Topic: HORIZON-MSCA-2022-SE-01-01

Call: HORIZON-MSCA-2022-SE-01

Type of Action: HORIZON-TMA-MSCA-SE

AI-ASsisted cybersecurity platform empowering SMEs to defend against adversarial AI attacks

AI-ASsisted cybersecurity platform empowering SMEs to defend against adversarial AI attacks

WP2 – Architecture Design, Requirements, and Data

D2.2 - Specifications & Business cases

Editors	ISI
Authors	Athanasia Sampazioti (UPRC), Ignacio Lacalle Úbeda (UPV), Giorgio Bernardinetti (CNIT), Maria Niculae (BEIA), Georgios Kalpaktsoglou (FOGUS), Cristina Alcaraz (UMA), Javier Lopez (UMA), Iman Hasnaouia Meskini (UMA), Ruben Ríos (UMA), Ilias Politis (ISI), Tafseer Akhtar (ISI), Cosmina Stalidi (BEIA)
Dissemination Level	PU
Type	R
Version	2

Project Profile	
Contract Number	101131292
Acronym	AIAS
Title	AI-ASsisted cybersecurity platform empowering SMEs to defend against adversarial AI attacks
Start Date	Jan 1st, 2024
Duration	48 Months

Partners

University of Piraeus Research Center	EL
BEIA CONSULT INTERNATIONAL SRL	RO
UNIVERSIDAD DE MALAGA	ES
K3Y	BG
ATHINA-EREVNITIKO KENTRO KAINOTOMIAS STIS TECHNOLOGIES TIS PLIROFORIAS, TON EPIKOINONION KAI TIS GNOSIS	EL
SUITE5 DATA INTELLIGENCE SOLUTIONS LIMITED	CY
CONSORZIO NAZIONALE INTERUNIVERSITARIO PER LE TELECOMUNICAZIONI	IT
FOGUS INNOVATIONS & SERVICES P.C	EL
UNIVERSITAT POLITECNICA DE VALENCIA	ES
PDM E FC PROJECTO DESENVOLVIMENTO MANUTENCAO FORMACAO E CONSULTADORIALDA	PT

Document History

VERSIONS

Table 1 Document history

Version	Date	Autho r	Remarks
0.1	9/1/2025	ISI	Initial ToC
0.2	15/2/2025	ISI	Industrial Networks use case, Methodology
0.3	31/03/2025	UMA	Hospital Environmental Monitoring (based on IoT devices)
0.3	31/03/2025	CNIT	Weaponizer Use Case
0.4	21/5/2025	K3Y	Industrial Networks use case updated
0.5	30/5/2025	ISI	Internal Review version
0.6	05/06/2025	BEIA	Contributions to Sections 3.1 & 4.1
1	13/06/2025	ISI	First version
1.1	20/06/2025	ISI	Review
2	30/06/2025	ISI	Final version

AIAS message

Executive Summary

This deliverable presents the detailed definition and analysis of the AIAS project's pilot use cases. These use cases will be instrumental in validating the technical components, architectural principles and functional requirements defined in D2.1. As outlined in the previous deliverable, the architecture and user-centric specifications have been built upon. In this regard, this document identifies real-world application scenarios in which AIAS will demonstrate its capacity to detect, deceive and mitigate adversarial AI attacks in diverse operational environments. The use cases selected for this deliverable reflect the core pillars of the AIAS platform: adversarial AI execution, defence, deception mechanisms, explainable AI (XAI)-based decision support, and secure data fusion. The following representative use cases are analysed in depth:

- ***Hospital Environmental Monitoring***.
- ***Industrial Network Security***.
- ***Weaponizer-Enhanced Malware Detection***.
- ***SME Providing Digital Services***.

Each use case is developed following a structured methodology that ensures alignment with AIAS's research and technical objectives. The methodology encompasses a range of activities, including stakeholder-driven requirements analysis, identification of attack vectors and vulnerabilities, and integration pathways with the AIAS platform components. Hence, the deliverable serves as a cornerstone for validating AIAS's holistic approach to "AI for Cybersecurity" and "Cybersecurity for AI," ensuring that the platform's innovative features are exercised in practical, impactful scenarios. These use cases will guide the future development, integration, and evaluation efforts outlined in subsequent deliverables across WP 3, 4, and 5.

Table of Contents

Table of Figures

Table of Tables

Table 2 Abbreviation Table

Abbreviation	Description
UPnP	Universal Plug and Play Protocol
SSH	Secure Shell
RFI	Remote File Inclusion
MHN	Modern Honey Network
ICS	Industrial Control System
ICT	Information Communication Technology
IoT	Internet of Things
IoV	Internet of Vehicles
AI	Artificial Intelligence
XAI	Explainable AI
API	Application Programming Interface
SQL	Structured Query Language
ML	Machine Learning
SME	Small and Medium-sized Enterprise
GMM	Gaussian Mixture Models
PCA	Principal Component Analysis
KNN	K-Nearest Neighbours
LIME	Local Interpretable Model-Agnostic Explanations
AIDM	AI-based Detection Module
LLRL	LifeLong Reinforcement Learning
SHAP	SHapley Additive exPlanations
IPFS	InterPlanetary File System
GDPR	General Data Protection Regulation
MTTR	Mean Time To Recovery
GAN	Generative Adversarial Network
TTPs	Tactics, Techniques, and Procedures
AI2EM	Adversarial AI Engine Module
DNN	Deep Neural Network
DoS	Denial of Service
IDS	Intrusion Detection System
UEBA	User and Entity Behavior Analytics
IDPS	Intrusion Detection and Prevention Systems
SIEM	Security Information and Event Management

UI/UX	User Interface/ User Experience
AAA	Authorization, And Accounting
OSS	Open-Source Software
RNN	Recurrent Neural Network
NLP	Natural Language Processing
URL	Uniform Resource Locator
HTML	HyperText Markup Language
XML	Extensible Markup Language
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart
TOR	The Onion Routing
I2P	Invisible Internet Project
CTI	Cyber Threat Intelligence
OSINT	Open-Source Intelligence
SOC	Security Operations Center
MTD	Moving Target Defence
CID	Content IDentifier
XSS	Cross Site Scripting
RDF	Resource Description Framework
MITRE ATT&CK	MITRE Adversarial Tactics, Techniques and Common Knowledge
STIX	Structured Threat Information Expression
CAPEC	Common Attack Pattern Enumeration and Classification
CVE	Common Vulnerabilities and Exposures
CWE	Common Weakness Enumeration
PCAP	Packet CAPture
CICIDS	Canadian Institute for Cybersecurity - Intrusion Detection Evaluation Dataset
XPath	XML Path Language
LSTM	Long Short-Term Memory
JSON	JavaScript Object Notation
XML	Extensible Markup Language
YAML	Yet Another Markup Language
AJAX	Asynchronous JavaScript and XML
SQL	Structured Query Language
DT	Digital Twin
VP	Virtual Persona

MQTT	Message Queuing Telemetry Transport
HTTP	Hypertext Transfer Protocol

1. Introduction

Deliverable D2.2 is dedicated to the specification and analysis of the pilot use cases that will serve as validation contexts for the AIAS platform. This document is a continuation of the architecture and requirements analysis provided in D2.1 [D2.1], focusing on grounding the AIAS [AIA] design and functionalities in real-world scenarios. These use cases will provide the practical foundation for the design, development, and evaluation of the platform's components and will ensure that all AIAS modules (deception, adversarial AI, detection, mitigation, XAI) are tested under realistic threat landscapes and stakeholder-driven requirements. The objectives of this deliverable are as follows:

- To define and describe representative and diverse use cases that reflect AIAS’s vision and research objectives.
- To align these use cases with the technical architecture and functional requirements specified in D2.1.
- To develop a common methodology for use case selection, analysis, and validation.
- To establish a reference framework for testing and evaluating the effectiveness, efficiency, and resilience of the AIAS platform in practical contexts.

1.1.Relation to other deliverables

This deliverable is directly related with the following deliverables:

- ***D2.1 – Requirements and Reference Architecture:*** The use cases defined in this deliverable comply with the user, functional, and non-functional requirements established in D2.1, ensuring technical and operational coherence.
- ***D3.1 – AIAS Deception Layer:*** The deception components (e.g., honeypots, digital twins, virtual personas) to be developed will be tested within the environments described in the use cases.
- ***D3.3 – Adversarial AI Engine:*** The adversarial scenarios detailed in the use cases will guide the development of targeted attack simulations for robust testing of AI models.
- ***D4.1 – AI-Based Detection of Adversarial Attacks:*** The use cases define the operational settings in which detection capabilities will be validated.
- ***D4.2 – Mitigation of Adversarial AI Attacks & XAI:*** The mitigation engine and human-in-the-loop explainability features will be exercised and evaluated within the defined scenarios.
- ***D5.1 – Platform Integration:*** Use cases will guide the integration sequence and validation milestones of the AIAS components.
- ***D5.2 – Platform Evaluation:*** The final system validation and performance assessment will be based on the successful implementation of these use cases.

1.2.Document structure

The remainder of this document is organised as follows:

- ***Section 2*** provides a high-level overview of the AIAS use cases, describing their relevance and strategic alignment with project objectives.
- ***Section 3*** presents the methodology adopted for the selection and definition of use cases, including criteria, stakeholder inputs, and threat modelling.
- ***Section 4*** offers an in-depth analysis of each use case, including their technical environment, attack surface, adversarial AI scenarios, and expected outputs.
- ***Section 5*** summarizes the key conclusions and outlines how these use cases will be leveraged in the next stages of platform development and evaluation.

2. AIAS Use Case High-Level Description

2.1. Use Case 1: Application of AIAS in Environmental Monitoring

This section describes two distinct scenarios of the principal Environmental Monitoring UC. The first one (UC Case 1a) is defined for monitoring of environmental conditions, where a set of Internet of Things (IoT) devices are deployed, monitored, and replicated by a Digital Twin (DT). The second approach (UC Case 1b) studies leveraging Virtual Persona (VP) for the simulation of user behaviour within DT systems. The application scenario comprises a specific UC based on healthcare spaces such as hospitals. In contrast, the first UC is focused on a more holistic and generic approach of environmental settings.

2.1.1. Use Case 1a: IoT-based DT for Environmental Monitoring

The IoT-based DT for Environmental Monitoring UC is primarily focused on generic scenarios where DT generates valuable insights for the protection of systems such as SMEs, external/internal physical spaces, automation scenarios, etc. The proposed DT integrates a network of autonomous IoT devices for the continuous monitoring and adjustment of critical ambiental parameters including, among others, temperature, humidity, air quality, lighting, and access control.

From the cybersecurity standpoint, this approach provides AIAS platform with the necessary mechanisms to simulate, study, and evaluate a wide range of threat scenarios, leveraging tools for deception and testing applications. For instance, the system can improve the protection against adversaries that target IoT devices attempting to alter the environmental conditions and compromising the comfort and welfare of individuals. In this direction, the UC demonstrates the power of smart technologies, like DTs, to improve safety, reliability, and robustness of working/living spaces where IoT devices leverages automation.

The physical part of the UC, i.e., the IoT devices, integrates different communication protocols (e.g., HTTP) and follows a decentralized model, ensuring modularity, scalability, and robustness. It enables the design of future solutions and the analysis of advanced studies within AIAS, where real-time control issues regarding environmental conditions is the main requirement.

