CISSAN

Collective intelligence supported by security aware nodes

D2.1 Definition of the initial CISSAN architecture and distributed system elements and interfaces

Editors: Niko Candelin (Netox), Ilgin Safak (University of Jyväskylä)

contact: niko.candelin@netox.fi

Abstract

The initial architecture document is a summary of architectural and design considerations based on the project context (including the use cases and partners' priorities), presenting the current landscape of Internet of Things (IoT) and Operational Technology (OT) security technologies and existing gaps, CISSAN design and architectural principles, core elements under development, and the key research domains / lines of the project.

Participants in the CISSAN project are:

- University of Jyväskylä
- Bittium Wireless Ltd
- Bittium Biosignals Ltd
- Geodata ZT GmbH
- Mattersoft
- Mint Security Ltd
- Netox Ltd
- Nodeon Ltd
- Scopesensor Ltd
- Wirepas Ltd
- Councilbox Ltd
- Affärsverken Karlskrona AB
- Arctos Labs
- Clavister AB
- Blekinge Tekniska Högskolan
- Blue Science Park
- Savantic AB
- Techinova AB

CISSAN – Collective intelligence supported by security aware nodes

D2.1 Definition of the initial CISSAN architecture and distributed system elements and interfaces

Editors: Niko Candelin (Netox), Ilgin Safak (University of Jyväskylä)

Project coordinator: Alexey Kirichenko (University of Jyväskylä)

CELTIC published project result

Executive Summary

The main objective of the CELTIC-NEXT CISSAN project is to enhance the cybersecurity, cyber resilience, and automation of Internet of Things (IoT) and Operational Technology (OT) ecosystems that utilize device, edge, and cloud computing capabilities. This report presents the rationale behind the design, architectural principles, and key research domains of CISSAN, including the project context based on the use cases and partners' priorities, the current landscape of IoT and OT security technologies, existing gaps, and core elements under development. As IoT and OT devices and systems are often vulnerable to cyberattacks, they can be used as entry points for attackers to target other systems or networks, such as critical infrastructures or enterprise IT environments. Therefore, securing these devices and systems is crucial for economies and societies. Despite the availability of various security solutions for IoT and OT, considerable risks, challenges, and limitations persist. CISSAN addresses several of these issues by employing collective intelligence (CI), artificial intelligence (AI), and distributed ledger technologies. The project explores CI methods for IoT and OT network security, leveraging the collection, coordination, and aggregation of threat intelligence from all entities in the system to enhance security measures. It identifies issues and deficiencies in the security of CI-enabled IoT and OT networks, including the absence of standardized protocols and the challenges associated with managing complex deployments. To address these issues, CISSAN proposes solutions, including advanced threat detection algorithms and mechanisms supporting near-real-time response. Additionally, the project outlines key architectural components for enabling CI in IoT and OT networks, such as secure communication protocols, robust authentication, data quality verification and event tracking mechanisms, and scalable infrastructure. By integrating these elements, CISSAN aims to enhance the security and resilience of the IoT and OT environments, ultimately contributing to the protection of critical infrastructures and the overall safety of digital ecosystems.

This report provides conceptual and architectural information for CISSAN stakeholders concerned with the security and resilience of IoT and OT networks, which are increasingly vulnerable to cyberattacks. This includes a review of contemporary security technologies and methodologies by examining the innovative strategies and solutions developed to safeguard assets and maintain operational continuity. Audiences benefiting from this report include technology companies, cybersecurity firms, infrastructure suppliers, and industry experts. The report offers practical insights for technology companies and cybersecurity firms regarding the latest methodologies and technologies to address current security gaps and difficulties, facilitating the development of more effective solutions. Infrastructure providers will acquire expertise to improve the resilience of vital systems, guaranteeing the continuous operation of critical services. The report is also a valuable resource for cybersecurity professionals and educators, improving knowledge and fostering a more skilled workforce. It emphasizes CI in IoT and OT networks, providing a framework for stakeholders to collaboratively tackle vulnerabilities and mitigate emerging risks. The stakeholders can thus improve the security and stability of digital ecosystems and critical infrastructures, safeguarding operations while promoting economic and societal resilience within the EU. The results promote innovation and facilitate the extensive implementation of IoT and OT technologies in new areas under enhanced security measures. This report enables stakeholders to implement proactive strategies against cyber threats, safeguarding their success and improving the broader cybersecurity environment. It is naturally a guiding document for the CISSAN partners as well in their project efforts. Together with deliverable D1.1, this document forms an initial foundation for the project work, presenting technical state-of-the-art and architectural considerations. This is complemented by the outcomes of WP3 on potential business models for exploiting the project results.

List of Authors

- Niko Candelin, Netox
- Ilgin Safak, University of Jyväskylä
- Alexey Kirichenko, University of Jyväskylä
- Ann Sjökvist, Mint Security Ltd
- Elina Partanen, Mint Security Ltd
- Tapio Frantti, University of Jyväskylä

Table of Contents

List of Figures

Abbreviations

ABAC	Attribute-Based Access Control
AES	Advanced Encryption Standard
AI	Artificial Intelligence
CI	Collective Intelligence
CISSAN	Collective Intelligence Supported by Security Aware Nodes
CRA	Cyber Resilience Act
DLT	Distributed Ledger Technology
DoS	Denial of Service (attack)
DT	Digital Twins
DTLS	Datagram Transport Layer Security
EAP	Extensible Authentication Protocol
ECC	Elliptic-Curve Cryptography
FL	Federated Learning
GAN	Generative Adversarial Network
GDPR	General Data Protection Regulation
IDPS	Intrusion Detection and Prevention System
IDS	Intrusion Detection System
IoT	Internet of Things
IP	Internet Protocol
IT	Information Technology
LoRaWAN	Long Range Wide Area Network
MAS	Multi Agent Systems
MFA	Multi Factor Authentication
ML	Machine Learning
MQTT	Message Queuing Telemetry Transport
OAuth	Open Authorization
OT	Operational Technology
PANA	Protocol for Carrying Authentication for Network Access
RBA	Risk-based Authentication
RSA	Rivest–Shamir–Adleman (public-key cryptosystem)
SDN	Software-Defined Network
SI	Swarm Intelligence
SOS	Self Organizing Systems
SSO	Single Sign-on
UC	Use Case
UMA	User-Managed Access
XACML	eXtensible Access Control Markup Language
VM	Virtual Machine
VU	Virtual User
ZT	Zero-Trust

1 Introduction

Traditional security solutions, such as firewalls, antivirus software, coding, and encryption, are often insufficient or unsuitable for IoT and OT networks. Therefore, novel security approaches and solutions are needed to address IoT and OT environments' specific security challenges and requirements.

The CELTIC-NEXT CISSAN (Collective Intelligence Supported by Security Aware Nodes) project is a collaborative research initiative that aims to enhance the cybersecurity, cyber resilience, and automation of IoT and OT ecosystems that utilize device, edge, and cloud computing capabilities (thus, including information technology (IT) elements). The project involves partners from Finland, Sweden, Spain, and Austria and includes three use cases (UCs): smart transportation (UC1); smart energy grids (UC2); and mining and tunnelling (UC3).

The project leverages multiple paradigms and approaches to address security challenges and threats that IoT and OT systems face, such as collective intelligence (CI), artificial intelligence (AI), and distributed ledger technologies (DLT). CI implies collecting, analyzing, and sharing intelligence (information and insights) from multiple sources and domains, such as IoT, OT, IT and cloud. CISSAN is an ambitious and pragmatic research project delivering considerable security improvements to its UCs while aiming at high generalizability of the produced results to IoT and OT networks ranging from the design stage to the operational stage (which requires project solutions to be appropriately modular and adaptive). Through interviews with the Use Case owners and other partners and through discussions and analysis at Use Case-focused workshops, the project identifies and analyses security challenges and threats in the three UCs in the IoT and OT domains and proposes theoretical platforms and frameworks that integrate security solutions and technologies to prevent and mitigate cyberattacks. The project will also produce documents covering IoT and OT security best practices, management, and governance, presenting relevant processes, policies, and standards both for the project team and the project stakeholders (including potential customers). In the documentation, we plan to discuss security controls and measures required but not provided by CISSAN and propose ways of integrating those with CISSAN technologies.

The initial architecture document D2.1 is a summary of architectural and design considerations based on the project context (including the use cases and partners' priorities), presenting the current landscape of Internet of Things (IoT) and Operational Technology (OT) security technologies and existing gaps, CISSAN design and architectural principles, core elements under development, and the key research domains / lines of the project. Together with deliverable D1.1, which presents technical state-of-the-art in the key CISSAN domains, this document forms an initial foundation for the project work.

