Grant Agreement No.: 101093069
Call: HORIZON-CL4-2022-DATA-01
Topic: HORIZON-CL4-2022-DATA-01-03
Type of action: HORIZON-RIA

INCODE

Programming Platform for Intelligent Collaborative Deployments over Heterogeneous Edge-IoT Environments

D2.1 - PLATFORM DESIGN: REQUIREMENTS AND REFERENCE ARCHITECTURE

Work package	WP2
Task	Task 2.1
Due date	30/09/2023
Submission date	03/10/2023
Deliverable lead	FIWARE Foundation, e.V.
Version	1.0
Authors	Chandra Challagonda, Fernando López, Stefano De Panfilis (FIWARE), Georgios P. Katsikas, Dimitrios Manolopoulos, Dimitrios Klonidis (UBITECH), Qi Wang, Jose M. Alcaraz-Calero, Mohammad AlSelek, Angel Gama-Gracia, Julio Diez-Tomillo, Ignacio Martinez-Alpiste (UWS), Maria N. Xekalaki (UMAN), Giorgia A. Marson (NEC), Stavroula-Isidora Giannakandropoulou, Ioannis Avramidis, Denis Avrilionis, Georgios Roumelas (UNIS), George Tziavas, Dimitrios Brodimas (UoP), George Nakas, Panagiotis Radoglou (K3Y), Benny Rochwerger (RedHat), Panagiotis Zikos, George Tsironis (iLINK), Vrettos Moulos (ARHS), Stefanos Venios (SUITE5), Eleftherios Mylonas, Alkiviadis Louridas (IPTO), Lorenzo Lanzara (MADE), Danish Abbas Syed (POLIMI), Benjamin Ertl (AGE)
Reviewers	Georgios P. Katsikas, Dimitrios Manolopoulos (UBI)
Abstract	This deliverable outlines an initial version of INCODE's platform design, requirements, and reference architecture. The INCODE architecture is designed according to (i) a broad set of requirements stemming from the INCODE ecosystem and stakeholders as well as (ii) the definition of four relevant application areas (AAs) and eight use cases (2 per AA). This initial version of the INCODE architecture aims to tame the emerging dynamicity of distributed and heterogeneous private edge infrastructures across the IoT-to-edge-to-core continuum, by leveraging a service and resource orchestration platform over a programmable data plane of IoT, 5G, network fabric, and cloud resources – through the respective sets of

	controllers and drivers - while accounting for increased security and trust.
Keywords	INCODE, platform, requirements, architecture, IoT-edge-core continuum, orchestration, controllers, drivers, security, trust, application area.

Document revision history

Version	Date	Description of change	List of contributors
V0.1	24/05/2023	Table of contents definition	FIWARE, UBI
V0.2	22/07/2023	First round of contributions	ALL
V0.3	16/09/2023	Second round of contributions	ALL
V0.4	22/09/2023	Editing and auxiliary content	FIWARE, UBI
V0.5	27/09/2023	Internal review	UBI, FIWARE
V0.8	29/09/2023	Addressing review comments	FIWARE
V1.0	03/10/2023	Final revisions and submission	UBI, FIWARE, UNIS

Disclaimer

Co-funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the other granting authorities. Neither the European Union nor the granting authorities can be held responsible for them.

Copyright notice

Project co-funded by the European Commission in the Horizon Europe Programme		
Nature of the deliverable:	R	
Dissemination Level		
PU	*Public, fully open, e.g., web*	**X**
SEN	*Sensitive, limited under the conditions of the Grant Agreement*	
Classified R-UE/ EU-R	*EU RESTRICTED under the Commission Decision No2015/ 444*	
Classified C-UE/ EU-C	*EU CONFIDENTIAL under the Commission Decision No2015/ 444*	
Classified S-UE/ EU-S	*EU SECRET under the Commission Decision No2015/ 444*	

* *R: Document, report (excluding the periodic and final reports)*
DEM: Demonstrator, pilot, prototype, plan designs
DEC: Websites, patents filing, press & media actions, videos, etc.
DATA: Data sets, microdata, etc

DMP: Data management plan
ETHICS: Deliverables related to ethics issues.
SECURITY: Deliverables related to security issues
OTHER: Software, technical diagram, algorithms, models, etc.

