Part 1: Models and Terminology GSMA CLP.11 IoT Security Guidelines Overview Document
Security by design	GP-PS-01: Treat IoT cybersecurity as a cycle - not as an end-to-end process. Take into consideration cybersecurity aspects in any activity of the development of the solution from the very beginning. Adopt security by design approach both from the devices as well as from the infrastructure perspective.	Security Guidance for Early Adopters of the Internet of Things Industry 4.0: Secure by design Automotive Cybersecurity Best Practices - Executive Summary NISTIR 8183: Cybersecurity Framework Manufacturing Profile IIC Endpoint Security Best Practices IoT Security Guidance

	In a "Security by design" concept, this relates to Continuous Security Improvement cycles at every step of a smart manufacturing system development lifecycle (Secure SDLC), that is analysis, design, implementation, testing, operations & maintenance.	NIST SP 800 82r2: Guide to Industrial Control Systems (ICS) Security The Malicious Use of Artificial Intelligence: Forecasting, Prevention, and Mitigation Baseline Security Recommendations for IoT IEC 62443-2-1:2010 Establishing an industrial automation and control system security program GSMA CLP.11 IoT Security Guidelines Overview Document
Security by design	GP-PS-03: Equip, as deemed appropriate after a security and safety assessment, even the most basic connected devices of very limited processing capabilities (e.g. actuators, converters) with identification and authentication features and ensure compatibility with IAM class solutions. This especially applies to protection against unauthorised re-calibration or re-configuration, e.g. of measuring devices, through: a) principle of least privilege for accessing device configuration and calibration engineering tools b) authorisation and authentication for engineers accessing engineering tools c) strong physical security for L0/L1 devices d) disabling of vulnerable wireless protocols e) disabling of test/debug features	Security Guidance for Early Adopters of the Internet of Things Industry 4.0: Secure by design An Internet of Things Reference Architecture Automotive Cybersecurity Best Practices - Executive Summary Identity and Access Management for the Internet of Things - Summary Guidance Industrial Internet of Things Volume G4: Security Framework IoT Security White Paper 2017 GSMA CLP.13 IoT Security Guidelines for Endpoint Ecosystems Draft NISTIR 8228: Considerations for Managing Internet of Things (IoT) Cybersecurity and Privacy Risks
Privacy by design	GP-PS-06: Address privacy-related issues based on applicable local and international regulations, such as The General Data	The General Data Protection Regulation (GDPR) (EU) 2016/679

	Protection Regulation (GDPR). A compliance function in the organisation should ensure that all new systems comply with regulatory requirements. This involves having written requirements in technical specifications during tendering/procurement process. Organisations should also take into account accountability aspect of privacy protection and implement measures that will enable them to demonstrate their relevant actions and their effectiveness.	Security Guidance for Early Adopters of the Internet of Things Internet of Things (IoT) Security Best Practices Industrial Internet of Things: Unleashing the Potential of Connected Products and Services Identity and Access Management for the Internet of Things - Summary Guidance Connected Consumer Products. Best Practice Guidelines Industrial Internet of Things Volume G4: Security Framework NIST SP 800 53r4: Security and Privacy Controls for Federal Information Systems and Organizations ISO/IEC 27002:2013 Information technology -- Security techniques -- Code of practice for information security controls IoT Security White Paper 2017 IEC 62443-2-1:2010 Establishing an industrial automation and control system security program GSMA CLP.11 IoT Security Guidelines Overview Document IoT Security Maturity Model: Description and Intended Use
Privacy by design	GP-PS-08: Establish the physical location of data stored by the organisation and define between which organisations data will be transferred. Restrict access to collected personal data only to authorised individuals. Periodically revise access rights and terminate them as soon as possible after an employee's	Security Guidance for Early Adopters of the Internet of Things ETSI TR 103 375 SmartM2M; IoT Standards landscape and future evolutions ANSI/ISA-95 Part 1: Models and Terminology IoT Security Guidance NISTIR 8183: Cybersecurity Framework Manufacturing Profile

	change of position/leaving company.	Identity and Access Management for the Internet of Things - Summary Guidance Industrial Internet of Things Volume G4: Security Framework IEC 62443-2-1:2010 Establishing an industrial automation and control system security program NIST SP 800 53r4: Security and Privacy Controls for Federal Information Systems and Organizations GSMA CLP.11 IoT Security Guidelines Overview Document
Privacy by design	GP-PS-10: Separate data that can be used to identify an individual from other information and ensure its security (for storing and retrieving information, communication services, cryptography, etc.). Any personal data transferred within the IIoT environment shall be encrypted in the traffic.	Security Guidance for Early Adopters of the Internet of Things ETSI TR 103 375 SmartM2M; IoT Standards landscape and future evolutions Connected Consumer Products. Best Practice Guidelines Internet of Things Security Guidelines v1.2 Industrial Internet of Things Volume G4: Security Framework IoT Security White Paper 2017 GSMA CLP.11 IoT Security Guidelines Overview Document
Privacy by design	GP-PS-07: Define the scope of the data that will be processed by the device as well as the objective of this processing during the design phase. Ensure that only a minimal amount of personal data is collected by the device. Avoid collecting sensitive data. If you are a user of an IIoT system, do not provide any personal or sensitive information if it is not necessary.	Security Guidance for Early Adopters of the Internet of Things Security Challenges on the Way Towards Smart Manufacturing Industry 4.0: Secure by design IoT Security Guidance Identity and Access Management for the Internet of Things - Summary Guidance

		Connected Consumer Products. Best Practice Guidelines Putting Industrial Cyber Security at the top of the CEO agenda NIST SP 800 53r4: Security and Privacy Controls for Federal Information Systems and Organizations ANSI/ISA-95 Part 1: Models and Terminology GSMA CLP.11 IoT Security Guidelines Overview Document
Privacy by design	GP-PS-09: Conduct a Privacy Impact Analysis (PIA) for the data that will be processed by the device. It may be integrated with the overall risk management process.	Security Guidance for Early Adopters of the Internet of Things Industrial Internet of Things: Unleashing the Potential of Connected Products and Services Baseline Security Recommendations for IoT Internet of Things Security Guidelines v1.2 Industrial Internet of Things Volume G4: Security Framework NIST SP 800 53r4: Security and Privacy Controls for Federal Information Systems and Organizations GSMA CLP.11 IoT Security Guidelines Overview Document