1.1 CISSAN Principles and Core Objectives for Design

The project will follow the following set of general security principles in its research and engineering efforts to develop technologies and propose methods to enhance the security and privacy of IoT and OT environments:

1. *Secure by design*: select and embed security and privacy features in target devices, products, systems, and platforms from the initial stages of the development lifecycle, but not as an afterthought or add-on. CISSAN will implement or support such security measures as network segregation, security monitoring and logging (including relevant data flows monitoring), intrusion detection and prevention, data privacy protection, physical security, and incident response.
2. *Least privilege*: grant the minimum level of access and permissions to devices and users, according to their roles and responsibilities, to reduce the potential impact of unauthorized or malicious actions.
3. *Data minimization*: collect and store only the necessary data for intended purposes, and delete or anonymize the data when no longer needed to protect the data privacy and reduce the risk of data exposure.
4. *Use of cryptography*: encrypt data in transit and at rest to prevent unauthorized access or modification of the data and sign and verify the data exchanged among endpoints, edge devices and cloud backends to ensure the authenticity and integrity of the data using strong and standardized cryptographic algorithms and protocols.

5. *Audit and accountability*: maintain logs and records of activities, events, and transactions in target devices, products, systems, and platforms and enable auditing and accountability mechanisms to monitor and verify the security and privacy of those.

CISSAN's initial core architectural objectives are:

1. CISSAN initial architecture anticipates the use of various techniques, such as deep learning, Generative Adversarial Networks (GAN), CI, and blockchain, to enhance the security capabilities and performance of IoT and OT environments. CISSAN leverages the cloud and edge computing paradigms to enable efficient and scalable data processing and to employ cloud-based features, such as threat intelligence, advanced analytics, and cross-domain collaboration.
2. CISSAN leverages CI of IoT and OT devices and backends to share security information and alerts, such as indicators of compromise, signatures, or policies, and to coordinate planned responses and actions.
3. CISSAN creates a set of security management and governance methods and documents, comprising processes, policies, standards, and best practices, to guide the design, implementation, integration, and operation of IoT and OT platforms and devices and to help achieve compliance with relevant regulations.
4. CISSAN covers various radio frequency technologies, such as Wi-Fi, Bluetooth, Zigbee, LoRaWAN, cellular, and others, to enhance communication security. The project utilizes gateways to aggregate data from IoT and OT devices, translate protocols, perform edge computing, and make local decisions. Gateways may implement other security functions such as device authentication, data encryption, and access control.
5. CISSAN's design incorporates a scalable and flexible IoT / OT security framework for the anticipated growth, new devices, novel technologies, and changing business requirements. The project follows a data-driven approach and implements data management processes such as data collection, storage, ingestion, processing, validation, analysis, visualization, security, and governance.
6. CISSAN ensures seamless integration of its security mechanisms with existing enterprise systems, third-party services, and APIs for data exchange, business process automation, and decision-making.

1.2 IoT and OT

IoT is a network of interconnected devices that collect, process, and exchange data over the Internet and other communication networks. IoT devices can range from smart home appliances and wearable devices to industrial machines and sensors that monitor and manage physical processes. Such industrial devices employing supervisory control and data acquisition (SCADA) are often considered Operational Technologies. IoT devices and systems are key elements in both CISSAN UC1 and UC3, while UC2 is an OT-centric UC. IoT and OT offer many benefits, such as improved efficiency, convenience, and productivity, but their use also poses significant security challenges.

Traditional security solutions, such as firewalls, antivirus software, and encryption, are often insufficient or unsuitable for IoT and OT environments, as they may not be able to cover entire environments/ecosystems, to scale with the volume of data or variety of devices, or to operate in resource-constrained hardware. For example, the networking/communication infrastructure of IoT and OT environments has multiple security weaknesses such as vulnerabilities in cellular networks, Wi-Fi networks, and communication protocols. Since these networks and protocols are critical for IoT and OT operations, attackers often target them to disrupt services or compromise safety.

Cyberattacks to IoT and OT devices and systems include malware infection, denial-of-service (DoS), data theft and tampering, and execution of unauthorized operations. Attackers can manipulate device identifiers, MAC addresses, or other parameters to deceive network gateways to compromise the confidentiality, integrity, and availability of data and devices. Moreover, IoT and OT devices and systems can be used as entry points or stepping stones for attackers to target other systems or networks, such as critical infrastructures or enterprise IT environments It is thus essential for IoT and OT systems, including e.g., cloud-based elements, to secure vast attack surfaces. To this end, there is a need for novel security approaches and solutions that can address the specific security challenges and requirements of these environments. Since the security of these environments is crucial for modern economies and societies, new EU regulations, such as the Cyber Resilience Act

and the NIS 2 Directive, place new requirements for securing and validating connected software, devices, and networks.

It is important to observe that cyberattacks are sometimes difficult to distinguish from other disruptions in IoT and OT networks, especially when those are detected as anomalies by ML-based detection engines. In the interviews with the Use Case owners and in the project workshop discussions, however, it was noted that from the network operator point of view, the difference between intentional cyberattacks and natural disruptions can be insignificant, and threats of the two types can be equally crucial to counter. Arguably, this often applies to networks of critical infrastructures and other safety-critical networks, and all the CISSAN's Use Cases belong with these categories. Nevertheless, helping network operators in understanding the root causes of incidents is valuable, and the project will explore ways of achieving that (e.g., through ML explainability techniques or higher specificity of anomaly detection models).

2 Current IoT and OT Security Solutions

2.1 Security Techniques for IoT and OT Networks

There is no one-size-fits-all solution for IoT and OT security because different devices and applications often have different security requirements and constraints and because opportunities for integrating security controls depend on the maturity level of systems and networks, ranging from the design stage to the operational stage. We list here several security techniques and solutions commonly used in IoT and OT systems:

- *Encryption*: Encryption refers to a process of transforming data into an unreadable form that can be decrypted only by the authorized parties. Encryption can protect the correctness, confidentiality, and integrity of data transmitted or stored by devices and other elements of IoT and OT systems. Encryption can be applied at different layers, such as data, network, transport, or application. Encryption algorithms can be symmetric or asymmetric, depending on whether they use the same or different keys for encryption and decryption. Examples of encryption primitives for IoT and OT are AES, RSA, ECC, and PRESENT, and many other schemes and algorithms based on those.
- *Authentication*: Authentication process aims to verify the identity of a device or a user that tries to access the system or communicate with a given device in the system. Authentication can prevent unauthorized access and impersonation attacks. Authentication can be based on different factors, such as passwords, tokens, certificates, biometrics, or behavioral patterns. Examples of authentication protocols for IoT and OT devices are EAP, PANA, DTLS, and MQTT.
- *Authorization*: Authorization refers to a process of granting or denying access rights or privileges to a device or a user that has been authenticated by a given device or system. Authorization can enforce access control policies and prevent unauthorized actions based on different models, such as role-based, attribute-based, or policy-based. Authorization frameworks for IoT and OT devices include OAuth, UMA, XACML, and ABAC.
- *Firewall*: Firewall is a software or hardware plus software component that monitors and filters the incoming and outgoing network traffic of an IoT/OT device or system. Firewall can prevent or block unwanted or malicious traffic, such as DoS attacks, malware, or spyware. Firewall can be implemented at different levels, such as device, gateway, or cloud based on different rules, such as packet filtering, stateful inspection, or application layer filtering. Firewall solutions for IoT and OT devices include IPTables, Netfilter, Snort, and Suricata.
- *Intrusion Detection and Prevention System (IDPS)*: IDPS is a software or hardware plus software component that detects and responds to anomalous or malicious intrusion attempts, data exfiltration, or botnet activities in a given IoT/OT device or system. IDPS can alert on, block, or mitigate such threats. IDPS can be based on different techniques, such as signature-based, anomaly-based, or specification-based. It can be deployed at devices, gateways, or cloud. Examples of IDPS solutions for IoT and OT systems are Bro, Snort, and Suricata.

2.2 CI Methods for IoT and OT Network Security

CI is one of the multiple paradigms and approaches to address security challenges and threats that IoT and OT systems face. CI implies collecting, analyzing, and sharing intelligence (information and insights) from multiple sources and domains, such as IoT, OT, IT, and cloud. This section briefly presents two key CI methods that can be employed for IoT and OT network security: AI and multi-agent systems (MAS). Details and additional CI methods for IoT/OT security can be found in CISSAN deliverables D1.1 (Section 3.1) and D2.2.

2.2.1 Artificial Intelligence

CI can utilize AI to process extensive data produced by aggregated inputs, which provide insight into threats, to make or support informed decisions using predictive analytics, anomaly detection, pattern recognition, clustering, natural language processing, and other approaches. Mohamudally [1] provides a comparison of mathematical models for CI and a discussion of their suitability for implementation on mobile devices. He also proposes, a framework for modeling CI systems using graph theory and artificial neural networks.