Co-funded by
the European Union

Executive Summary

This document, titled "D2.1 - Platform Design: Requirements and Reference Architecture," presents an in-depth exploration of the INCODE project, focusing on the platform's design, requirements, and reference architecture. The project aims to create a versatile and comprehensive orchestration platform capable of seamlessly integrating across diverse IoT domains, edges, and core systems.

The document begins with an introduction outlining the project's objectives and its relationship with other deliverables and tasks. It sets the stage for an extensive exploration by introducing the document's structure.

The definitions and prerequisites section elucidates the state of the art in six distinct areas critical to INCODE's objectives. They include smart and scalable orchestration, network programmability, collaborative programming tools, device management, trusted data sharing, and cross-layer innovation. Essential terminology and the applied methodology are also defined in this section.

Moving on to the Initial version of the INCODE Architecture, the document delves into the ecosystem's layers and domains. It highlights the vastly heterogeneous multi-domain Physical Layer (PHY-L), Distributed Resource Management Layer (RM-L), end-to-end Service Management Layer (SM-L), and vertical Security and Trust services Layer (ST-L) that constitute the INCODE platform. The overall architecture is mapped with technical work packages and tasks, providing a comprehensive view of the project's structure.

The logical components in the SM-L and the RM-L are thoroughly detailed. These include components like the User Interface, Role-Based Access Control (RBAC) API, Service Orchestrator, and various controllers and drivers APIs. The document then explores components within the ST-L, focusing on identity Management, Attestation, Data Provenance, and Attack Mitigation.

The requirements section outlines business requirements, functional requirements, and non-functional requirements. Business requirements cater to different stakeholders, including vertical end users, service providers, developers, infrastructure providers, vendors, and system integrators. Functional requirements are delineated for both the SM-L, RM-L, and ST-L, addressing diverse components like schedulers, resource managers, controllers, and drivers across various domains, together with the security requirements to access them.

The document extends its scope to define Application Areas (AA1 to AA4) and their linkage with the established requirements. Application Areas encompass logistics and transport quality value chain, utilities inspection, smart factories with intelligent worker assistance, and communities with public protection and disaster relief (PPDR).

In conclusion, this document serves as a foundational guide for the INCODE project, providing a comprehensive understanding of the platform's design, requirements, and reference architecture. The exploration of logical components, essential requirements, and application areas lays the groundwork for the successful implementation and advancement of the INCODE platform.

Co-funded by
the European Union

Table of Contents

Co-funded by the European Union

Co-funded by
the European Union

Co-funded by the European Union

Co-funded by the European Union

Co-funded by
the European Union

Co-funded by the European Union

Co-funded by
the European Union

List of Figures

Co-funded by the European Union

Co-funded by
the European Union

List of Tables

Co-funded by
the European Union

Co-funded by
the European Union

Co-funded by
the European Union

Co-funded by
the European Union

Co-funded by
the European Union

Co-funded by
the European Union

Co-funded by
the European Union

Co-funded by
the European Union

Co-funded by
the European Union

Co-funded by
the European Union

Co-funded by
the European Union

Co-funded by
the European Union

Abbreviations

3GPP	3rd Generation Partnership Project
4G	Fourth Generation
5G	Fifth Generation
5Gaas	5G as a Service
5GC	Fifth Generation Core
AA	Application Area
ABAC	Attribute Based Access Control
AD	Anomaly Detection
AF	Application Function
AI	Artificial Intelligence
AM	Attack Mitigation
AMF	Access and Mobility Management Function
AMM	Attack Mitigation Module
AMPQ	Advanced Message Queuing Protocol
ANT+	Ultra-low Power Wireless Protocol
API	Application Programming Interface
AT	Attestation
AUSF	Authentication Server Function
AWS	Amazon Web Services
BLE	Bluetooth Low Energy
BRM	Binary Receiving Module
BSR	Business Requirements
CCTV	Closed-circuit television