2.1.2. Use Case 1b: Hospital Monitoring through VP-assisted DT

Maintaining rigorous control over environmental factors such as temperature, humidity, and air quality is essential for ensuring safety and achieving optimal patient outcomes in critical clinical environments like surgical operating rooms (ORs).

The architecture of the Virtual Persona (VP)-assisted Digital Twin (DT) addresses this need by generating a virtual replica of the operating room environment that continuously monitors and upholds rigorous standards.

The DT integrates real-time sensor data, such as climatic measures, air quality indices, oxygen levels, and occupancy, into a model that evolves over time. The VP introduces a cognitive layer that contextualizes this data effectively. The VP can employ role-based reasoning to examine issues from the perspectives of various stakeholders, such as medical doctors, nurses, patients, IT administrators, and potential attacker profiles. This information allows for early detection of anomalies by applying machine learning models to highlight patterns that differ from the norm, facilitating customized responses that consider the circumstances. The system can identify an unexpected increase in CO2 levels or an unauthorized access attempt, next generating an appropriate response that adjusts to the situation.

The VP uses a knowledge base to provide specific guidance or alerts relevant to the active role, such as a virtual nurse recommending adjustments to ventilation or a virtual IT admin issuing a security alert. This innovative virtual monitoring layer enables individuals to continuously interact with their environment, transforming passive sensor observation into responsive interpretation and guidance. The VP-DT framework serves as an intelligent protector of the Operating Room environment. It links raw data with clinical decision-making to ensure that conditions remain within precise safety parameters.

To address this aspect, the AIAS platform can be based on VP-assisted DT, in charge of charging real-time data processing and behavioural monitoring of the environment where different stakeholders arise, such as medical staff (medical doctors, nurses), IT administrators, patients, and even the profile of attackers. These actors interact with the environment in different levels, for example, doctors and nurses provide patients an acceptable environmental state where patients need to coexist according to their diseases or treatments, while patients can also regulate their own environment according to their comfort.

2.2.Use Case 2: Application of AIAS in industrial networks

The use case within this framework illustrates AIAS's capacity to discern and counteract adversarial AI-driven threats in Modbus-based industrial networks. The system is implemented within an ICS environment comprising SCADA systems, PLCs, remote terminal units (RTUs), and industrial sensors that communicate over Modbus TCP/IP. First, the ***AI-Driven Detection Module*** perpetually observes network traffic, extracting Modbus function codes, transaction identifiers, and payload structures for subsequent analysis. Based on the Modbus/TCP network traffic data, the ***deep packet inspection (DPI) engine*** receives this data and generates network flow statistics, leveraging CICFlowMeter. CICFlowMeter is a network traffic flow generator that reads a PCAP file, then extracts flow-based features from the traffic and outputs the results as a CSV file for further analysis. Then these statistics are passed to the ***Adversarial AI Engine***, which is responsible for detecting relevant Modbus/TCP attacks. Based on P. Radoglou-Grammatikis et al. [RAD], the described Modbus/TCP cyberattacks are considered by the ***AI-Driven Detection Module***. These specific cyberattacks are further detailed in the next section. Once the attack is detected, the security event generation module receives the detection outcomes and generates the relevant security event, which is sent to the SIEM.

Based on the operation of the ***AI-Driven Detection Module*** as described above, the ***Adversarial AI Engine*** aims to generate similar but adversarial network flow statistics, in order to mislead the AI model responsible for the detection of the Modbus/TCP attacks. Therefore, the ***Adversarial AI Engine*** can check the robustness of the AI model under different adversarial attacks, including evasion and poisoning attacks. Furthermore, the ***Adversarial AI Engine*** is tasked with discriminating if the network flow statistics are adversarial or not. Based on this discrimination, if the network flow statistics are adversarial a relevant security event is sent to the SIEM and the ***Mitigation Engine*** blocks these statistics.

Subsequently, it is worth mentioning that based on the detection of the Modbus/TCP attacks, the ***Mitigation Engine*** enforces predefined security policies, such as automatically resetting manipulated sensor values, blocking malicious IP addresses, and triggering isolation mechanisms for affected PLCs. The system also supports human-in-the-loop intervention, allowing ICS operators to review mitigation recommendations and adjust response actions through an intuitive security dashboard. Deception techniques further enhance AIAS’s defence strategy by deploying a combination of high-interaction honeypots and digital twins, each serving distinct roles. ***High-interaction honeypots*** emulate realistic ICS services to lure and monitor adversaries, capturing detailed attacker tactics, techniques, and procedures (TTPs). In contrast, ***digital twins*** are used for secure system modelling and behavioural analysis, not direct attacker engagement, thereby avoiding unnecessary exposure of the real operational environment. The architecture carefully isolates these components to ensure security and realism without risking compromise. The captured intelligence is fed into the ***Security Data Fusion System***, where it is correlated with historical attack data and threat intelligence feeds to improve future threat prediction capabilities. The ***explainability layer of AIAS*** ensures that security teams receive clear, actionable insights on detected anomalies, providing justifications for each mitigation action. The system's integration with i***ndustry-standard security frameworks, such as the MITRE ATT&CK for ICS knowledge base***, further strengthens its ability to classify and counter emerging threats.

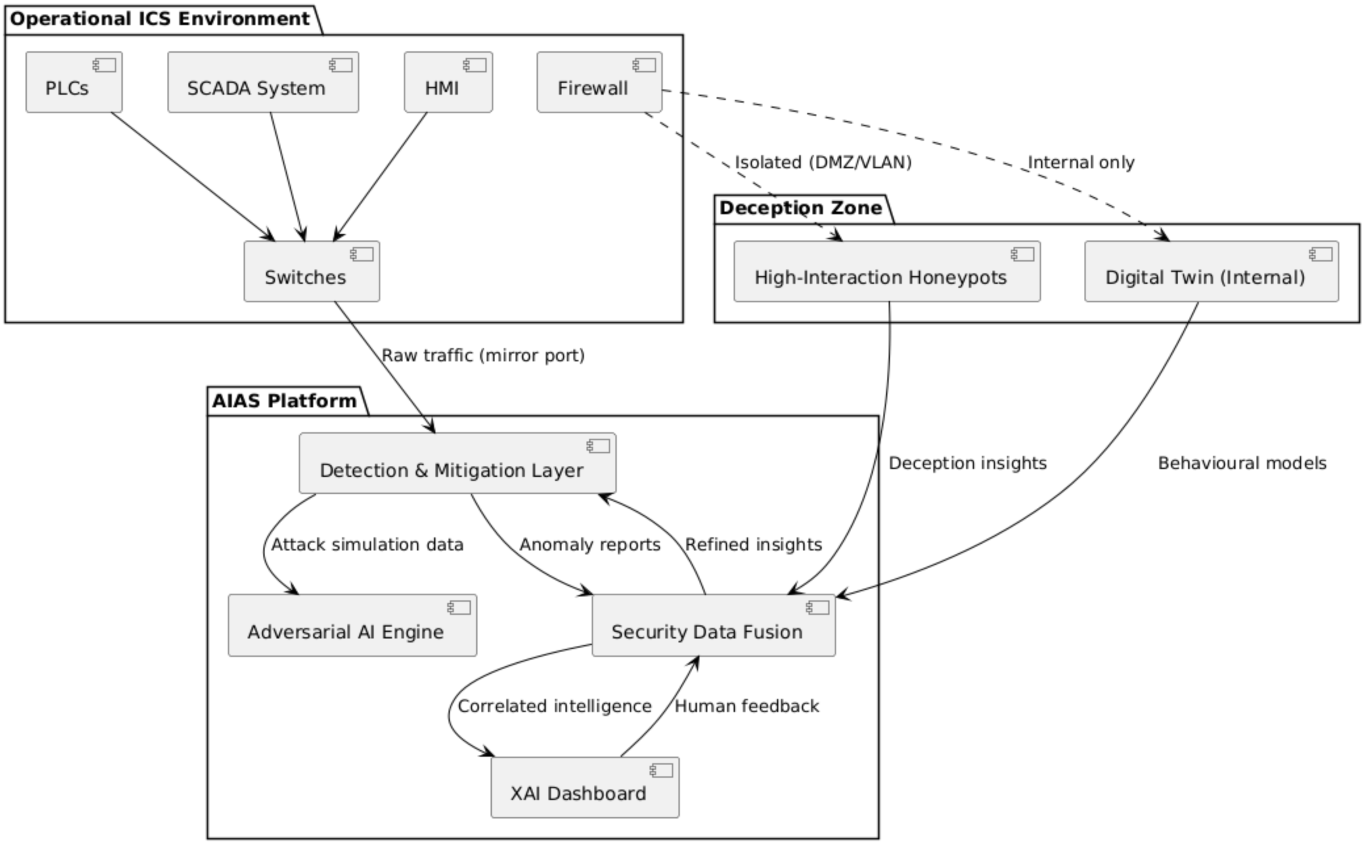


Fig. 1 High Level Industrial Networks Use Case description

2.3. Use Case 3: Detection and Mitigation of Adversarial Attacks on Malware detection software

This use case focuses on enhancing the robustness and reliability of AI-based malware detection systems by identifying and mitigating adversarial attacks. Within this context, the AIAS platform—particularly its Weaponizer component—is deployed in a controlled malware detection environment to evaluate the system’s resilience against sophisticated adversarial techniques [PBF].

The platform integrates three key capabilities. First, it employs an AI-powered malware detection module that uses machine learning algorithms to identify known malware based on learned behavioral and structural patterns. Second, the AIAS Weaponizer functions as an AI-powered malware generator, capable of producing synthetic malware samples. These samples, which may include non-functional or obfuscated code, are specifically crafted to test the limits of the malware detector by attempting to mislead it into classifying malicious content as benign. Finally, an AI-powered anomaly detection module continuously monitors for signs of adversarial manipulation, identifying irregularities in the behavior of the malware detection system that may indicate an ongoing attack.

Together, these components allow for a comprehensive evaluation of both the feasibility and effectiveness of the Weaponizer in a standard anti-malware setting. The integrated approach enables systematic testing, refinement, and fortification of AI-based cybersecurity tools against adversarial threats.

2.4. Use Case 4: SME Providing Digital Services

SMEs that provide digital services (such as web resources, backend applications or commercial software-as-a- service) make use of a series of common technological utilities. Among those, some stand out such as their own mailing service, a web backend and frontend, databases (for storing all sorts of data) and, usually, file sharing systems. Due to the vibrant and challenging ecosystem (of native-digital companies) that SMEs tend to face, as well as the demanding reality of day-to-day operations, it is feasible that their infrastructure and services might be unprotected at the face of adversarial AI attacks (among other types). The enhancement of such a protection via incorporating deception mechanisms seems to be a promising field of action.

The methodology for defining and implementing the use case incorporates three core principles: (i) real digital services (e.g., mail, web server, database…) are not compromised, (ii) the attackers can pervade the “deception system” to a certain extent, enough to allow for advanced analytics, and (iii) the system admin to be aware of the attacks captured by the deception system (honeypots) to the different “targets”. In order to do so, a classic setup scenario will be put in place (hosted by UPV) to become an essential testbed for the AIAS project.

The goal is to demonstrate that, with the innovations included in the Deception Layer of AIAS [LCF], SMEs will have better (open source) tools to more easily detect (and protect against) attacks targeting those services.