Hierarchical machine learning (ML) is a sophisticated methodology that arranges data and learning processes into stratified structures, mirroring the intrinsic hierarchical characteristics of numerous real-world issues. This methodology employs various degrees of abstraction, with each layer analyzing data at distinct granularity, hence improving the model's capability to identify intricate patterns and relationships within the data. Hierarchical models frequently integrate unsupervised and supervised learning methodologies, facilitating enhanced accuracy and interpretability of outcomes. This method is very efficient in extensive data contexts, such as cloud computing, where it adeptly manages substantial data volumes while minimizing computational expenses and enhancing scalability. Moreover, hierarchical ML is significantly pertinent to CI, since it reflects the operational dynamics of collective systems by utilizing several levels of abstraction and collaboration. Organizing data processing and learning activities into layered frameworks enhances cooperation and information sharing among agents, hence fostering CI and improved problem-solving abilities.

Federated learning (FL) is a methodology that enables the training of a ML model across several devices and/or servers, hence eliminating the need for data centralization. FL includes the following steps (see Figure 1):

1. A global model is established and sent to participant nodes in the network.
2. Each node autonomously trains and updates the model with its local dataset.
3. Nodes only transmit changes to the model, such as weights or gradients, to an aggregator, rather than sending their local data.
4. The aggregator enhances the global model by consolidating updates from all participant nodes by using various aggregation methods to enhance the learning process.
5. The revised global model is sent to participant nodes for further training or deployment.

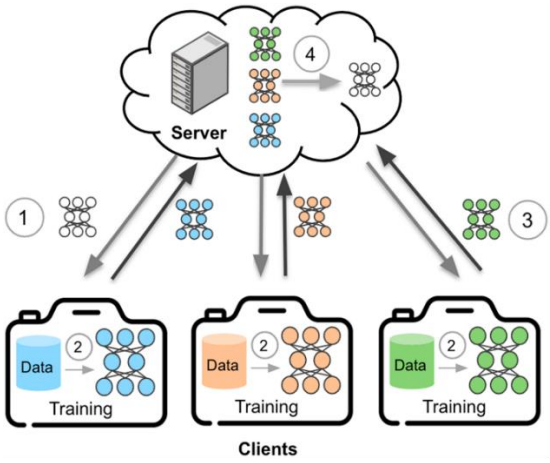

Figure 1. Federated Learning (source: *Sony AI*)

The FL process is incrementally improved by using a broader data set with each iteration. The FL paradigm harnesses the CI of distributed devices to facilitate collaborative model training. It leverages decentralized computation to improve network resilience against evolving threats [2], [3].

FL can be categorized into three main types [4]:

1. Centralized FL: A central server orchestrates the training process. Local devices (clients) train models using their data and transmit model changes (e.g., weights and gradients) to the central server. The server consolidates these updates to create a global model, which is subsequently transmitted back to the clients for additional training. This approach improves privacy as raw data remains on local devices. Centralized FL can be categorized as follows:
    - Horizontal FL (HFL): In HFL, data is segmented by samples, indicating that several clients possess datasets with identical feature spaces but distinct sample spaces. This is beneficial when various organizations or devices gather analogous data kinds from distinct people. For instance, hospitals possess patient data, with each institution maintaining records for distinct patients yet utilizing the same sorts of medical documentation.
    - Vertical FL (VFL) entails data segmented by features, wherein many clients possess datasets with identical sample spaces but distinct feature spaces. This is relevant when various organizations possess complementary information regarding the same group of users. For example, a bank and an insurance business may partner,

with the bank possessing financial data and the insurance company holding health data on the same individuals.
    - Federated Transfer Learning (FTL) integrates FL with transfer learning to address situations where clients possess distinct features and sample spaces. This approach is especially beneficial when there is minimal data overlap among clients, facilitating knowledge transfer across domains to enhance model performance.
2. Decentralized FL: In contrast to the centralized FL, decentralized FL operates without a central server. Clients engage in direct communication to exchange model updates among themselves. Every client disseminates model updates to its counterparts, and the updates are consolidated in a decentralized fashion. This peer-to-peer methodology can enhance resilience and mitigate the risk of a singular point of failure. Nevertheless, it may pose difficulties in preserving synchronization and consistency throughout the network.
3. Heterogeneous FL (HeteroFL): HeteroFL tackles the challenge of heterogeneity in FL settings when clients exhibit varying computing capabilities, data distributions, and network conditions. This approach facilitates the training of models capable of adapting to varied settings, so ensuring that all clients can effectively contribute to the global model. HeteroFL methodologies seek to develop resilient models capable of effective generalization despite variances, frequently employing strategies such as personalized models or domain adaptation. HeteroFL can enhance the overall efficacy and equity of the FL system. It can be categorized based on device or data heterogeneity:
    - Device Heterogeneity: Examines the variations in computational capability and resources among clients. It guarantees that clients with differing capacities for processing power, memory, and battery life can nevertheless engage effectively in the FL process. Methods like model compression and adaptive training can be employed to address these disparities.
    - Data Heterogeneity: Data heterogeneity pertains to the discrepancies in data distributions and types among various clients. This is prevalent in real-world situations when data gathered by various devices or organizations may not exhibit identical distribution.

However, the extensive use of AI for CI presents considerable risks, such as model poisoning and model evasion in FL. Model poisoning occurs when a malevolent individual intentionally introduces erroneous data into the training process of an ML model, leading to lower performance, incorrect learning by the model, and predictions yielding erroneous or unfavorable outcomes [5]. Model evasion refers to the intentional alteration of input data by a malicious actor to deceive an ML model, resulting in misclassification by the model. This strategy is often used to obscure cyber threats that would typically be identified by the system. Implementing multi-faceted solutions, including data cleaning, regular model retraining, continuous performance monitoring, differential privacy measures, and utilizing reliable data sources, may effectively prevent or minimize model evasion and model poisoning attacks. These models' resilience against such attacks may also be enhanced by training them by employing challenging examples and ensemble techniques [6].

2.2.2 Multi-Agent Systems

MAS are made up of independent multiple agents (devices) that communicate with each other and cooperate to perform specific tasks (see Figure 2), whereas crowdsourcing utilizes a substantial collective of individuals to jointly contribute to a job or project. MAS offer great potential in solving complex problems, efficient decision-making processes, and adapting to dynamic environments [7]. These systems can detect threats, perform data analysis, and develop rapid response mechanisms in IoT networks [8]. Inter-agent interaction and coordination enable CI systems to work proactively against threats and produce safer and more effective solutions [9].

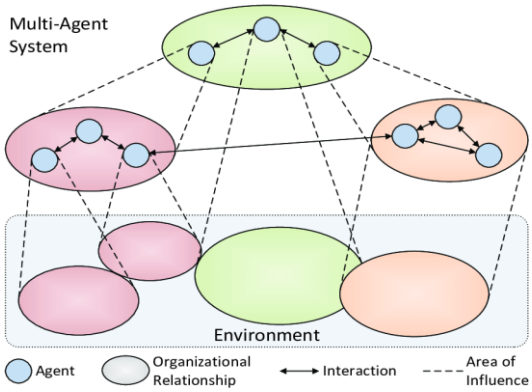

Figure 2. Multi-agent systems [7]

The primary characteristics of MAS encompass [10]:

- *Decentralization*: MAS function autonomously, allocating work across several agents to improve robustness and scalability.
- *Autonomy*: Each agent is capable of independently and autonomously making choices using local knowledge and established procedures, without requiring an overarching system control.
- *Collaboration*: Agents exchange information and negotiate with one another to attain shared objectives, enhancing overall system efficacy and resilience. MAS can be used for CI in IoT/OT networks, where distinct devices of various technologies collaborate to accomplish certain tasks (e.g., load balancing and distributed decision-making).
- *Adaptability*: MAS can flexibly adjust to changes in environmental or system circumstances, making them appropriate for complex and evolving attack scenarios.
- *Heterogeneity*: Agents may possess varying capacities and fulfill distinct tasks within the system.

However, the MAS architecture also has security problems, including susceptibility to cyber-attacks, the need for secure communication protocols, and the significance of fault-tolerant technologies [10].

Example MAS usage in IoT networks [11]:

- Intelligent transportation systems, whereby vehicles (agents) cooperatively optimize routes.
- Distributed sensor networks, in which sensors (agents) exchange data and collaboratively choose environmental monitoring strategies.
- In a smart home IoT ecosystem, diverse products, such as thermostats, lighting systems, and security cameras, may independently collaborate to enhance energy efficiency according to customer preferences.

2.3 Problems and Gaps in Security Solutions for IoT and OT Networks

Despite the availability and development of various security solutions for IoT and OT, there are still several challenges and limitations that need to be addressed, such as:

- *Resource constraints*: IoT and OT devices are often constrained by limited resources, such as battery power, memory, processing power, or communication bandwidth. These constraints can negatively affect the performance, scalability, and usability of security solutions, as they may require too many resources or introduce too much overhead. For example, encryption algorithms may require complex computations, authentication protocols may require frequent message exchanges or verification of credentials, and firewalls or IDPS may require constant monitoring or updating of detection rules and signatures. Therefore, security solutions for IoT and OT devices should be lightweight, efficient, and adaptive to limited resources.