Co-funded by the European Union

CD	Continuous Delivery
CEP	Complex Event Processing
CI	Continuous Integration
CIM	CONTEXT Information Management
Cloud-C	Cloud Controller
Cloud-D	Cloud Driver
CN	Core Network
CNN	Convolutional Neural Networks
COTS	Commercial Off-The-Shelf
CPE	Customer Premises Equipment
CSR	Context Source Registration
DB	Database
DevOps	Software Development (Dev) and IT Operations (Ops)
DID	Decentralized Identifier
DN	Data Network
DP	Data Provenance
DPU	Data Processing Unit
EBSI	European Blockchain Services Infrastructure
ECG or EKG	Electrocardiography
eIDAS	electronic IDentification, Authentication and trust Services
EMG	Electromyography
eNB	eNodeB
ETSI	European Telecommunications Standards Institute
EUDI	European Digital Identity
ExFa	Experimentation Facility

Co-funded by the European Union

FGM	Flow Generation Module
FNR	Functional Requirements
FPGA	Field-Programmable Gate Array
GDAT	Generic Data Processing Requirements and Regulations
GDPR	General Data Protection Regulation
gNB	gNodeB
GPR	Generic Platform Requirements
GPROG	Generic Programming Requirements
GPS	Global Positioning System
GPU	Graphics Processing Unit
GRML	Generic RM-L Implementation Requirements
GSEC	Generic Security Requirements
GSMA	Global System for Mobile Communications Association
GSML	Generic SM-L Implementation Requirements
GUIR	Generic User Interfacing Requirements and Information sharing
HRV	Heart rate variability
HTTP	Hypertext Transfer Protocol
HV	High-Voltage
IaC	Infrastructure as Code
IAM	Identity and Access Management
ID	Identifier
IDE	Integrated Development Environment
IDP	Internal Developer Platform
IDM	Intrusion Detection Module
IM	Identity Management

Co-funded by
the European Union

IO	Infrastructure Owners
IoT	Internet of Things
IoT-C	IoT Controller
IoT-D	IoT Driver
IP	Internet Protocol
IRI	International Resource Identifiers
ISO	International Organization for Standardization
ISP	Business Requirements for Infrastructure Service Providers
IT	Information Technology
JSON	JavaScript Object Notation
JSON-LD	JavaScript Object Notation – Linked Data
k8s	Kubernetes
KPI	Key Performance Indicators
LoRaWAN	Long Range Wireless Access Network
LWM2M	Open Mobile Alliance Lightweight Machine to Machine
MAC	Medium or Media Access Control
MDC	Malware Detection Components
MDM	Malware Detection Module
MES	Manufacturing Execution Systems
MISP	Malware Information Sharing Platform
ML	Machine Learning
Mob-C	Mobile Core Controller
Mob-D	Mobile Core Driver
MQTT	Message Queuing Telemetry Transport
Naf	Service-based interface exhibited by AF

Co-funded by
the European Union

NEF	Network Exposure Function
NetFabric-C	Network Fabric Controller
NetFabric-D	Network Fabric Driver
NF	Network Fabric
NFR	Non-functional Requirements
NFV	Network Functions Virtualization
NGSI-LD	Context Information Management API - Linked Data
NM	Notification Module
Nnef	Service-based interface exhibited by NEF
Nnrf	Service-based interface exhibited by NRF
Nnssaaf	Service-based interface exhibited by NSSAAF
Nnssf	Service-based interface exhibited by NSSF
Npcf	Service-based interface exhibited by PCF
NR	New Radio
NRF	Network Repository Function
NSSAAF	Network Slice-specific and SNPN Authentication and Authorization Function
NSSF	Network Slice Selection Function
NUDM	Service-based interface exhibited by UDM
O-CU	Open RAN - Centralized Unit
O-CU-C	Open RAN - Centralized Unit - Control plane
O-CU-U	Open RAN - Centralized Unit - User plane
ODRL	Open Digital Rights Language
O-DU	Open RAN - Distribution Unit
OIDC	OpenID Connect