3. Methodology for Use Case Definition

3.1. Use Case 1: Application of AIAS in Environmental Monitoring

3.1.1. Use Case 1a: IoT-based DT for Environmental Monitoring

For this scenario, autonomous IoT devices collaborate to ensure optimal environmental conditions. The system incorporates a physical IoT infrastructure with sensors that monitor parameters such as temperature, humidity, air quality index, CH4 level, light level, together with actuators that control lighting and door access. To ensure the system remains operational, all components communicate with one another using a decentralized HTTP framework. The DT is a virtual replica of the physical system that receives data from it, allowing for real-time surveillance, modelling of network attacks, and evaluation under severe circumstances without compromising actual operations.

As for the stakeholders in this UC, the first identified are those users directly affected by the ambiental conditions (e.g., high temperature, gas presence, etc.), as well as those responsible entities, such as human operators or administrators, in charge of managing the settings of the physical environment to ensure the appropriate conditions. In last instance, attackers are identified as the main stakeholders, aiming to compromise the state of the devices and configurations. The use of this DT facilitates the execution of actions from constant monitoring of the environment, to launch malicious scenarios for testing purposes and research.

The methodology addresses functional benefits for AIAS: (i) reliability, guaranteeing that security measures never compromise the environment and its safety with impact on end user; (ii) flawless imitation, understanding the specific workflows and protocols of the application context; and (iii) application of the obtained insights for deception and testing purposes. Thus, this cyber-physical system contributes as one of the possible AIAS Proof of Concept (PoC) for the validation and assessment actions, which will be performed within the project. It could even demonstrate the efficacy of the AI-driven security platform in defending against adversarial AI, employing strategic deception, and provide comprehensive protection in a real-world critical infrastructure environment.

Steps Identifying the use case are presented below:

Step 1: Identify Key Challenges and Fundamental Scenario

To provide a functional PoC based on a set of cyber-physical components, the first stage comprises the study of the functioning of the system, identifying involved IoT devices, types of measurements, communication protocols, dataflows and formats, and, last but not least, the behavioral logic of the physical space where IoT devices coexist. This enables further imitation of the environments where these IoT devices are deployed, creating authentic behavioral patterns for the DT, and ensuring deceptive features that maintain realism while engaging attackers.

Step 2: Use Case Stakeholder-Based Requirements Analysis

As mentioned above, this UC is composed of three relevant stakeholders:

- End users: This classification encompasses all those users that are directly affected by environmental conditions.
- Administrators and human operators: They are responsible for monitoring and managing the physical space to provide the best environmental conditions for end users, aiming towards the proactive protection of the system.
- Adversaries: Malicious entities that attempt to manipulate the ambiental conditions to harm the end users, compromise their comfort and welfare.

Step 3: Deception and Definition: Core Components and their Roles

The IoT-based UC for environmental monitoring covers two principal components:

- Physical IoT system: Comprises environmental sensors (temperature, humidity, air quality, etc.), human input sensors (motion detectors, buttons, etc.), and actuators (lighting, door locks, stretcher motors). All devices communicate via HTTP in a decentralized architecture that enhances resilience by eliminating single points of failure. The motivation for implementing a non-secured communication approach is to leverage a clear understanding and monitoring of the functioning of the system, focusing on more relevant AIAS purposes.
- DT counterpart: A virtual replica of the IoT infrastructure, enabling real-time monitoring, predictive analytics, and controlled simulations. It features data flow from the physical system to prevent compromising real-world devices. The DT allows for cybersecurity attack simulations under extreme conditions to study system behavior and the impact of these threats. Lastly, data is processed and analyzed for further design of mitigation actions.

Step 4: Validation & Testing Mechanisms

The purpose of this DT is to allow AIAS researchers to gain a deeper understanding in distinct cybersecurity areas, such as deception, testing, and autonomous protection. Among the features provided by the proposed DT, it is worth noting the following requirements that must be satisfied to validate the approach effectiveness:

- Anomaly prevention and detection: DT must gather functioning information through different measurement options to design robust IA/ML-based solutions for anomaly detection and subsequent mitigation approaches.
- Flawless imitation: The system must be able to imitate the states and behavior of the devices in the application context, integrating the protocols that conform not only the physical space but also the communication space, settled between the real and digital counterparts.

3.1.2. Use Case 1b: Hospital Monitoring through VP-assisted DT

In more digital medical environments, the ability to adeptly analyze environmental data in real time is essential for maintaining safe and effective treatment. This study introduces a software-based architecture wherein a Virtual Persona (VP) system functions concurrently with a Digital Twin (DT), serving not as a physical duplicate but as a real-time software proxy that delivers organized sensor data over RESTful APIs. The VP module utilizes role-specific logic (e.g., nurse, doctor, IT administrator, patient) to analyze data using anomaly detection, threshold logic, and generative AI for communication purposes. This architecture is entirely virtualized, devoid of reliance on actual sensor equipment, and facilitates intelligent, adaptive decision-making for simulated medical environments, including operating rooms.

Steps identifying the use case are presented below:

Step 1: Identify Key Challenges & Fundamental Scenario

Modern hospitals function as complex ecosystems, where slight changes in environmental factors, including air quality or CO_2 concentrations, can adversely affect patient outcomes and staff efficacy. Conventional systems frequently show limitations in context-awareness, lack adaptive notifications, and encounter manual obstacles in interpretation.

This use case addresses these challenges by employing software to simulate a hospital environment. The Digital Twin generates synthetic but organized environmental data (e.g., CO_2, oxygen levels, temperature), whereas the Virtual Persona functions as a digital agent that can analyze data in real time, guided by specific role-related objectives. This interaction seeks to substitute passive monitoring with cognitive, role-aware alertness.

Step 2: Use case Stakeholder-Based Requirements Analysis

- ***IT Hospital Admin:*** detection and logging of security anomalies, such as off-hour access or unexpected data flow
- ***Medical Doctor:*** alerts for clinically significant anomalies (air contamination or low oxygen)
- ***Nurse:*** operational notifications on the following parameters (temperature, humidity, occupancy)
- ***Patient:*** indirect assurance through system-enforced safety and stability in the environment

The Virtual Persona engine adapts its interpretations and replies to specific stakeholder profiles. For example, a sudden rise in CO_2 may be a clinical warning for a doctor, but a system alert for an IT admin to investigate ventilation system logs.

Step 3: Description & Definition: Core Components and their Roles

Digital Twin Backend: a Flask application serves synthetic environmental values at /sensor_data, including:

- temperature, humidity, air quality
- CO_2 and oxygen levels
- light intensity, noise, occupancy
- door status

This constitutes the real-time, software-based data foundation for all following reasoning.

Virtual Persona Engine: the Virtual Persona system includes:

- *Role selector:* initiated at runtime, defines interpretive behavior
- *Sensor fetcher:* periodically polls DT API
- *Threshold checker:* verifies measurements against defined clinical safety thresholds
- *Anomaly detector:* uses an Isolation Forest to detect subtle data shifts
- *Response generator:* produces natural-language feedback using OpenAI’s GPT-3.5, informed by role and knowledge base
- *Security logger:* detects behavioral anomalies, logs all activities, and simulates potential attack vectors (e.g., IT admin during non-operational hours).

Communication and Logging Layer: all events, warnings and AI responses are published to MQTT topics (hospital/alerts, hospital/response) and stored in CSV/JSON formats. A post-processing utility ensures JSON validity for analytics (fix_json_log).

Step 4: Validation & Testing Mechanisms

To ensure reliability and functional depth, the system incorporates the following validation layers:

- ***Threshold cross-validation:*** manually verifies that safety boundaries comply with clinical standards
- ***Synthetic anomaly injection:*** simulates edge cases (e.g., IT login at 3 AM) to test behavioral responses
- ***Response evaluation:*** Virtual Persona outputs are assessed for semantic correctness and role relevance
- ***Temporal runtime testing:*** monitors memory/log stability, loop behavior, and data fidelity across time
- ***Log integrity checks:*** converts log files to structured arrays for downstream analytics
- ***Anomaly model calibration:*** iterative testing with edge-case vectors refines the Isolation Forest model

The presented VP-assisted DT system demonstrates how software-only components can replicate core decision-making dynamics of physical hospital monitoring infrastructures. By layering role-sensitive cognition, real-time virtual telemetry, and adaptive language generation, the framework enables anticipatory and personalized monitoring within a simulated clinical environment. This abstraction not only facilitates prototyping in the absence of physical hardware but also lays the foundation for scalable, secure, and intelligent hospital informatics systems.

3.1.3. Matching of D2.1 requirements for UC 1a and 1b

The functional and non-functional requirements supporting this UC are detailed in D2.1, and these requirements ensure that AIAS provides deception based on the most modern and cutting-edge defense technologies such as DTs and VPs. Table 3 presents the relevance of each requirement for the two scenarios belonging the UC number 1 (a and b).

Table 3 Use Case 1 mapping with Requirements from D2.1

Requirement ID	Relevance to Use Case

REQ-DECEPTION- DEC-1	Ensures systems completely imitate real devices to deceive attackers effectively.
REQ-DECEPTION- DEC-2	Provides controlled discrepancies in honeypot systems to protect sensitive device infrastructure while maintaining deception effectiveness.
REQ-DECEPTION-SEC-3	Secures communication channels outside the deception system to prevent data leaks and unauthorized access to real systems.
REQ-DECEPTION- DEC-4	Implements fake cooperation strategies to monitor attacker behavior throughout the attack lifecycle in healthcare environments.
REQ-DECEPTION- DEC-5	Utilizes honeytokens to track attacker movements and gather intelligence on attack patterns targeting networks.
REQ-DECEPTION- DEC-6	Maximizes attacker engagement time to divert resources away from real systems and gather comprehensive threat intelligence.
REQ-DECEPTION-DEC-7	Ensures automatic redirection of malicious traffic to deceptive environments, protecting real infrastructure and user safety.
REQ-DECEPTION-DEC-8	Maintains realistic, protected, and consistent data presentation in honeypot systems for long-term deception effectiveness.
REQ-DECEPTION-NFR-9	Guarantees optimal performance, real-time synchronization between DT and real systems, and seamless interoperability with the infrastructure.
REQ-DECEPTION-NFR-10	Ensures system maintainability and reliability under all operational conditions.

3.2. Use Case 2: Application of AIAS in industrial networks

The ***Detection and Mitigation of Adversarial Attacks on Industrial Networks (Modbus)*** use case has been defined through a structured methodology in order to ensure its technical feasibility, alignment with AIAS's objectives, and relevance to industrial cybersecurity needs. The Modbus-based Industrial Control Systems (ICS) are vulnerable due to their lack of built-in security mechanisms, such as authentication and encryption. This use case aims to enhance the resilience of critical infrastructure by detecting, classifying, and mitigating adversarial AI-driven threats in real time. The methodology follows a systematic, multi-step approach that incorporates real-world attack scenarios, AI-powered detection mechanisms, and automated mitigation strategies. The use case is designed to provide a scalable, real-time security framework that integrates with existing SCADA systems, programmable logic controllers (PLCs), and industrial security tools. This is achieved by leveraging AIAS's machine learning-based anomaly detection, deep packet inspection (DPI), and adversarial attack classification. This methodology ensures that the use case:

- Aligns with ***AIAS’s research objectives*** in adversarial AI detection, cyber-physical resilience, and secure data fusion.
- Addresses real-world cybersecurity challenges in industrial networks.
- Is ***technically validated*** through AIAS’s detection, deception, and mitigation modules.
- Supports ***seamless integration*** with existing ICS security infrastructures, including intrusion detection systems (IDS), industrial firewalls, and SIEM platforms.