- *Heterogeneity*: IoT and OT devices are heterogeneous in terms of their hardware, software, functionality, communication, applications, etc. This heterogeneity can pose interoperability, compatibility, and standardization issues for security solutions, as they may not work well across different devices or platforms. For example, encryption algorithms and authentication protocols may not be supported by or compatible between different devices or systems.

Similarly, firewalls or IDPS may not be able to filter traffic or detect malicious entities or behavior in traffic from different sources or in different formats. Therefore, security solutions for IoT and OT devices need to be flexible, interoperable, and compliant with relevant standards.

- *Use case diversity*: IoT and OT systems are diverse in terms of the number of devices, location, ownership, and usage. This diversity can pose scalability, management, and privacy issues for security solutions in protecting large and dynamic networks of IoT and OT devices. For example, encryption algorithms may not be able to generate or distribute high-quality keys for a large number of devices, authentication protocols may not be able to authenticate or revoke rights from dynamic and distributed devices, firewalls or IDPS may not be able to monitor or carry out response actions for large and diverse traffic. Therefore, security solutions for IoT and OT devices should be scalable, distributed, and privacy-preserving in diverse settings.

In Section 2 of CISSAN deliverable D2.2, we discuss in detail architectural issues in CI-enabled IoT and OT networks, including security issues and threat examples. Section 3 of D2.2 presents paradigms and techniques that can be employed for addressing those issues, including DLT, Zero Trust Architecture (ZTA), Dynamic Isomorphism, Knowledge Graphs and Ontologies, and Digital Twins.

3 Key Architectural Components for Enabling CI in IoT and OT Networks

Successful implementation of CI in IoT and OT networks requires a resilient, scalable and adaptable architecture capable of supporting decentralized decision-making, real-time data processing, and effective communication among various devices. The essential architectural elements for facilitating CI in IoT and OT systems are described in this section.

3.1 IoT Network Architecture

The IoT network infrastructure comprises physical and virtual elements that facilitate network operations, including nodes, servers, routers, and switches. The IoT network infrastructure serves as the foundation of CI in IoT, facilitating uninterrupted connectivity across devices, edge nodes, fog systems, and the cloud. Data transmission between IoT devices and central systems may be enabled using backbone connections to ensure efficient and reliable network operation. IoT network backbone connections are the principal pathways that connect a variety of IoT devices and systems to the central network infrastructure. These connections are crucial for facilitating data transmission between IoT devices and central servers or data centers. The backbone generally comprises high-capacity connections, such as fiber optics, that provide dependable and rapid communication across the network. This infrastructure facilitates the extensive data produced by IoT devices, allowing for effective data processing, storage, and analysis. The backbone serves as the primary nervous system of an IoT network, enabling uninterrupted connection and communication. IoT networks may also function without a backbone, which often depends on decentralized or ad hoc connectivity, whereby devices interact directly with one another or via local gateways. These networks may exhibit more flexibility and facilitate simpler deployment in certain contexts, such as distant locations or temporary configurations. Nonetheless, they may have issues regarding scalability, dependability, and data throughput in comparison to backbone-supported networks. In the absence of a strong backbone, overseeing substantial data volumes and maintaining continuous connectivity across the network may be challenging. This may result in possible bottlenecks and diminished efficiency, particularly when the quantity of linked devices escalates. The design must accommodate several communication protocols (e.g., Wi-Fi, Bluetooth, Zigbee, Long Range wide-area network (LoRa WAN)) to address the diversity of IoT devices. Low-power WAN (LPWAN) and 5G technologies are essential elements, offering high-speed and dependable connections for IoT devices in urban and rural settings. Mesh networks are often used to provide resilience and fault tolerance, allowing devices to connect directly with one another without dependence on centralized routers or gateways. This is particularly crucial in IoT implementations where network dependability and scalability are vital [12].

The IoT architecture (see Figure 3) has a multi-tiered framework intended to address the complexity and heterogeneity of IoT systems. The predominant model is the three-tier architecture, comprising the perception, network, and application layers. The perception layer comprises sensors and devices that gather data, the network layer manages data transfer, and the application layer processes and employs the data for diverse applications. Additional concepts include middleware architecture, service-oriented architecture, and five-layer architecture, each catering to distinct requirements such as scalability, interoperability, and effective data management. The IoT environment comprises four principal elements: devices, connectivity, data processing, and user interface. Devices comprise sensors and actuators that gather and respond to data. Connectivity denotes the diverse communication networks and protocols for data flow among devices. Data processing encompasses the analysis and administration of gathered data, frequently employing cloud or edge computing technologies. The user interface includes the applications and services that enable users to engage with the IoT system, offering insights and control over connected devices. This extensive framework emphasizes the interrelated characteristics of IoT components and their functions in establishing a viable IoT ecosystem [13].

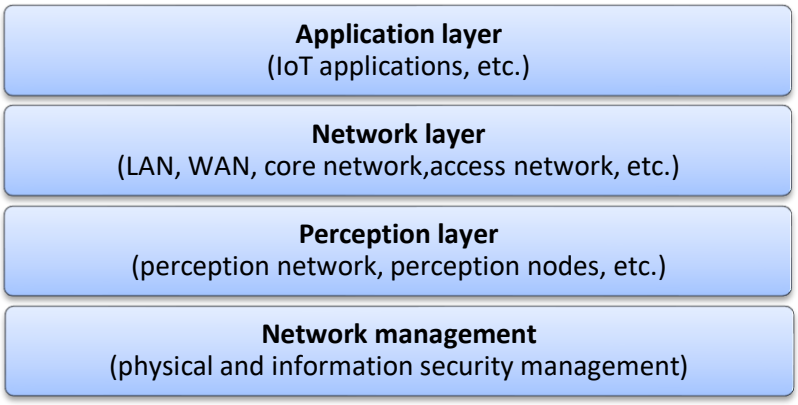

Figure 3. IoT architecture [13]

3.2 Architectural Patterns

Architectural patterns provide reusable solutions to prevalent design issues in software architecture. Architectural patterns for CI examine different frameworks and approaches aimed at improving the coordination, learning, and problem-solving capabilities of distributed systems. These patterns are fundamentally connected to network architecture, as they depend on the underlying infrastructure to enable communication, data exchange, and the integration of various computational resources, thus facilitating the effective operation of CI systems. [14] examines architectural patterns for CI, highlighting the incorporation of stigmergic coordination for indirect communication among agents, reactive and adaptive infrastructures to enable dynamic interactions, and hybrid human-computer systems that promote the collective generation and dissemination of knowledge. Stigmergy is a nature-inspired coordination mechanism that facilitates the indirect coordination of agents or actions through the environment (which can be a valuable enabler for achieving the objectives of CISSAN T5.2). Agents leave environmental traces that subsequently motivate consecutive actions by the same or different agents. These elements collaborate to improve the design process by using CI to address complex architectural challenges. [15] introduces an architectural pattern for CI, which leverages stigmergy to enable indirect communication among agents via environmental traces. Key components include stigmergic coordination, a reactive and adaptive infrastructure, a hybrid human-computer system, and a virtual artifact network, all of which facilitate the collective creation and sharing of knowledge. Even in the absence of direct communication, this technique enables sophisticated, coordinated action. [16] presents a framework designed to enhance collective learning and CI through a structured approach. Key components include a MAS for distributed problem-solving, stigmergic coordination for indirect communication among agents, and a feedback loop to continuously improve the system's performance based on user interactions and environmental changes.

Graph theory could represent network architecture by modeling devices as nodes and communication channels as edges, facilitating the optimization of network performance and security using different graph algorithms. [17] explores methods and algorithms from graph theory for optimizing the placement of security services in IoT networks (which is the primary objective in CISSAN T5.3) by modeling the network as a weighted graph and applying algorithms like dominating sets and shortest paths. This strategic placement enhances network coverage and efficiency, ensuring robust security by leveraging nodes' interdependencies and capabilities without overburdening any single device.

3.3 Network Models

Network models, including centralized (cloud and SDN) and distributed (edge and fog computing) models, represent and structure data and business logic, as described below.

3.3.1 Centralized Models: Cloud Computing and Software-Defined Networking

Cloud computing and SDN utilize centralized control to improve efficiency and manageability. Cloud computing consolidates computer resources and services into data centers, enabling users to access and administer these resources via the Internet, hence promoting effective resource allocation and scalability. The cloud functions as a centralized repository for the consolidation of data from many devices, facilitating big data analytics, ML, and long-term storage. SDN is a software-driven networking architecture that consolidates network management by decoupling the control plane from the data plane, wherein a central SDN controller determines traffic routing while the data plane transmits the traffic. The centralization in SDN streamlines network administration and enhances adaptability. Collectively, these technologies enhance efficiency, scalability, and management in IoT and OT networks [18]. In a CI-enabled IoT or OT network, the cloud may assist global coordination across multiple edge and fog levels, ensuring that collective insights derived from dispersed intelligence are disseminated throughout the network. Hybrid cloud-edge architectures, enabling effective collaboration between cloud and edge devices, are gaining popularity since they combine the advantages of each [19].