Co-funded by the European Union

OIDC4VP	OpenID Connect for Verifiable Presentations
OPC-UA	Open Platform Communications - Unified Architecture
OpenID	Open Standard and Decentralized Authentication Protocol
OpenPDC	Open PDC service
ORAN or O-RAN	Open RAN
OSM	ETSI Open Source MANO (Management and Orchestration)
P4	Programming Protocol-independent Packet Processors
PAP	Policy Administration Point
PCF	Policy Control Function
PDC	Phasor Data Concentrator
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PHY-L	Physical Layer
PIP	Policy Information Point
PMU	Phasor Measurement Unit
PoC	Proof of Concept
PPDR	Public Protection and Disaster Relief
PRP	Policy Retrieval Point
QoS	Quality of Service
RaaS	Resources as a Service
RAN	Radio Access Network
RAN-C	RAN Controller
RAN-D	RAN Driver
rApps	Non-Real Time RAN Intelligent Controller Applications
RBAC	Role-Based Access Control

Co-funded by the European Union

REST	Representational State Transfer
RFS	Resource Facing Service Specifications
RIC	RAN Intelligent Controller
RL	Reinforcement Learning
RM-L	Resource Management Layer
RO	Resource Orchestrator
RPC	Remote Procedure Call
RRM	Radio Resource Management
RT	Real Time
SAR	
SBA	Service-Based Architecture
SD	Business Requirements for Vertical Service Developers
SDK	Software Development Kit
SDN	Software-Defined Networking
SIOPV2	Self-Issued OpenID Provider v2
SL	Software Lifecycle
SLA	Service-Level Agreement
SmartNIC	Smart Network Interface Controller
SMF	Session Management Function
SM-L	Service Management Layer
SMO	Service Management & Orchestration
SNPN	Standalone Non-Public Network
SO	Service Orchestrator
SP	Service Provider
SQL	Structured Query Language

Co-funded by
the European Union

SRIOV or SR-IOV	Single Root Input/Output Virtualization
SSOP	Self-Issued OpenID Providers
ST-L	Security and Trust vertical Layer
SYS	Business Requirements for System Integrators
TCM	Traffic Capturing Module
TCP	Transmission Control Protocol
TIN	Taxpayer Identification Numbers
TMF	TM Forum
ToF	Time of Flight
TPA	Third-Party Auditor
TRL	Technology Readiness Level
TSO	Transmission System Operator
TSP	Trust Service Providers
UAV	Unmanned Aerial Vehicle
UC	Use Case
UDM	Unified Data Management
UE	User Equipment
UGV	Unmanned Ground Vehicle
UI	User Interface
UL	Ultralight
UPF	User Plane Function
URI	Uniform Resource Identifier
USB	Universal Serial Bus
USER	Business Requirements for Vertical End User
UTR	Universal Trust Registry

Co-funded by
the European Union

UWB	Ultra Wideband or Ultra-wide Band
VC	Verifiable Credentials
VEN	Business Requirements for Hardware and Software Vendors
VGG	Visual Geometric Group
VIM	Virtual Infrastructure Management
VNF	Virtualized Network Functions
VRGM	Visual Representation Generation Module
W3C	World Wide Web Consortium
WP	Work package
XACML	eXtensible Access Control Markup Language
xApps	Near-RT RIC Applications
XML	Extensible Markup Language
XPU	Any Processor Architecture
YAML	YAML Ain't Markup Language
YANG	Yet Another Next Generation data modeling language

Co-funded by
the European Union

1 Introduction

The emergence of edge computing and the shift of processing intelligence towards the end users has lowered the bar for private edge deployments at scale. However, to leverage the true potential of edge intelligence, it takes more than simply increasing the capacity at the edge (through commodity and specialized edge processors) near the various Internet of Things (IoTs). Today's edge computing deployments are highly challenged by the emerging dynamicity of the IoT ecosystem as IoT swarms, edge/core processing elements, and end-user applications form ad-hoc spatiotemporal communication paths with stringent requirements.