Steps Identifying the use case are presented below:

Step 1: Identifying Key Security Challenges in Modbus-Based ICS

Industrial control systems utilising Modbus TCP/IP are inherently vulnerable to cyber threats due to the absence of security features such as authentication, encryption, and integrity verification. Common attack vectors include:

- **modbus/dos/arp:** DoS attack with Address Resolution Protocol (ARP) poisoning
- **modbus/dos/galilRIO:** DoS attack against Galil RIO-47100 Programmable Logic Controller (PLC)
- **modbus/dos/writeAllRegister:** DoS attacks trying to write all registers
- **modbus/dos/writeSingleCoils:** DoS attacks trying to write all coils
- **modbus/function/fuzzing:** Fuzzing Modbus functions
- **modbus/function/readCoils**: Reads a specific of Coils
- **modbus/function/readCoilsException:** Fuzzing read coils exception function
- **modbus/function/readDiscreteInput:** Reads the status of specific discrete inputs
- **modbus/function/readDiscreteInputException:** Fuzzing read discrete inputs exception function
- **modbus/function/readExceptionStatus:** Fuzzing read exception status function
- **modbus/function/readHoldingRegister:** Reads a specific amount of holding registers
- **modbus/function/readHoldingRegisterException:** Fuzzing read holding registers exception function
- **modbus/function/readInputRegister:** Reads a specific amount of input registers
- **modbus/function/readInputRegisterException:** Fuzzing read input registers exception function
- **modbus/function/writeSingleCoils:** Writes either 0 or 1 to a given coil
- **modbus/function/writeSingleRegister:** Writes a specific value to a single register
- modbus/scanner/arpWatcher: ARP watcher
- **modbus/scanner/discover:** Identifies if the Modbus service is running in a field device
- **modbus/scanner/getfunc**: Enumerates the function codes supported by a field device
- **modbus/scanner/uid:** Enumerates the function codes supported by a field device
- modbus/sniff/arp: ARP poisoning

The AIAS platform is designed to detect and mitigate these threats in real time, ensuring the availability, integrity, and reliability of industrial network communications.

Step 2: Stakeholder-Driven Requirements Analysis

In order to ensure the practical relevance of this use case, requirements were derived from key stakeholder groups, namely:

- Industrial Network Administrators & ICS Engineers: Require real-time monitoring and automated mitigation of adversarial Modbus traffic.
- Cybersecurity Analysts & Incident Responders: Need AI-driven threat classification, forensic analysis, and real-time security alerts.

- SCADA Operators: Require seamless integration of AIAS into industrial monitoring systems without disrupting operational processes.

By incorporating input from these stakeholders, AIAS ensures that the use case is both technically feasible and operationally effective.

Step 3: Referencing Functional and Non-Functional Requirements from D2.1

The functional and non-functional requirements supporting this use case are outlined in D2.1 – Section 4 (User and Technical Requirements). These requirements ensure that AIAS provides real-time detection, adversarial attack classification, and automated mitigation for Modbus-based industrial networks.

Table 4 Use Case 2 mapping with Requirements from D2.1

Requirement ID	Relevance to Use Case
Req-Detection-SEC-1	Enables continuous monitoring of Modbus traffic for anomaly detection.
Req-Detection-FUNC-3	Provides forensic analysis and attack pattern recognition.
Req-Detection-FUNC-4	Ensure real-time response to detected attacks.
Req-Detection-SEC-5	Uses machine learning-based threat detection for adversarial Modbus activity.
Req-Mitigation-FUNC-1	Supports AI-driven mitigation strategies while allowing human oversight.
Req-Mitigation-FUNC-2	Ensures human-in-the-loop control over security responses.
Req-Mitigation-FUNC-4	Provides adaptive threat response through reinforcement learning.
Req-SecurityDataFusion-FUNC-1	Standardizes attack data for AI-powered analysis.
Req-SecurityDataFusion-FUNC-3	Enforces access control and data integrity in industrial networks.

Step 4: Validation & Testbed Testing

In order to ascertain the effectiveness of AIAS in protecting Modbus-based industrial networks, the use case undergoes technical validation through controlled testing and pilot deployments.

- ***Industrial Testbeds:*** AIAS is deployed in simulated SCADA environments, monitoring Modbus traffic for adversarial patterns.
- ***Red Team vs. Blue Team Simulations:*** Attack scenarios, including modbus/dos/arp, modbus/dos/galilRIO, modbus/dos/writeAllRegister, modbus/dos/writeSingleCoils, modbus/function/fuzzing, modbus/function/readCoils, modbus/function/readCoilsException, modbus/function/readDiscreteInput, modbus/function/readDiscreteInputException,

modbus/function/readExceptionStatus, modbus/function/readHoldingRegister, modbus/function/readHoldingRegisterException, modbus/function/readInputRegister, modbus/function/readInputRegisterException, modbus/function/writeSingleCoils, modbus/function/writeSingleRegister, modbus/scanner/arpWatcher, modbus/scanner/discover, modbus/scanner/getfunc, modbus/scanner/uid, modbus/sniff/arp attacks are executed to test the AIAS's detection and response capabilities.

- ***Integration with Industrial Security Solutions:*** AIAS is validated alongside SIEM platforms, IDS solutions, and industrial firewalls to assess interoperability.

Through this structured validation process, AIAS ensures that its detection and mitigation mechanisms are both effective and deployable in real-world ICS environments.

3.3. Use Case 3: Detection and Mitigation of Adversarial Attacks on Malware detection software

The following steps outline the process of identifying and implementing this use case:

- Scenario Setup: Definition of the testing environment, including selection of appropriate malware detection software and configuration of the computational and operational infrastructure required for the experiments.
- Deployment of Detection and Mitigation Mechanisms: Integration of adversarial detection and response techniques within the selected software framework, enabling the system to identify and counteract potential adversarial inputs.
- Development phase:
  - Malware Creation or Selection: Generation or curation of representative malware samples, covering various threat categories and attack vectors.
  - Data Collection: Acquisition of datasets comprising both benign and malicious inputs, with or without adversarial perturbations, to support training, testing, and validation processes.
  - Cybersecurity Data Analysis: Application of analytical methods to assess vulnerabilities, detect anomalies, and study the impact of adversarial strategies on malware detection models.
- Evaluation: Systematic assessment of detection and mitigation performance through controlled experiments, using predefined metrics to measure effectiveness and robustness.
- KPI Calculation: Quantitative evaluation of system performance based on project-specific KPIs, ensuring alignment with the defined objectives and technical requirements.

Table 5 Use Case 3 mapping with Requirements from D2.1

Requirement ID	Relevance to Use Case
REQ-WEAPONIZER-FUNC-2	Enable collection of data from target systems
REQ-WEAPONIZER-FUNC-3	Enables generation of attacks
REQ-WEAPONIZER-FUN-4	Enables generation of specific attacks against the malware detector

REQ-WEAPONIZER-FUNC-5	Allows the Weaponizer to perform in specific environment
REQ-WEAPONIZER-FUNC-6	Enables the use-case description
REQ-WEAPONIZER-FUNC-8	Evaluates the adversarial attacks against the malware detection

3.4.Use Case 4: SME Providing Digital Services

This use case focuses on the protection of an SME that offers digital services, with an emphasis on evaluating the resilience of AI-based malware detection systems against adversarial threats. The process is structured into four key phases:

- Deploying real services of an SME
- Deploying deception layer (honeypots mimicking those services)
- Identify the main vulnerabilities and actions that a company or institution may suffer as a result of an adversarial AI attack.
- Capture the generated traffic to carry out data analysis..

Requirement ID	**Relevance to Use Case**
REQ-DECEPTION- DEC-1	Honeypots of the associated services are activated upon the attacks on those, generating stats to be analysed by the user.
REQ-DECEPTION- DEC-2	The simulation will stop before revealing sensitive system elements; actually in the use case the real services will be hidden behind the honeypots.
REQ-DECEPTION- DEC-4	Custom module (and also existing ones) will bring attackers the feeling they are penetrating the system. The information gathered will be analysed by the user.
REQ-DECEPTION-DEC-7	Proxies will be used so that the attacks captured by honeypots are not diverted into the real services.
REQ-DECEPTION-NFR-10	New honeypots can be installed (out of the list of those built inside T-POT), and also the capacity of building custom ones.
USR-004	The user will be able to activate/deactivate selected honeypots, therefore retaining level of autonomy of the deception layer in the use case.
Req-Detection-FUNC-8	The user will have at their disposal the logs in Elasticsearch database, which will retain all attacks captured by any honeypots in T-POT.
Req-Mitigation-FUNC-3	The results from the use case will feed the Data Fusion (DF) component (Elasticsearch installed may work as enabler of the DF component)
USR-023	Kibana will act as the visualization interface for the data generated in the use case.

Req-Data_Fusion-FUNC-4	Attacks in the use case may come in the form of stream of messages on in a batch fashion.
Req-Data Fusion- NFUNC-6	An API (relying on Elasticsearch API) will be available for data retrieval in the use case.

4. Detailed analysis of Use Cases

4.1. Use Case 1: Environmental Monitoring

4.1.1. Use Case 1a: IoT-based DT for Environmental Monitoring

Scenario Name	Environmental Monitoring
Objective	The selected scenario represents a critical IoT generic scenario. Specifically, the UC focuses, for example, on an advanced network of autonomous IoT devices, consisting of environmental sensors, human input sensors, and actuators that collaborate to ensure optimal environmental conditions and manage additional factors such as access control, user comfort, and security through the goals of virtualization, simulation and deception strategies. The UC is only applied within the context and goals of WP3, and particularly for: • Detection of threats against the HTTP-based IoT networks. • Detection of anomalies on the functioning of autonomous devices. • Detecting anomalies on user’s behaviour to cover the goals of the WP3 and its virtual persons. • Laboratory tests under simulated conditions, that would not usually be reached within a normal functioning state.
Motivation	The integration of IoT technologies facilitates continuous environmental monitoring and automated control of infrastructures, but concurrently introduces systemic vulnerabilities, particularly concerning cybersecurity, interoperability, and fault tolerance. The motivation for this UC resides in the importance of the industrial paradigm such as for Industry 5.0, where sustainability and safety are pivotal aspects to perform trustworthy scenarios. Based on this assumption, any security risk that may cause a deviation in the natural environmental signifies serious impact on the application contexts and, consequently, on the end users. Precise control over factors such as temperature, humidity, air quality, and presence of toxic substances is essential to minimize risks in critical scenarios. Moreover, the wide range of IoT devices and their use of lightweight communication protocols, (e.g. HTTP), further exacerbates the potential for network-level threats and unintended system behaviour.