3.3.2 Distributed Models: Edge and Fog Computing

In conventional cloud-centric designs, data from IoT or OT devices is sent to a centralized cloud for processing. However, the latency and bandwidth constraints render centralized cloud systems ineffective for real-time applications. Edge and fog computing are essential in enabling CI by localizing data processing near the sources of data generation. Edge computing delivers computational resources to the network's periphery, facilitating localized processing, expediting decision-making, and diminishing reliance on continuous cloud connection. Fog computing enhances this notion by including an intermediary layer between edge devices and the cloud, ensuring that only essential data is communicated to the cloud, while more regular or immediate operations are managed locally. This hierarchical design facilitates efficient resource allocation and reduced latency, essential for applications like smart grids, autonomous cars, and industrial IoT and OT systems.

3.4 Software Implementation Models for IoT

Software implementation models in IoT networks, including *Network Function Virtualization* (NFV), containerization, and Virtual Machines (VMs), are essential for improving the flexibility, scalability, and efficiency of IoT systems. These approaches facilitate the dynamic allocation and management of network resources, permitting the efficient deployment and scaling of IoT devices and services. Utilizing these technologies, IoT networks can process and analyze extensive data in real-time, enabling the integration and coordination of many devices and systems. This capacity is crucial for facilitating CI, wherein interconnected IoT devices cooperate to make informed judgments, enhance operations, and elevate overall system performance.

NFV is a network design paradigm that virtualizes comprehensive categories of network node functions, enabling their operation as software on conventional servers, storage systems, and switches. The separation of network services from proprietary hardware facilitates more flexible, scalable, and economical network management. Within the realm of IoT network security, NFV is especially pertinent as it facilitates the swift implementation and adaptive scaling of security functions, including firewalls, intrusion detection systems, and content filtering solutions. Through the virtualization of security functions, NFV improves the agility and reactivity of IoT networks, allowing for rapid adaptation to evolving threats while sustaining strong security measures.

VMs are software-based simulations of real computers that execute operating systems and applications like a physical computer. They let numerous VMs function on a single physical host, each segregated from the others, hence improving resource efficiency and adaptability. Within the realm of IoT network security, VMs are particularly pertinent as they facilitate the implementation of security services, including firewalls, intrusion detection systems (IDS), and antivirus software, in a virtualized setting. This isolation guarantees that if one VM is compromised, the others remain unscathed, hence improving the overall security stance of the IoT network. Moreover, VMs may be rapidly built, scaled, and administered, offering a resilient and flexible security solution for dynamic IoT ecosystems.

Containerization is a lightweight type of virtualization that encapsulates an application and its dependencies into a singular, portable container, guaranteeing uniform functionality across diverse settings. In contrast to VMs, containers utilize the host operating system's kernel while functioning in

distinct user areas, resulting in enhanced efficiency and expedited deployment. In the realm of IoT network security, containerization is significant as it bolsters security via isolation, hence diminishing the attack surface by confining potential breaches within separate containers. Moreover, container orchestration platforms such as Kubernetes offer inherent security functionalities including role-based access control (RBAC), network policies, and automatic vulnerability assessment, hence enhancing the security framework of IoT networks. [20] introduces an innovative architecture aimed at incorporating Human DT into the social IoT. This architecture utilizes containerization to effectively deploy and manage services, integrating Virtual Users (VUs) and Social Virtual Objects (SVOs) within a scalable Cloud/Edge infrastructure. Essential elements comprise a host controller for container orchestration, a deployer for automated service deployment, user clusters for consolidating VUs, SVOs, and apps to provide secure and efficient data sharing. The suggested system seeks to tackle issues related to scalability, efficiency, and automation, exhibiting enhanced performance in the management of high-volume installations relative to conventional platforms.

3.5 Network Frameworks for CI

Frameworks provide a systematic methodology for structuring software code. IoT / OT network frameworks for CI provide a systematic methodology for developing systems capable of autonomously managing, configuring, optimizing, and safeguarding themselves, using self-organization principles to improve resilience and adaptability.

Self-Organizing Architectures in IoT/OT networks provide a structural framework enabling devices to establish ad-hoc networks and collaborate to attain system-level objectives without predetermined roles or central coordination. This architectural model is optimal for facilitating continuous integration in extensive, dynamic settings in Self Organizing Systems (SOS, see Section 3.1.4 in D1.1) where centralized oversight is unfeasible [21].

The operational role of SOS may be summarized as follows [21]:

- *Self-Healing Networks*: In the event of a device failure in an IoT/OT network, remaining devices within the network may autonomously reorganize to sustain connection, allocating tasks or rerouting data to guarantee uninterrupted functioning. The self-healing capacity is essential for the robustness and resilience of IoT networks.
- *Resource Optimization*: In SOS, devices may autonomously negotiate the allocation of resources such as bandwidth, energy, or computational power, therefore optimizing resource use across the network. Devices may reallocate workloads in edge/fog computing systems to minimize latency and energy consumption.
- *Dynamic Topology Management*: In IoT networks including mobile devices, such as smart cars or drones, agents independently identify peers and establish connections without depending on a fixed infrastructure. Consequently, SOS may oversee dynamic network topologies by modifying connections when agents relocate, hence assuring optimal routing and data transmission.

Examples of SOS in IoT are:

- *Smart Grids*: In a smart grid, self-organizing IoT devices (e.g., smart meters, sensors) autonomously balance supply and demand in real-time by modifying energy consumption patterns or redistributing electricity according to local conditions, optimizing the energy network without centralized oversight.
- *Autonomous Vehicle Fleets*: SOS of connected vehicle networks enable cars to create dynamic platoons or modify routes according to real-time traffic information, improving road safety and optimizing traffic flow.
- *Drone Swarms*: A consortium of drones endowed with self-organizing skills may cooperate in search-and-rescue missions, whereby drones automatically synchronize search patterns, exchange information, and adjust to the environment in real-time.

Key benefits of SOS for CI in IoT/OT networks [22] are:

- *Resilience*: By removing single points of failures and allowing devices to autonomously adjust to failures or changes, SOS improve the resilience of the IoT/OT network.
- *Efficiency*: SOS facilitate more effective resource use, enabling devices to dynamically assign tasks according to real-time requirements, hence preventing the overloading of certain network components.
- *Scalability*: The decentralized decision-making and dynamic structure of SOS render it extremely scalable, which is essential for continuous integration in extensive IoT/OT implementations, such as smart cities or industrial IoT systems.

DLT technologies can be used to enhance security, privacy, and trust in self-organizing IoT ecosystems. DLT-based approaches can help ensure non-repudiation of actions among IoT devices by using DLT to securely manage and verify interactions and data exchanges, contributing to collective intelligence by improving the reliability and integrity of IoT networks. DLT technologies can be combined with AI, for continuous monitoring or event-driven inference, to identify security threats and anomalies in real-time and / or to retrospectively investigate incidents, supporting resilient and autonomous IoT ecosystems. Complementing distributed event logging with computing event anomaly scores is in the plan of CISSAN T4.3.

4 CISSAN Initial Architecture

4.1 CISSAN Framework

How CISSAN results can be used by the stakeholders is heavily use case-specific. It depends on the maturity of a target IoT or OT system / network, application domain, threat and risk prioritization, technology choices, and other factors. In particular, one has significantly greater freedom for integrating security controls to systems and networks at the design stage, when sophisticated CI-based mechanisms can be planned and implemented, while choices at the operational stage are usually limited. Thus, the initial architecture is unavoidably quite open-ended and shaped depending on layered frameworks and core processes, functions, elements, and lifecycle stages.