Detailed Description	The AIAS platform replicates an existing IoT network for environmental monitoring. The space is equipped with an advanced network of IoT devices, consisting of environmental sensors, human input sensors, and actuators to ensure optimal conditions. • **Environmental Sensors**: Monitor critical parameters such as temperature, humidity, air quality, light intensity, motion detection and toxic gas presence to maintain a safe and controlled environment. • **User Interaction Sensors**: Facilitate user interactions with the system, including fingerprint authentication, motion-based triggers, brightness control interfaces and control systems. • **Actuators**: Responsible for dynamically adjusting environmental conditions by controlling lighting systems, automated door locks, motorized devives, alert mechanisms, and environmental control units. All components operate within a decentralized architecture, communicating via HTTP to ensure efficient data transmission, eliminate single points of failure and enhance system resilience. Therefore, the proposed UC would allow for: • Threat detection in the HTTP-based IoT network by monitoring traffic. Permitting for simulations of network attacks without affecting the real system. • Anomaly detection in autonomous devices by analysing their interactions and behaviour under real-world conditions to identify inconsistencies. • User behaviour analysis by studying behavioural patterns to detect deviations and differentiate between normal and potentially malicious activities. • Laboratory testing under extreme conditions by manipulating sensor data within the DT to replicate scenarios such as high temperatures, toxic gases or poor air quality, ensuring system resilience.
Infrastructure to be used	The UC infrastructure consists of an advanced IoT network designed for environmental monitoring. It includes environmental sensors for temperature, humidity, air quality, light intensity, and toxic gas presence to maintain a safe and controlled environment. User interaction sensors, such as motion detectors and brightness controls, facilitate operation. Actuators dynamically adjust conditions by managing lighting, automated doors, motorized stretchers, and alert systems. All components communicate via HTTP within a decentralized architecture, ensuring efficient data exchange and system coordination. More specifically, this UC will include:

	• **Physical IoT System:** Each space integrates a decentralized network of IoT devices designed to sustain optimal environmental and operational conditions. • **Environmental Sensors** for monitoring temperature, humidity, air quality, light levels, and toxic gases. • **User Interaction Sensors** such as motion detectors, fingerprint readers, brightness controls and device interfaces. • **Actuators**: managing lighting, door locks, air filtration, motorised devices and alarms based on real-time inputs. • **Data Exchange and Protocol Stack**: communication relies on HTTP/REST APIs for configuration and monitoring.
Involved AIAS modules	This UC is specifically designed for determined actions within the AIAS project, and particularly with the Deception Layer (defined in WP3), which includes a Digital Twin.
Scenario flow	1. Environmental conditions are altered or network attacks simulated to assess system resilience and security. **DT** serves as the prototype where these alterations and simulations take place. It would realistically mimic the physical system's response to altered environmental conditions (e.g., changes in temperature, humidity, etc.) or simulated network attacks. 2. System status is continuously monitored to track real-time performance and detect deviations. **DT** continuously provides real-time environmental and status data. This data includes key sensor readings like temperature, humidity, air quality index, and CH-4 level, which are essential for monitoring system performance. 3. System understanding is enhanced through data analysis and insights gained from these experiments. **DT** generates the comprehensive sensor data during these experiments, which serves as material for analysis. 4. Proactive actions are taken. **DT** can trigger proactive actions to protect the system when deviations are alerted.
Stakeholder	• End users: This classification encompasses all those users that are directly affected by environmental conditions. • Administrators and human operators: They are responsible for monitoring and managing the physical space to provide the best environmental conditions for end users, aiming towards the proactive protection and safety of the system. • Adversaries: Malicious entities that attempt to manipulate the ambiental conditions causing harm to the end users and compromising their comfort and welfare.

4.1.2. Use Case 1b: Hospital Monitoring through VP-assisted DT

Scenario Name	Hospital Environmental Monitoring
Objective	To maintain an AI-driven digital twin and virtual persona system in a hospital ward, ensuring continuous surveillance of the environment to guarantee patient safety and comfort. The virtual persona (VP) monitors sensor data from the digital twin in real time and issues alerts or recommendations. Its objective is to promptly identify issues and maintain optimal conditions for healthcare and safety.
Motivation	Due to the inability of hospital personnel to consistently monitor room quality, an automated VP facilitates early issue detection. Environmental conditions should be maintained within safe parameters (for example: a temperature range of 18 to 30 degrees Celsius to ensure comfort). The VP improves monitoring through continuing analysis of these metrics and timely notification of any changes or unusual changes. This increases patient safety and comfort while mitigating security issues, including unauthorized access to IT systems outside of regular hours. This protects both the physical environment and the digital infrastructure.
Detailed Description	Hospital Monitoring using VP-assisted DT combines a digital twin of a hospital room with a more advanced virtual person. The Digital Twin is a Flask-based application that uses a REST API to send real-time data from room sensors, such as temperature, humidity, and air quality. The Virtual Persona component (a Python service) periodically fetches this sensor data and applies multiple validation layers: a threshold checker for any readings outside predefined safe limits, and an ML-based anomaly detector (IsolationForest) to identify abnormal sensor patterns. If an anomaly or threshold breach is found, the VP logs a warning and publishes an alert on the hospital’s MQTT alert topic. The VP then uses a GPT-3.5 based AI model to generate a context-aware response appropriate to its role (doctor, nurse, patient, or IT admin) using the sensor data and a domain knowledge base. The VP integrates sensor data with a domain knowledge base to deliver a contextually relevant answer according to the individual's role (doctor, nurse, patient, or IT administrator). This answer reflects the actions or statements that someone would exhibit in that situation, prioritizing security and comfort. The AI-generated message is recorded and sent via a MQTT response topic, enabling stakeholders to review or act upon it. Log files (CSV and JSON) document all sensor data, identified alerts, and AI responses for subsequent analysis and auditing.
Infrastructure to be used	The use case runs on a local setup combining IoT simulation and AI services. • On localhost:5000, a Flask app acts as the Digital Twin and gives sensor data endpoints. • The Virtual Persona service runs on an edge device or server, connecting to a local MQTT broker for internal messaging. An

	MQTT broker (on port 1883) handles the publish/subscribe channels for alerts and responses. • To get replies from the OpenAI API (GPT-3.5 model), the VP application has to be connected to the internet and have an API key. It also loads a local knowledge base file that gives it context for the domain. • A pre-trained IsolationForest anomaly detection model is embedded in the VP software for ML-driven validation. • Additionally, the environment includes logging facilities (writing CSV and JSON logs to local storage) to persist all events. All components (DT service, VP agent, MQTT broker, network connectivity, and storage) work together to support this scenario within the AIAS project’s testbed.
Involved AIAS modules	This UC is specifically designed for determined actions within the AIAS project, includes a Monitoring analytics, Digital Twin and Virtual Persona.
Scenario flow	1. ***Data Retrieval:*** The Virtual Persona regularly requests the digital twin’s REST API to obtain the most recent room sensor data. 2. ***Validation:*** The VP evaluates each measurement against established parameters, such as temperature and humidity restrictions, and compiles alerts for any values that exceed those limits. The system executes the anomaly detection algorithm on the dataset to identify any sensor patterns that deviate from the norm. 3. ***Alerting:*** If an anomaly or critical threshold breach is detected, the system logs a warning and publishes an alert message (with event ID, role, timestamp, and data) to the hospital’s MQTT alerts topic for immediate attention. 4. ***Security Check (IT Admin role):*** If the VP holds the IT Administrator role, it also simulates monitoring for unauthorized access events, such as flagging any admin activity occurring outside of 1–4 AM as a temporal anomaly, and logs a security warning if triggered. 5. ***AI Response Generation:*** The VP uses the GPT-based AI service to create a written response suitable to its role, leveraging the latest sensor readings and knowledge base context to provide a relevant and supportive reply (for instance, a nurse persona might recommend adjusting the thermostat or checking on a patient). 6. ***Response Delivery & Logging***: The response that has been generated is recorded and shared on the MQTT response topic, along with the event ID and role. All actions, including data, alerts, and responses, are recorded in daily log files for review.

Stakeholder	• ***Doctors:*** Use the VP assistance to monitor patient room conditions and receive notifications regarding any health-critical environmental changes. • ***Nurses:*** Depend on the system to ensure patient comfort (e.g. appropriate temperature, noise levels) and obtain guidance or alerts for immediate adjustments. • ***Patients:*** Experience a more secure and properly managed room environment and may indirectly gain access to immediate assistance or information through the VP to improve their comfort and safety. • ***Hospital IT Admin:*** Manage the hospital’s digital infrastructure, the VP in IT Admin mode assists in identifying anomalies such as unusual access times or device malfunctions, which improves cybersecurity monitoring. All these stakeholders are addressed through corresponding virtual persona roles (doctor, nurse, patient, it_admin) in the AIAS use case.

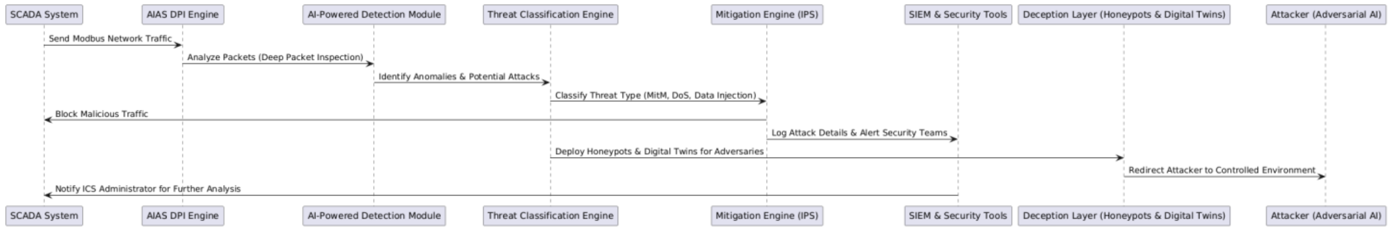
4.2.Use Case 2: Application of AIAS in industrial networks

4.2.1. Scenario 1

Scenario Name	Operational Defense Against Modbus Cyberattacks in Industrial Networks
Objective	The objective of this scenario is to enhance the cybersecurity posture of operational industrial control systems (ICS) that utilize the Modbus protocol, by deploying the AIAS platform to detect and mitigate real-world cyber threats. The focus is on securing live SCADA environments, programmable logic controllers (PLCs), and field devices through real-time monitoring, anomaly detection, deception-based defenses, and automated countermeasures against attacks such as MitM, replay, DoS, and data manipulation. The scenario demonstrates how AIAS protects process integrity, enables explainable alerts for human operators, and actively responds to threats within live industrial networks.
Motivation	Modbus is one of the most widely used communication protocols in industrial control systems (ICS), yet it lacks core security features such as encryption, authentication, and integrity verification. This makes it highly vulnerable to a wide range of cyberattacks, including replay attacks, manipulation of control commands, and denial of service. Traditional signature- and rule-based defenses are increasingly ineffective against modern threats. This scenario is motivated by the urgent need to protect live ICS environments from these protocol-based attacks. AIAS addresses this challenge by applying AI-driven detection, deception-based defense, and real-time mitigation strategies to ensure operational continuity, process integrity, and system resilience.
Detailed Description	The AIAS platform is deployed within an industrial environment that utilises Modbus-based ICS for process automation and control. AIAS

	continuously monitors, analyses and protects Modbus communication channels, detecting anomalies and mitigating threats in real time. The system is composed of several AIAS components that work together to detect, classify and respond to Modbus/TCP cyberattacks. 1. ***Continuous Monitoring of Modbus Traffic:*** AIAS deploys deep packet inspection (DPI) to extract and analyze Modbus transaction identifiers, function codes, and data payloads. The system correlates traffic patterns using unsupervised machine learning models, detecting deviations that indicate potential malicious activity. Normal behavior baselines are established using historical Modbus communication data, allowing AIAS to identify out-of-band commands and unexpected traffic flows. 2. ***AI-Powered Anomaly Detection:*** AIAS employs deep learning-based intrusion detection models, such as Long Short-Term Memory (LSTM) networks and autoencoders, to detect data manipulation, replay attacks, and MitM threats. AIAS incorporates reinforcement learning algorithms to adapt to new attack strategies and minimize false positives. The Adversarial AI Engine detects Modbus/TCP attacks. 3. ***Attack Classification and Threat Intelligence Fusion:*** AIAS leverages graph-based AI models to classify attacks and map cyber kill chain progressions based on MITRE ATT&CK for ICS tactics. The Security Data Fusion module aggregates intelligence from network telemetry, IDS alerts, and external threat feeds to enhance detection accuracy. AIAS provides real-time attack visualization dashboards, enabling security teams to assess risk levels and incident severity. 4. ***Automated Threat Mitigation and Deception-Based Defense:*** Upon detecting an attack, AIAS automatically applies countermeasures, including: a. Blocking malicious Modbus packets using intrusion prevention systems (IPS). b. Isolating compromised PLCs to prevent lateral movement. c. Rolling back unauthorized Modbus commands to maintain process integrity. AIAS deploys high-interaction honeypots and digital twins to deceive attackers and gather intelligence on attack methodologies. Explainable AI (XAI) ensures that human operators receive interpretable alerts and can approve or override automated mitigation responses.
Infrastructure to be used	• Supervisory Control and Data Acquisition (SCADA) Systems: AIAS integrates with SCADA environments, providing real-time monitoring of Modbus traffic and attack visualization.

	• Programmable Logic Controllers (PLCs) & Remote Terminal Units (RTUs): AIAS monitors and protects PLC communications, ensuring command integrity. • Industrial Firewalls & Intrusion Detection Systems (IDS): AIAS interfaces with existing network security solutions to provide multi-layered protection. • Cloud-Based Security Operations Center (SOC) Integration: AIAS logs detected threats and mitigation actions for incident response coordination.
Involved AIAS modules	AI-Driven Detection Module; Adversarial AI Engine; Mitigation Engine; Security Data Fusion System; Deception Layer (Honeypots & Digital Twins, Virtual Personas); Explainable AI (XAI) Dashboard
Scenario flow	1. AIAS monitors live Modbus traffic, extracting function codes and control messages. 2. AIAS’s DPI engine detects an unexpected Modbus command sequence that deviates from normal baselines. 3. The AI-Driven Detection Module classifies the anomaly as a potential Modbus/TCP attack, with a confidence score. 4. AIAS alerts the network administrator, providing explainable insights into the detected threat. 5. The Mitigation Engine blocks the malicious packet and isolates the compromised PLC. 6. Deception mechanisms engage, redirecting the attacker to a simulated industrial environment (digital twin). 7. AIAS logs the attack details and refines its threat models for future detection improvements.
Stakeholder	Industrial Network Administrators; SCADA Engineers; Cybersecurity Teams & SOC Analysts


SCADA System
AIAS DPI Engine
AI-Powered Detection Module
Threat Classification Engine
Mitigation Engine (IPS)
SIEM & Security Tools
Deception Layer (Honeypots & Digital Twins)
Attacker (Adversarial AI)
Send Modbus Network Traffic
Analyze Packets (Deep Packet Inspection)
Identify Anomalies & Potential Attacks
Classify Threat Type (MitM, DoS, Data Injection)
Block Malicious Traffic
Log Attack Details & Alert Security Teams
Deploy Honeypots & Digital Twins for Adversaries
Redirect Attacker to Controlled Environment
Notify ICS Administrator for Further Analysis
SCADA System
AIAS DPI Engine
AI-Powered Detection Module
Threat Classification Engine
Mitigation Engine (IPS)
SIEM & Security Tools
Deception Layer (Honeypots & Digital Twins)
Attacker (Adversarial AI)

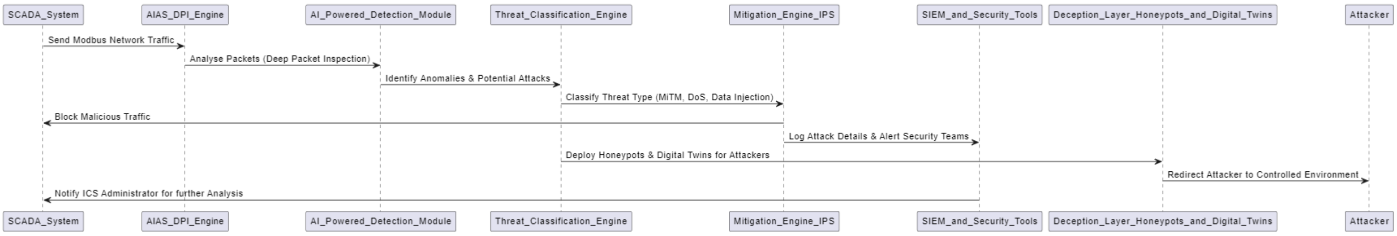

SCADA_System
AIAS_DPI_Engine
AI_Powered_Detection_Module
Threat_Classification_Engine
Mitigation_Engine_IPS
SIEM_and_Security_Tools
Deception_Layer_Honeypots_and_Digital_Twins
Attacker
Send Modbus Network Traffic
Analyse Packets (Deep Packet Inspection)
Identify Anomalies & Potential Attacks
Classify Threat Type (MiTM, DoS, Data Injection)
Block Malicious Traffic
Log Attack Details & Alert Security Teams
Deploy Honeypots & Digital Twins for Attackers
Redirect Attacker to Controlled Environment
Notify ICS Administrator for further Analysis
SCADA_System
AIAS_DPI_Engine
AI_Powered_Detection_Module
Threat_Classification_Engine
Mitigation_Engine_IPS
SIEM_and_Security_Tools
Deception_Layer_Honeypots_and_Digital_Twins
Attacker

Fig. 2 Use Case 2, Scenario 1 workflow for threat detection and Mitigation

4.2.2. Scenario 2

Scenario Name	AI Model Hardening Against Adversarial Attacks in Modbus-Based ICS
Objective	This scenario aims to evaluate and strengthen the robustness of AI models used for detecting Modbus/TCP anomalies by simulating adversarial AI attacks. The AIAS platform is used in a controlled industrial environment to generate, inject, and detect adversarially crafted network flow statistics designed to mislead anomaly detection models. The focus is on discriminating between legitimate and adversarial inputs, blocking misleading data before it corrupts detection logic, and improving the AI model’s resilience against evasion techniques. This scenario ensures the trustworthiness and robustness of AIAS’s detection capabilities under adversarial conditions.
Motivation	As AI models become central to ICS security, they themselves are becoming targets of sophisticated adversarial attacks. Malicious actors can craft adversarial inputs—specially designed data patterns—that are intended to fool anomaly detection models, causing them to misclassify threats as benign. This scenario is motivated by the growing threat of such AI evasion techniques in Modbus-based environments. It highlights the need for proactive robustness testing and adversarial resilience. AIAS addresses this by simulating and detecting adversarial network flow statistics, assessing the model's vulnerability, and applying safeguards to prevent model manipulation, ensuring the reliability of AI-based security decisions.
Detailed Description	The AIAS platform is deployed within an industrial environment that utilises Modbus-based ICS for process automation and control. AIAS continuously monitors, analyses and protects Modbus communication channels, detecting anomalies and mitigating threats in real time. The system is composed of several AIAS components that work together to detect, classify and respond to adversarial AI-driven cyberattacks. 1. ***Continuous Monitoring of Modbus Traffic:*** AI-Driven Detection Module perpetually observes network traffic, extracting Modbus function codes, transaction identifiers, and payload structures for subsequent analysis. 2. ***Network Flow Statistics Generation:*** AIAS’s DPI engine receives this data and generates network flow statistics, leveraging a network traffic flow generator, CICFlowMeter. Adversarial AI Engine receives the generated network flow statistics and also aims to generate similar but adversarial network flow statistics to check the effectiveness and efficiency of the AI model. 3. ***Adversarial network flow statistics detection:*** Adversarial AI Engine discriminates whether the network flow statistics are adversarial or not. This way the AI model is tested in terms of robustness.

	4. ***Automated Adversarial Mitigation:*** Once adversarial statistics are detected, AIAS automatically applies countermeasures, including: a. Generating a security event and sending it to SIEM. b. Mitigation Engine blocks these statistics.
Infrastructure to be used	• Supervisory Control and Data Acquisition (SCADA) Systems: AIAS integrates with SCADA environments, providing real-time monitoring of Modbus traffic and attack visualization. • Programmable Logic Controllers (PLCs) & Remote Terminal Units (RTUs): AIAS monitors and protects PLC communications, ensuring command integrity. • Industrial Firewalls & Intrusion Detection Systems (IDS): AIAS interfaces with existing network security solutions to provide multi-layered protection. • Cloud-Based Security Operations Center (SOC) Integration: AIAS logs detected threats and mitigation actions for incident response coordination.
Involved AIAS modules	AI-Driven Detection Module; Adversarial AI Engine; Mitigation Engine;
Scenario flow	1. AI-Driven Detection Module perpetually observes network traffic, extracting Modbus function codes, transaction identifiers, and payload structures for subsequent analysis. 2. AIAS’s DPI engine receives this data and generates network flow statistics, leveraging CICFlowMeter. 3. Adversarial AI Engine receives the generated network flow statistics and aims to generate similar but adversarial network flow statistics, to mislead the AI model responsible for the detection of the Modbus/TCP attacks. 4. Adversarial AI Engine discriminates if the network flow statistics are adversarial or not, to check the robustness of the AI model. 5. If the network flow statistics are adversarial a security event is generated and sent to the SIEM. 6. Mitigation Engine blocks these statistics.
Stakeholder	Industrial Network Administrators; SCADA Engineers; Cybersecurity Teams & SOC Analysts

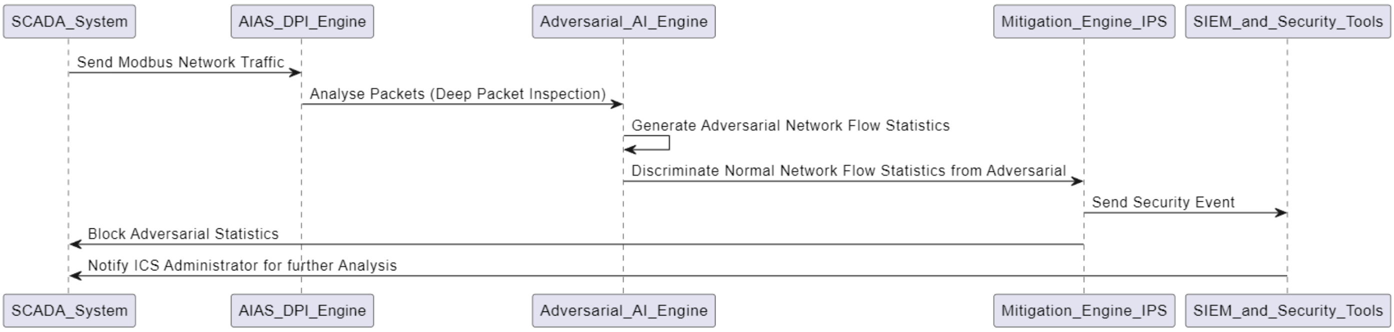

SCADA_System
AIAS_DPI_Engine
Adversarial_AI_Engine
Mitigation_Engine_IPS
SIEM_and_Security_Tools
Send Modbus Network Traffic
Analyse Packets (Deep Packet Inspection)
Generate Adversarial Network Flow Statistics
Discriminate Normal Network Flow Statistics from Adversarial
Send Security Event
Block Adversarial Statistics
Notify ICS Administrator for further Analysis
SCADA_System
AIAS_DPI_Engine
Adversarial_AI_Engine
Mitigation_Engine_IPS
SIEM_and_Security_Tools