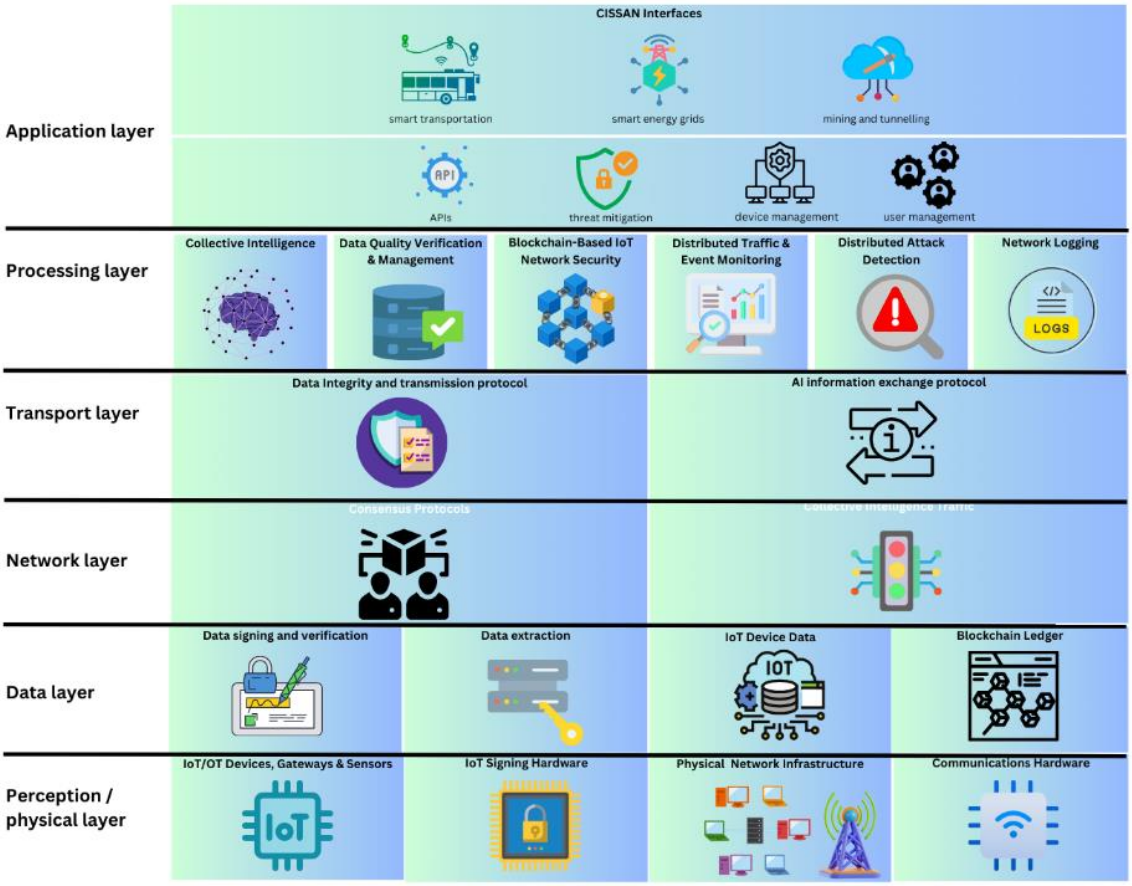

Figure 4. Layered architecture of CISSAN framework

The CISSAN framework provides a comprehensive and holistic view of the security requirements and challenges for IoT/OT environments. The CISSAN framework has a layered structure consisting of six layers: the perception/physical layer, the data layer, the network layer, transport layer, the processing layer, and the application layer (Figure 4). Each layer, containing several components, has a specific role and function in the framework, and represents a level of abstraction and granularity for the IoT/OT data and processing.

To guide the design and implementation of security solutions, the CISSAN framework should help elaborate a security model for IoT/OT environments, with four main cybersecurity functions: Detection, Response, Protection, and Intelligence. Each function is composed of several sub-functions that comprise specific security activities and objectives.

- **Detection**: The function to detect cyberattacks in IoT and OT environments using various methods and techniques, such as anomaly detection and signature-based (or rule-based) detection through network traffic analysis, device-behaviour and user-behaviour analyses. The sub-functions of detection are:

- *Meta Data*: The process of extracting relevant information from network traffic, such as the source, destination, protocol, payload, and impact of data packets. Similar approaches can be applied to events in an endpoint device, such as timings, parent-child process chains, etc.
- *Labeling*: The process of assigning labels to network traffic, such as normal, suspicious, malicious, or unknown, based on the analysis of relevant metadata and the comparison with baseline/normal models and profiles and threat intelligence. Similar approaches can be applied to events in an endpoint device.
- *Source/Impact*: The process of identifying the source and the impact of network traffic, such as the device, the service, the vulnerability, or the threat that generated or affected the observed data packets.
- *Settings Management*: The process of managing the settings and parameters of a detection function, such as thresholds, rules, policies, and alerts.
- *Reporting*: The process is designed to report the results and findings of a detection function, such as the metadata, labels, sources, impacts, and alerts, to the relevant stakeholders and systems, such as users, response functions, or cloud backends.

- **Response**: The response function to cyberattacks and anomalies in IoT/OT environments using various methods and techniques, such as automated actions, manual actions, or collective actions. The sub-functions of response are:
  - *Self/Collective Awareness*: The process of developing and keeping awareness about the current state of an IoT/OT environment, including devices, services, vulnerabilities, threats, and incidents, and sharing this information with other systems and stakeholders, such as cloud backends, the protection function, or the intelligence function.
  - *Automated Response*: The process of executing appropriate predefined actions to mitigate or prevent cyberattacks, such as blocking, isolating and/or quarantining affected devices, operating system processes or data packets, or applying patches or updates to devices or services.
  - *Reducing Attack Surface*: The process of reducing the exposure and the risk of an IoT/OT environment, such as disabling or removing unnecessary or unused devices, services, or protocols, or enforcing secure configurations and policies for devices and services.
  - *Deny/Restrict*: The process of denying or restricting the access or the communication of devices or services, such as implementing authentication, authorization, encryption, or firewall rules, or applying whitelisting or blacklisting policies.
  - *Configuration*: The process of configuring and tuning the settings and parameters for the response function, such as actions, rules, policies, and alerts for the sub-functions of automated response, attack surface reduction, or deny/restrict.
- **Protection**: The function of protecting an IoT/OT environment from cyberattacks, using various methods and techniques, such as device security, network security, or cloud security. The sub-functions of protection are:
  - *Identify posture improvement*: The process of identifying and assessing the current security posture of an IoT/OT environment, including devices, services, vulnerabilities, threats, and incidents, and suggesting improvements and recommendations to enhance the security level and performance.
  - *Initiate change*: The process of initiating and implementing changes and improvements to an IoT/OT environment, such as installing or upgrading devices or services, or applying patches or updates.
  - *Implement protection*: The process of implementing and enforcing protection measures and mechanisms for an IoT/OT environment, such as device security, network security, or cloud security.
  - *Enterprise Posture Management*: The process of managing and monitoring the security posture of an IoT/OT environment, including devices, services, vulnerabilities, threats, and incidents, and reporting the status and the results to

relevant stakeholders and systems, such as users, cloud backends, or the intelligence function.

- **Intelligence**: The function of providing and consuming intelligence (information and insights) for an IoT/OT environment, using various methods and techniques, such as threat intelligence, vulnerability intelligence, or CI. The sub-functions of intelligence are:
  - *Internal/External Threat Intelligence*: The process of collecting, analyzing, and sharing threat information and indicators from internal or external sources, such as network traffic, devices, services, cloud backends, or third-party providers.
  - *Vulnerabilities*: The process of collecting, analyzing, and sharing vulnerability information and indicators from internal or external sources, such as network traffic, devices, services, cloud assets, backends, or third-party providers.
  - *Collective Intelligence*: The process of collecting, analyzing, and sharing intelligence (information and insights) from multiple sources and domains, such as IoT, OT, IT and cloud environments.
  - *Protection Engineering*: The process of applying intelligence (information and insights) to the protection function, such as identifying and assessing the security posture, initiating and implementing changes and improvements, or implementing and enforcing protection measures and mechanisms.

Figure 5 shows a typical mapping between security functions and core elements of IoT / OT systems and environments introduced in Section 4.2 below.

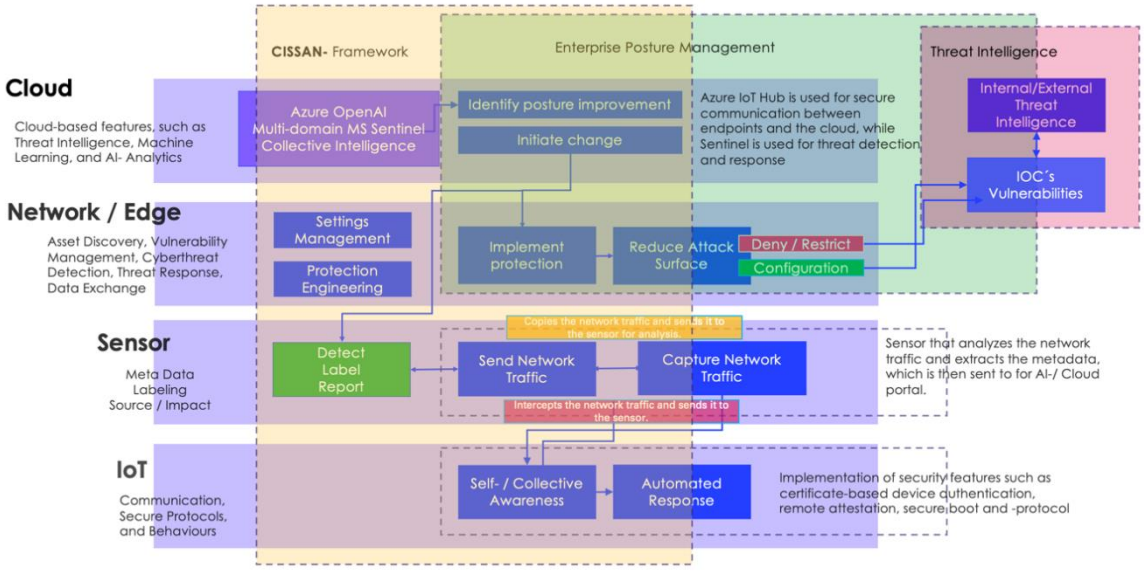

Figure 5. Key cybersecurity control points and functions at device, sensor, network / edge, and cloud levels