Fig. 3 Use Case2 Scenario 2 workflow for Adversarial detection and Mitigation

4.3.Use Case 3: Weaponizer

Scenario Name	Detection and Mitigation of Adversarial Attacks on Malware detection software
Objective	The objective of this use case is to enhance the resilience of malware detection software against adversarial attacks by developing robust detection and mitigation mechanisms. This involves identifying vulnerabilities exploited by adversarial techniques, designing countermeasures to defend against evasion and poisoning attacks, and improving the overall security and reliability of malware classification models. The project aims to integrate advanced AI-driven defense strategies, real-time monitoring, and adaptive response mechanisms to safeguard cybersecurity infrastructures.
Motivation	As cyber threats evolve, adversarial attacks on malware detection software pose a significant risk to cybersecurity. Attackers increasingly leverage adversarial techniques to evade detection, compromise classification models, and undermine the effectiveness of security solutions. Traditional malware detection approaches struggle to keep pace with these sophisticated attacks, leading to potential breaches, data loss, and system compromises. This use case is motivated by the need to strengthen the resilience of malware detection systems against adversarial manipulation. By developing advanced detection and mitigation strategies, the project aims to enhance the trustworthiness of AI-based cybersecurity solutions, ensuring robust protection for critical infrastructures, enterprises, and individuals across the EU.
Detailed Description	The AIAS platform, more specifically the **Weaponizer**, is deployed within a malware detection environment. The whole platform contains the following modules: **AI-Powered malware detection:** An AI/ML model which is able to detect known malware. ***AI-Powered malware generator:*** The AIAS Weaponizer generates synthetic attack scenarios to test and refine AIAS malware detector. This implies generating malware (even a non-working one) such that the classifier mis-predicts it as benign. ***AI-Powered Anomaly Detection:*** An AI/ML model which is able to detect adversarial attacks to the AIAS malware detector. The composition of these three modules allows to correctly evaluate the feasibility and the performance of the Weaponizer in a standard anti-malware environment.
Infrastructure to be used	• PC Workstations with AIAS detector; Server with GPUs for Weaponizer and anomaly detector.

Involved AIAS modules	AI-Driven Detection Module; Adversarial AI Engine; Weaponizer;
Scenario flow	1. AIAS malware detector is configured to detect malware 2. A malware is injected in the PC Workstation, AIAS detector correctly detects it 3. The Weaponizer generates adversarial malware 4. The new malware are injected in the PC Workstation, AIAS detector is not able to detect them 5. AIAS AI-Powered Anomaly Detection will detect these malware as adversarial attacks against the model
Stakeholder	Network Administrators; Cybersecurity Teams & SOC Analysts

4.4. Use Case 4: SME Providing Digital Services

Scenario Name	Deception framework for digital service providing SMEs based on mid and high-interaction honeypots
Objective	Development of a real-world scenario aimed at describing and interpreting the daily/common actions that a **SME offering digital services** performs today. In this way, following the line of work in cybersecurity, we will simulate the most common cyber problems such SME may face today based on this use case. Through our developed technology, we can provide the client with a method/strategy to ensure greater cybersecurity—in other words, to protect the company's sensitive information by enabling early detection of unwanted intrusion activities.
Motivation	As of today, cybersecurity continues to be one of the major challenges and risk factors in the business and institutional world. Recently, new vulnerability factors have emerged, where adversaries have created a new attack surface to exploit the vulnerabilities of AI systems, targeting machine learning (ML) and deep learning (DL) systems to impair their functionality and performance. Adversarial AI is a new threat that could have serious effects in crucial areas such as finance and healthcare, where AI is widely used. The AIAS project aims to conduct in-depth research on adversarial AI to design and develop an innovative AI-based security platform for the protection of AI systems and AI-based operations within organizations. This platform is based on adversarial AI defense methods (e.g., adversarial training, adversarial AI attack detection) and deception mechanisms (e.g., high-interaction honeypots). In this way, through the AIAS project, we have developed and emulated the following use case, which presents a real-life situation where we study and analyze these new types of attacks, in order to find a solution and provide greater security to the companies and entities that require it.
Detailed Description	Following the research line proposed for AIAS, which is based on investigating adversarial AI to design and develop an innovative AI-based security platform for the protection of AI systems and AI-based operations within organizations, relying on adversarial AI defense methods and the implementation of deception mechanisms such as honeypots, we have implemented and developed the following use case, which aligns with the work line of WP3 and WP4. Specifically, for WP3, we will model this deception method based on honeypots to capture information, data, strategies, and steps taken by attackers, whether to attempt data

modification and capture or to access and disrupt services of the targeted company or institution. After this implementation, all relevant information will be processed and transferred to the project's general Data Lake, which will contain abundant information from various sources regarding adversarial AI attacks—thus completing the perfect synchronization and collaborative work between the corresponding parts of WP3 and WP4 in this developed use case.

To cover the part corresponding to the deception mechanism through honeypots (WP3), we have deployed and implemented the T-Pot tool (T-Pot is a platform that hosts more than 20 types of honeypots, allowing for a wide variety of analysis and visualization). Our T-Pot tool is deployed on a VM within the private network of the UPV laboratory, meaning the honeypots are not exposed to external (internet) traffic but are instead in a controlled environment under the supervision of UPV personnel.

To **recreate this real-world scenario**, we must identify the main vulnerabilities and actions that a company or institution may suffer as a result of an adversarial AI attack. Accordingly, we have identified and analyzed the following basic services that are common to any organization and which could be potentially exposed:

- Web page service
- Internal messaging service via email (mail server)
- Remote machine access services (remote access protocols such as SSH)
- Database servers (data access protocols such as SQL)

Once these services have been identified, the idea is to compromise these services via **specific attacks**, and subsequently, with the help of **T-Pot, capture the generated traffic in order to carry out data analysis**.

In relation to the previously mentioned services, our T-Pot tool includes honeypots that meet our needs. For the continuation of our use case, we have selected four key honeypots upon which the implementation will be carried out. These honeypots are:

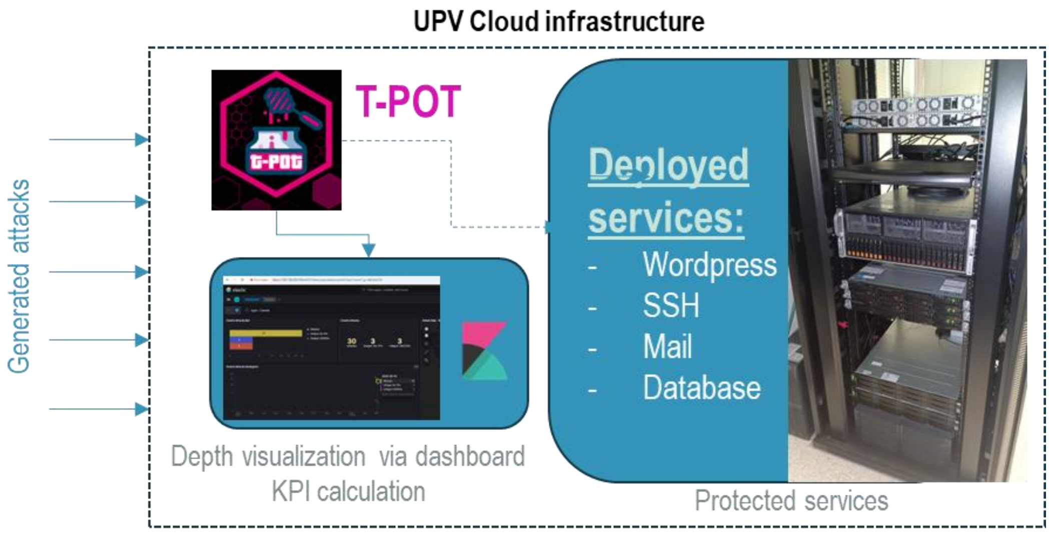
- Cowrie -> For remote access services via the SSH or Telnet protocol
- Mailoney -> Simulates an email server using SMTP
- Wordpot -> Simulates a WordPress server
- Dionaea -> A comprehensive honeypot that simulates various important protocols for our study, such as FTP, HTTP/HTTPS, and MySQL

In this way, the attacks carried out on each service will be:

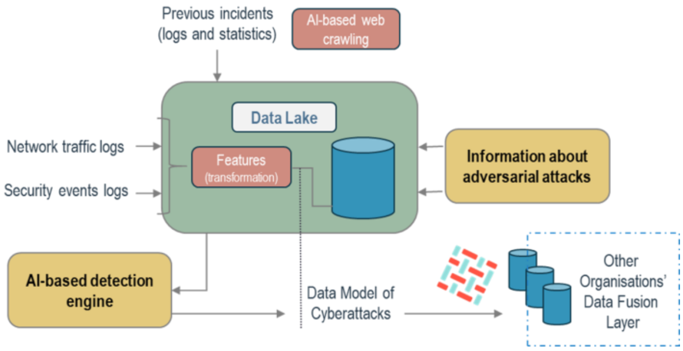
- Web page service (Wordpot): attempts to access the web server presented by Wordpot
- Internal messaging service via email (Mailoney): simulation of phishing attacks using the **swaks** tool
- Remote machine access services: attempted connections via SSH and TELNET
- Dionaea: implementation of Publisher-Subscriber, file transfers between machines, access to database information via MySQL

Following these attacks, our honeypots would then contain data and information to work with and analyze. In addition to the deployment of the honeypots and the data capture from each of them, **T-Pot offers the ability to visualize the data through Kibana**. By using dashboards, important and valuable information can be broken down such as the

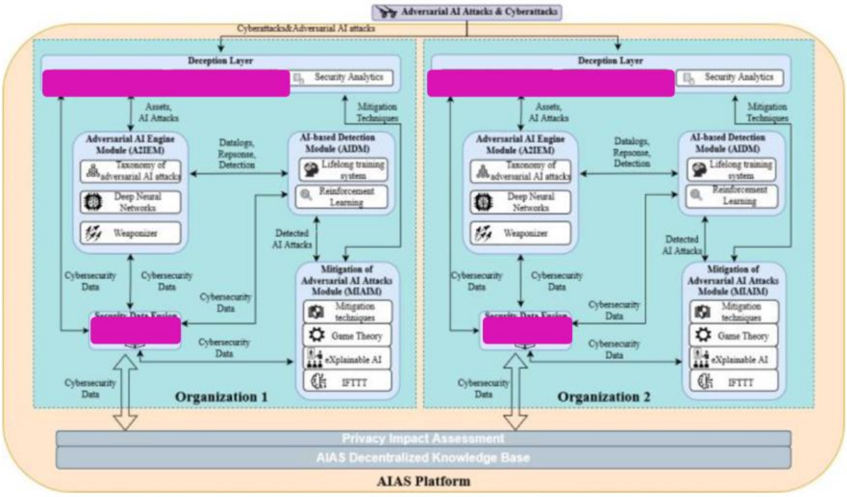
	attacker's IP address, the port used, the region from which the attack originated, plaintext commands used by the attacker, and more. Encompassing all these factors and the deployment strategy to meet the objectives of WP3 under our use case, the following lines of work (tacked in WP3) will play a role in this use case: • The transformation of the honeypots provided by T-Pot into high-interaction honeypots. This involves providing real services and systems with the aim of establishing greater interaction in the honeypot-attacker traffic, thereby enabling more in-depth studies. • The development and implementation of a custom honeypot to be deployed on T-Pot. The goal of this line of research is to explore the possibilities of expansion, structuring, and customization that T-Pot can offer. In this way, we also advance the study of honeypots by addressing the potential implementation of a customized honeypot to provide enhanced security for specific or independent services tailored to each target company or institution—adjusting and personalizing the defense needs for each case, and offering a higher level of security through customization. To conclude, all the information collected and analyzed by the honeypots deployed in our network will be unified and transferred to the AIAS Data Lake, becoming part of the data fusion process under the AIDM (AI-based Detection Module) structure defined in the project, where all information on adversarial AI attacks will be gathered and unified.
Infrastructure to be used	UPV (DRTSL group) counts with an on-premise cloud infrastructure with two owned servers with 2x Intel Xeon Gold 6320R, 512 GB RAM, 10 TB SSD and 4 GPUs of high performance apart from the support of the HPC Center of the University via two clusters with 72 and 128 nodes and a GPU cluster with 100TB of storage. Cloud infrastructure: The cloud infrastructure enables the deployment and validation of virtualized services in a high-performance environment: • Three servers, with large memory and high-performance processors. • An additional server with hardware acceleration capabilities. • Two NAS enclosures • Two UPS equipment. • Two SDN switches at 10 Gbps.
Involved AIAS modules	Our use case contributes an additional module to the overall project architecture. Our module is based on our T-Pot structure—that is, the honeypots that capture suspicious activities, data modeling, and their analysis and monitoring. The structure of the module exploited in this use case responds to the following schema:



And the interaction with the Data Lake (from WP4) will be as follows:


Previous incidents (logs and statistics)
AI-based web crawling
Data Lake
Network traffic logs
Security events logs
Features (transformation)
Information about adversarial attacks
AI-based detection engine
Data Model of Cyberattacks
Other Organisations' Data Fusion Layer

In this way, the global overview of how this use case is integrated/makes use of the overall AIAS architecture is as follows:


Adversarial AI Attacks & Cyberattacks
Cyberattacks&Adversarial AI attacks
Deception Layer
Security Analytics
Assets, AI Attacks
Mitigation Techniques
Adversarial AI Engine Module (A2IEM)
Taxonomy of adversarial AI attacks
Deep Neural Networks
Weaponizer
Datalogs, Reponse, Detection
AI-based Detection Module (AIDM)
Lifelong training system
Reinforcement Learning
Detected AI Attacks
Cybersecurity Data
Security Data Fusion
Mitigation of Adversarial AI Attacks Module (MIAIM)
Mitigation techniques
Game Theory
eXplainable AI
IFTTT
Organization 1
Organization 2
Privacy Impact Assessment
AIAS Decentralized Knowledge Base
AIAS Platform

It can be observed that the use case focuses on Deception Layer (deception mechanisms area) and the Security Data Fusion module. It provides each of these modules with the information gathered by the set of honeypots, thus validating the relevance of the collected

	data by integrating it into the next phases of the project and giving purpose to the remaining modules in terms of their function.
Scenario flow	To carry out the deployment and development of this use case, we can outline the following Flow scenario: 1. **Deployment of the T-Pot tool** in our private and controlled environment, hosted on a VM within the UPV laboratory network. In this way, we will have T-Pot running in a Dockerized setup, along with all the honeypots and services provided by the tool. This setup gives us great flexibility and adaptability in the face of possible changes that may arise in the project, as well as the potential for additional actions to be analyzed and studied within the deployed environment. 2. Since T-Pot is hosted in a private environment, the honeypots will not be receiving suspicious activity, as the machine hosting them is not exposed to the Internet. Therefore, we must **generate specific suspicious activities targeting the four honeypots** we previously identified (which correspond to the basic services typically present in a company or institution) in order to obtain data flow and proceed with the study and analysis of the use case. In this way, we have implemented the following attacks for each honeypot: **Wordpot:** Wordpot is a honeypot specifically designed to emulate vulnerabilities in WordPress, one of the most targeted content management systems (CMS) worldwide. Wordpot operates on ports 80 and 8080. On these ports, it emulates a web server based on Python + Flask/Django, creates fake WordPress URLs, logs all HTTP requests to these routes, and stores logs containing notable attack information such as attacker IPs, payloads, and user agents. Therefore, to generate traffic toward this honeypot, it is enough to attempt access to the web server; any such attempt will be logged and recorded in detail. **Mailoney:** Mailoney is a honeypot specialized in emulating misconfigured or vulnerable email servers, designed to capture attack attempts against email services (such as SMTP, IMAP, or POP3). It simulates a vulnerable mail server (written in Python), deceives attackers by responding like a real server (without sending legitimate emails), and logs all interactions. The chosen method to generate traffic toward Mailoney is by simulating a phishing attack using the *swaks* tool: swaks --to victim@fake.com --from fakebank@secure.com --header "Subject: Urgent - Verify Your Account" --body "Dear user, please verify your account: http://malicious.link" --server x.x.x.x (*Replace x.x.x.x with the IP address of the VM hosting T-Pot.*) **Cowrie:** Cowrie is a high-interaction honeypot designed to emulate SSH and Telnet servers, deceiving attackers and bots scanning for vulnerable systems. It simulates an interactive Linux server, using a custom SSH protocol implementation in Python called Twisted (a library for building asynchronous network protocols in Python). This allows attackers to log in and execute commands, giving them the impression they are interacting with a real server. All actions are logged and stored for later analysis. Thus, by simply making SSH (port 22) or Telnet (port 23) connections to the VM hosting

	T-Pot, we can generate data flows which are recorded and made available for further study. **Dionaea:** Dionaea simulates multiple commonly exploited protocols and services, such as FTP, HTTP, SMB, MSSQL, TFTP, MySQL, and SIP (VoIP). In our case, we generate traffic using the FTP protocol by performing basic file transfer activities through FileZilla. Just like the other honeypots, all activity is logged in a corresponding file, which can be accessed for future analysis. 3. **Analysis of the Generated Data Flow -** As mentioned for each honeypot, after implementing the attacks, each honeypot stores all captured logs in its own file. The goal is to extract this data and carry out a thorough study to obtain relevant information about patterns, steps, structures, commands, etc., used by attackers to attempt to compromise our services. T-Pot follows the following data flow: **Logstash → Elasticsearch → Kibana** • **Logstash:** A data processing pipeline that collects, transforms, and sends information to Elasticsearch. (In other words, Logstash captures the data logged by our honeypots and sends it to Elasticsearch.) • **Elasticsearch:** A distributed, highly scalable NoSQL database that receives data from Logstash and indexes it into structured indices. • **Kibana:** A web interface that allows visualization, exploration, and analysis of data stored in Elasticsearch through interactive dashboards, based on the defined data indices. Likewise, we may consider restructuring or modifying the data to improve interpretation and manipulation of the captured information, as well as to facilitate integration of the data into the AIDM module. 4. **Final Step** - Finally, the use case concludes with the transfer of information captured by the honeypots (within the working environment and under the simulated attack conditions and data flow) to the project's central Data Lake. This repository will contain a large volume of records regarding adversarial AI attacks, gathered from various data flow sources defined throughout the project. The data structure must be standardized across all sources; therefore, a data restructuring process will be carried out with the aim of improving the interpretation and handling of the captured information, as well as ensuring seamless integration into the AIDM module of the project.
Stakeholder	The initial identification of potential stakeholders that would benefit from this use case has generated a first list: • All companies providing digital services, such as private SMEs/cloud providers or owners of digital products (such as ERPs, CRMs, Business Intelligence, websites…). • Public Entities hosting certain digital services, such as schools and universities, with online platforms and services. • Banking institutions, which store highly sensitive and critical customer information, face countless attempted attacks every day.

	• As adversarial AI use spreads, all companies in the ICT sector that rely on AI for their business and projects • Telecommunications companies, which handle large volumes of data and services, some of which are managed and assigned with the help of AI systems. The outcomes of the use case can also benefit individual users, as the use of AI and the vulnerabilities it presents are accessible to everyone. Additionally, the project will be publicly presented, so anyone worldwide could benefit from it once it's published.

5. Conclusions

Deliverable D2.2 has defined and detailed the four pilot use cases that will serve as validation frameworks for the AIAS platform. Each use case has been carefully selected to reflect a distinct application domain, enabling the comprehensive evaluation of AIAS’s core functionalities and its alignment with the project’s research and innovation objectives.

The Hospital Environmental Monitoring use case demonstrates AIAS’s ability to protect critical healthcare environments, where adversarial AI threats can compromise patient safety and system reliability. This scenario highlights the platform’s capability for early detection of AI-driven anomalies in sensor networks and the deployment of explainable and trustworthy mitigation actions.

The Industrial Network Security use case places AIAS in an operational ICS environment, focusing on securing industrial communication protocols such as Modbus/TCP. It showcases the integration of AI-driven intrusion detection, automated threat response, and deception mechanisms in a domain where process integrity and uptime are paramount.

The Weaponizer-Enhanced Malware Detection use case explores the application of AIAS in the dynamic and adversarial landscape of AI-generated malware. It allows the platform to demonstrate its resilience against advanced evasion techniques, adversarial AI samples, and its ability to reinforce cybersecurity tools through continuous adaptation and adversarial stress testing.

The SME-related use case will explore the effectiveness of honeypot and the Security Data Fusion uder a realistic scenario that mimics real services oh an SME.

Together, these use cases provide a diverse and robust foundation for the development, integration, and validation of the AIAS platform. They span different operational domains, threat models, and technical challenges, enabling AIAS to demonstrate its full-stack capabilities across detection, deception, adversarial defense, mitigation, and human-in-the-loop explainability. The defined scenarios ensure that AIAS will be tested under realistic and high-impact conditions, aligning the platform’s technical components with end-user needs and the project’s overarching goal: to safeguard AI systems and use AI securely in critical environments. These use cases will directly inform the development and evaluation activities in the subsequent technical work packages.

References

[D2.1]	Deliverable D2.1 “Requirements and Reference Architecture”, https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e516b61a44&appId=PPGMS
[RAD]	Radoglou-Grammatikis, P., Siniosoglou, I., Liatifis, T., Kourouniadis, A., Rompolos, K., & Sarigiannidis, P. (2020, September). Implementation and detection of modbus cyberattacks.

	In *2020 9th International Conference on Modern Circuits and Systems Technologies (MOCAST)* (pp. 1-4). IEEE.
[AIA]	Petihakis, G., Farao, A., Bountakas, P., Sabazioti, A., Polley, J., & Xenakis, C. (2024, July). AIAS: AI-ASsisted cybersecurity platform to defend against adversarial AI attacks. In Proceedings of the 19th International Conference on Availability, Reliability and Security (pp. 1-7).
[PBF]	Pantelakis, V., Bountakas, P., Farao, A., & Xenakis, C. (2023, August). Adversarial machine learning attacks on multiclass classification of iot network traffic. In Proceedings of the 18th International Conference on Availability, Reliability and Security (pp. 1-8).
[LCF]	Lacalle, I., Cuñat, S., Farao, A., Xenakis, C., Xenakis, D., & Palau, C. E. (2024, October). Deception Mechanisms for Cyber-Security Enhancement in the Internet of Things. In *2024 IEEE 29th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)* (pp. 1-7). IEEE.