4.2 Core Elements in Initial Architecture

IoT and OT devices, such as sensors, actuators, cameras, or smart meters, are the endpoints to collect, process, and exchange data over the Internet and other communication networks. Depending on the device capabilities and the use case domain, cybersecurity functions at the device level can include certificate-based authentication, remote attestation, secure boot, and secure protocols to prevent data theft, tampering, and spoofing. For instance, an IoT sensor is a specific type of device (or part of a device) that detects events or changes in its environment, sending collected data to an IoT gateway, other edge devices, other IoT devices, or cloud backends. IoT sensors directly interact with the physical world. For example, a temperature sensor in a smart thermostat collects data about the room temperature and sends it to the system to adjust heating. IoT node refers more broadly to any physical device within an IoT system, which includes sensors but also other components such as actuators, cameras, and gateways. An IoT node can be as simple

as a sensor or as complex as a gateway that aggregates data from multiple sensors and performs certain data processing before sending it to the cloud backend or other systems.

IoT and OT devices often have limited resources to run security functions, or their vendors simply do not make it possible to integrate security functions into such devices. To enhance security in such scenarios, security sensors can be deployed in IoT/OT environments to analyse network traffic from other devices. These sensors can intercept or duplicate the traffic, label it, and extract metadata for further use in AI/ML-based cybersecurity solutions. ML models can also be trained and do inference in clouds, endpoints, or edges (e.g., gateway devices). There are multiple options for training, including local training, aggregation of local data in a cloud, and aggregation of locally trained models in a cloud, or in a group of devices. While inference is usually done either in a cloud or locally, the results of local inference in multiple devices can be combined/aggregated further or inference tasks can be distributed among multiple devices or delegated to other devices[1]. Various forms of aggregation, distribution, and delegation of training and inference tasks (and more broadly other security tasks) can be considered CI. Identifying and implementing CI for relevant use cases (including the project use cases) is on the CISSAN agenda. CISSAN is exploring the possibility of using the security functionality of IoT/OT nodes jointly with security sensors, for example, to detect or request blocking of peer-to-peer traffic between nodes that exhibit malicious activities. In addition to ML-related tasks, reporting and sharing threat information among IoT/OT nodes is also a form of CI supporting security awareness in IoT/OT environments. Nodes and security sensors can be used to initiate and implement protection and mitigation measures, such as blocking, isolating, or device patching, to reduce the attack surface and prevent further damage.

An example of a security sensor is network tap, which is a device that captures the network traffic from IoT/OT devices and sends it for analysis. It can be either a physical device installed in a network infrastructure, e.g., a switch or a router, or a piece of software installed in an IoT/OT or other devices as a virtual machine or a container. A network tap can be either a passive device that only copies the network traffic and sends it for analysis or an active device that can also intercept and modify the network traffic, such as a firewall or a proxy. It can capture the network traffic from either a single IoT/OT device or multiple devices, e.g., from a network segment or a subnet.

Edge is a distributed and local computing platform that provides certain services and resources for IoT and OT environments, including data processing, data analysis, and data exchange. CISSAN can benefit from such edge-based features (parts of the framework) as Asset Discovery, Vulnerability Management, Cyberthreat Detection, Threat Response, Data Exchange, Data Filtering, Data Aggregation, and Data Compression. Examples of edge device types are gateway, router, and switch. As edge devices can also be a target of a vector of cyberattacks, they need certain security features, such as access control, encryption, or firewall.

Cloud can be used to support leveraging AI and ML techniques to enhance the security of IoT and OT environments[2]. AI and ML can be instrumental in analyzing large and complex data, detecting and classifying known and unknown cyberattacks, providing situational awareness and risk assessment, and automating and optimizing attack response and mitigation[3]. Threat analysis, implementation, integration, and operation of cloud environments have to be taken into account, since they can also be a target of a vector of cyberattacks.

4.3 CISSAN vs. Current State-of-the-Practice

There are some security solutions for IoT and OT security in the market, such as Cisco IoT Security, IBM Watson IoT Platform, Symantec IoT Security, McAfee Secure Home Platform, Fortinet Security Fabric, Microsoft Defender for IoT, Darktrace Unified OT Protection, and Tenable OT Security. However, most of these solutions are either focused on specific aspects of security, such as device authentication, encryption, or firewalling, or on specific domains, such as smart home, smart city, or smart factory. Moreover, most of these solutions are either based on traditional security methods, such as signature-based detection, rule-based policies, or manual response, or on cloud-based

[1] For example, the scope of an attack detection ML model can vary from an individual device to a group of devices to a network or even a group of networks.

[2] While many OT environments are still not connected with clouds, OT operators are increasingly considering options for getting cloud computing benefits.

[3] Of course, the use of AI and ML in IoT/OT security also presents challenges, such as the need for good data sources; efficient and scalable data processing; regular model updates; integration with existing security tools; and ethical and legal compliance.

security methods and capabilities, including cloud computing, cloud storage, and cloud services. We can mention certain limitations and drawbacks of the existing security solutions:

- *Lack of scalability and flexibility*: Difficulties in coping with the high numbers and variety of IoT/OT devices and environments and in adapting to changing and evolving threats and attacks.
- *Lack of intelligence and awareness*: Existing solutions cannot analyse and understand the complex and dynamic behaviour and context of IoT and OT devices and environments, and cannot detect and label unknown and advanced threats and attacks.
- *Lack of automation and coordination*: Difficulties in responding and reacting to IoT and OT threats and attacks in a timely and effective manner, and in coordinating and synchronizing the protection and mitigation measures across devices and environments.

To address these limitations and drawbacks, CISSAN provides a novel and comprehensive framework and a collection of IoT/OT security enablers that leverage the powers of AI, ML, CI, and threat intelligence. CISSAN offers the following advantages and benefits:

- *Higher scalability and flexibility*: CISSAN security mechanisms will be deployed in different scenarios, applications, and points, such as cloud, edge, or fog computing, and the mechanisms will be able to interact with different devices, systems, and users.
- *Higher collective intelligence and awareness*: Deeper analysis and understanding will be based on the use of AI and ML. CISSAN security mechanisms will be able to report and share threat information with other IoT/OT devices and environments, creating collective security intelligence and awareness.
- *Higher automation and coordination*: CISSAN will support security function, task distribution, delegation, and aggregation of results, improving the response effectiveness. CISSAN security mechanisms will be able to initiate and implement protection and mitigation measures, such as blocking, isolating, patching, updating, configuring, or restoring devices, and to reduce the attack surface for preventing further damage.

5 Key Lines of the CISSAN Efforts

In this section, we present an initial view of the key lines of the project efforts. The objectives and priorities for the efforts will be monitored and adjusted throughout the project's lifecycle.

5.1 Local Anomaly Detection at IoT/OT Nodes and Aggregation in the Backend

- The project aims to develop and deploy local anomaly detection models in IoT/OT nodes, which can detect communication and process anomalies based on network traffic and sensor measurements.
- The anomaly detection methods are based on various techniques, such as time series analysis, low-dimensional projections, and neural networks.
- The anomaly scores produced by the local models are sent to the backend, where they can be further analysed and aggregated using rule-based logic, ML, or visualizations.
- The project also seeks to make the anomaly scores or their aggregate values explainable and interpretable for the human operator.
- The project acknowledges that the anomaly detection methods are not always focused on or tuned for cybersecurity and cyberattacks, and that it is ultimately up to the operator how to handle anomaly alerts and scores.

5.2 Traditional Data Collection and Response in Security Sensors and Backend

- In cases where IoT/OT nodes have no or limited security functionality, or data might be compromised, security sensors in the edge analyse the network traffic, extract metadata, and then send the results to the cloud for further analysis and decision-making.
- The project explores the possibility of using the security functionality of IoT/OT nodes jointly with security sensors, for example, to detect or request blocking of peer-to-peer traffic between nodes suspected of malicious activities.
- The project also investigates the scope and granularity of ML models in the backend, which can be applied to individual devices, groups of devices, individual networks, and groups of networks.
- The project also articulates the response functionality of security sensors, which can take actions to mitigate or stop attacks, such as blocking traffic, isolating devices, or alerting the operator.

5.3 Use of Blockchain and Sensor Data Signing

- The project leverages the use of blockchain and Lightning Network for securing data transfers in multi-sensor monitoring IoT networks, where data is transferred via multiple nodes.
- The objective is to detect and prevent unauthorized data deletion and tampering, both in intermediate nodes and in the cloud, by recording any data sent by a node in a blockchain and verifying its presence and integrity in the backend (or by third parties).
- The project also uses blockchain-based methods for recording and verifying actions carried out by IoT and OT nodes, including node interactions with users, which can support forensic investigations and public auditing of IoT and OT systems.
- The project also uses hardware-based solutions (e.g., Infineon chips) for data signing in IoT nodes, which can provide sensor data authentication and ensure its origin and validity.
- The project will consider the use of blockchain-based methods for creating incentives for nodes to participate in FL and other collaborative schemes, where nodes share their local data or model updates with other nodes or the backend. Note that this approach may make sense only when there are multiple collaborating owners of IoT/OT nodes and backends. Therefore, finding a good UC is prerequisite for considering such an approach.

5.4 Distribution of Security Functions

- The project is in search of a technology for optimizing the distribution of computing functions, such as security functions, among nodes, to avoid the need to place the code of all the functions in all the nodes.
- The function specifications and the relevant parameters of a system, such as node resources, communication channels, constraints, etc. will thus determine how to distribute the functions across the nodes.
- The project faces the challenge of finding good UCs where such a capability can be valuable and validated. Distributing ML functionality is currently under consideration (for both training and inference).

5.5 Sensor Data Analysis in the Backend

- The project conducts sensor data quality assessment in the backend, to detect the abuse of sensor deployment and operating rules and practices, typically by the operators responsible for installing and maintaining the sensors.
- The project notes that sensor data quality assessment is not always related to cyberattacks but can be crucial in audits and investigations of the abuse of contractual responsibilities.
- The project also considers the possibility of using sensor data analysis for other purposes, such as detecting environmental changes, optimizing resource consumption, or improving service quality.

5.6 Collective Intelligence

- The project investigates approaches for security task delegation, including protocols and mechanisms suitable for peer-to-peer node communication and how they can be used for running security tasks collaboratively, such as malware scanning, storing and exchanging attack-related information, or coordinating responses.
- The ability to evaluate the trustworthiness and reputation of a node is an important ingredient in task delegation, and potential approaches to that are currently under investigation.
- The project may explore the use of FL for attack or anomaly detection, where nodes collaboratively train a model by iteratively updating it locally and sending the updates to the backend or other nodes. The project evaluates the advantages and challenges of this approach for considered UCs.

5.7 Use of GANs for Data Generation

- The project investigates the use of GANs for generating synthetic normal and anomalous data for training attack and anomaly detection models.
- This can help address the challenge of the lack of rich and diverse training data, which is a major obstacle in using ML for near-real-time or online attack detection.
- The project also considers the possibility of using other generative models, such as GPT-based approaches, which may produce more realistic and diverse data.

5.8 Asset Discovery, Vulnerability Management, Certificate-based Device Authentication, Remote Attestation, Secure Boot, etc.

- The project integrates and improves various methods and tools for asset discovery, vulnerability management, certificate-based device authentication, remote attestation, secure boot, and other security functions.
- Some of these methods and tools can be found in open-source repositories or the toolkits of selected CISSAN partners (e.g., Netox, Bittium, Clavister, and Savantic).
- The project evaluates the effectiveness and efficiency of these methods and tools in improving the security and resilience of IoT and OT systems.

5.9 Dealing with AI-powered Attacks, Stealthy Detection Functionality

- The project monitors the developments and trends in the use of AI in real-world cyberattacks. It assesses the feasibility and impact of such attacks on the CISSAN research and development domains.
- The project may explore adapting the CISSAN methods and solutions for countering the use of AI by the attackers, for instance, by hiding or obfuscating the attack detection functionality.
- The project will also explore the possibility of using AI-powered attacks to test and improve the CISSAN methods and solutions.
- It should be noted that apart from social engineering and – to a lesser extent – reconnaissance, there is almost no evidence yet of the use of AI in real-world cyberattacks. We note, however, that countering social engineering is not on the CISSAN's agenda.

References

[1] N. Mohamudally, "Paving the way towards collective intelligence at the IoT edge," *Procedia Computer Science,* vol. 203, pp. 8-15, 2022.

[2] R. Lazzarini, H. Tianfield and V. Charissis, "Federated Learning for IoT Intrusion Detection," *AI,* vol. 4, p. 509–530, 2023.

[3] M. A. Ferrag, O. Friha, L. Maglaras, H. Janicke and L. Shu, "Federated Deep Learning for Cyber Security in the Internet of Things: Concepts, Applications, and Experimental Analysis," *IEEE Access,* vol. 9, pp. 138509-138542, 2021.

[4] R. Chaudhary, R. Kumar and N. Saxena, "A systematic review on federated learning system: a new paradigm to machine learning," *Knowledge and Information Systems,* 2024.

[5] G. Xia, J. Chen, C. Yu and J. Ma, "Poisoning Attacks in Federated Learning: A Survey," *IEEE Access,* vol. 11, pp. 10708-10722, 2023.

[6] L. Lyu, H. Yu, J. Zhao and Q. Yang, "Threats to Federated Learning," *Lecture Notes in Computer Science (LNCS),* vol. 12500, 2020.

[7] W. Lepuschitz, *Self-Reconfigurable Manufacturing Control based on Ontology-Driven Automation Agents,* Vienna, Austria: Technische Universität Wien, 2018.

[8] R. Coulter and L. Pan, "Intelligent agents defending for an IoT world: A review," *Computers & Security,* vol. 73, pp. 439-458, 2018.

[9] P. Hoen and S. M. Bohte, "COllective INtelligence with Sequences of Actions," in *Machine Learning: ECML 2003. Lecture Notes in Computer Science (LNCS)*, Berlin, Heidelberg, 2003.

[10] R. Owoputi and S. Ray, "Security of Multi-Agent Cyber-Physical Systems: A Survey," *IEEE Access,* vol. 10, pp. 121465-121479, 2022.

[11] M. Gheysari and M. Tehrani, "The Role of Multi-Agent Systems in IoT," in *Multi Agent Systems: Technologies and Applications towards Human-Centered*, S. Gupta, I. Banerjee and S. Bhattacharyya, Eds., Singapore, Springer, 2022, p. 87–114.

[12] K. Iniewski, IoT and Low-Power Wireless: Circuits, Architectures, and Techniques (Devices, Circuits, and Systems), C. Siu, Ed., CRC Press, 2018.

[13] G. BB and Q. M., "An overview of Internet of Things (IoT): Architectural aspects, challenges, and protocols.," *Concurrency Computation Practice and Experience,* 2020.

[14] C. Hight and C. Perry, Eds., Collective Intelligence in Design, Wiley, 2006.

[15] J. Musil, A. Musil and S. Biffl, "SIS: an architecture pattern for collective intelligence systems," in *20th European Conference on Pattern Languages of Programs (EuroPLoP '15)*, Kaufbeuren, Germany, 2015.

[16] A. Vengerov, "AN ARCHITECTURAL PATTERN OF COLLECTIVE LEARNING AND COLLECTIVE INTELLIGENCE," *International Journal of Arts & Sciences,* vol. 4, no. 20, pp. 87-100, 2011.

[17] T. Godquin, M. Barbier, C. Gaber, J.-L. Grimault and J.-M. L. Bars, "Applied graph theory to security: A qualitative placement of security solutions within IoT networks," *Journal of Information Security and Applications,* vol. 55, 2020.

[18] T. Rajmohan, P. Nguyen and N. Ferry, "A decade of research on patterns and architectures for IoT security," *Cybersecurity,* vol. 5, no. 2, 2022.

[19] T. D. P. Bai and S. Rabara, "Design and Development of Integrated, Secured and Intelligent Architecture for Internet of Things and Cloud Computing," in *2015 3rd*

International Conference on Future Internet of Things and Cloud, Rome, Italy, 2015.

[20] R. Girau, M. Anedda, R. Presta, S. Corpino, P. Ruiu, M. Fadda, C.-T. Lam and D. Giusto, "Definition and implementation of the Cloud Infrastructure for the integration of the Human Digital Twin in the Social Internet of Things," *Computer Networks*, vol. 251, 2024.

[21] W. Banzhaf, "Self-organizing Systems," in *Encyclopedia of Complexity and Systems Science*, R. Meyers, Ed., New York, NY., Springer, 2009, p. 8040–8050.

[22] Y. Ding, Y. Jin, L. Ren and K. Hao, "An Intelligent Self-Organization Scheme for the Internet of Things," *IEEE Computational Intelligence Magazine*, vol. 8, no. 3, pp. 41-53, 2013.