INCODE is conceived as an open and trusted cloud-native programming platform poised to tame the emerging dynamicity of distributed and heterogeneous private edge infrastructures. Among the important objectives of INCODE is the constitution and orchestration of a truly programmable data plane of resources that span beyond the typical edge-to-core cloud computing, incorporating a combination of intelligent edge nodes and diverse types of smart IoT devices with processing capabilities.

1.1 Objective

This document comprises the INCODE deliverable 2.1 (D2.1) and aims to provide an initial definition of the INCODE architecture, the design of which will pave the way for achieving our main vision. To do so, this document defines the ecosystem around INCODE, key stakeholders relevant to INCODE, and the requirements that necessitate the INCODE platform. Moreover, this document provides an analysis of the INCODE application areas (AAs), the definition of several relevant use cases (UCs) in each application area, the necessary hardware and software components to realize these UCs, as well as their requirements from the INCODE platform.

1.2 Mapping INCODE's outputs

The purpose of this section is to map INCODE's Grand Agreement commitments, both within the formal deliverable and Task description, against the project's respective outputs and work performed.

Table 1-1: Adherence to INCODE deliverables' & tasks' descriptions.

Project GA Component Title	Project GA Component Outline	Respective Document Chapter(s)	Justification
TASKS			
T2.1 – Requirement, use cases' definition, and business logic	This task focuses on identifying requirements, defining use cases, and establishing business logic for the platform, considering its position in distributed IoT infrastructures and interconnection technologies. The outcomes of this task are functional requirements, business perspective needs, and redefined use cases that inform the platform's design and implementation. The results will be documented in D2.1, scheduled for	Section 4 and Section 5	Section 4 outlines the INCODE requirements categorized into three main groups: business requirements, functional requirements, and non-functional requirements. The business requirements are further subdivided into seven subgroups based on stakeholders. Functional requirements are divided into

Co-funded by the European Union

	delivery in month 9, and will contribute to subsequent tasks such as T2.2, WP3, and WP4		four subgroups, specifying desired behavior and capabilities. Non-functional requirements describe system properties crucial for judging the system's operation. The MoSCoW technique is adopted for requirement prioritization. Section 5 defines the 4 INCODE Application Areas (AA1 – AA5) and their respective Use Cases (UCs) and links their requirements.
T2.2 - Reference platform architecture and testing methodology	In Task 2.2 the INCODE platform architecture is deconstructed into functional elements, each with a detailed design and defined interfaces. The initial architecture is reassembled into a unified reference architecture stack, considering feedback from T2.1 and technical WP3 and WP4 kick-offs. The reference architecture will guide development activities and is set to be provided by M09 in D2.1. In the second phase, feedback from WP3 and WP4 informs updates to the architecture, aiming to create a detailed common reference platform design to guide further development. Simultaneously, a testing methodology is defined based on use case requirements and test-bed capabilities from T5.1, focusing on specific Key Performance Indicators (KPIs) to demonstrate project outcomes and objectives. The outputs include D2.1 in M09 and D2.2 in M18, feeding into T2.3, WP3, WP4, and T5.1.	Section 3	Section 3 introduces the initial version of the INCODE architecture, positioning it within a relevant ecosystem (Section 3.1). It elaborates on the logical components constituting the architecture (Section 3.3) and their preliminary interactions through interfaces (Section 3.4). Additionally, it outlines preliminary workflows depicting essential INCODE platform operations (Section 3.5). Further refinement of the INCODE platform, its interfaces, and workflows will occur in WP2 during the development of D2.2 and D2.3.

1.2.1 Relation with other deliverables and tasks

This document is the basis for the implementation of the Resource Management Layer (RM-L) and Security and Trust Layer (ST-L) to be developed in WP3 as well the Service Management Layer (SM-L) defined in WP4. Additionally, D2.1 includes the initial definition of the requirements and hardware and software components of the application areas to be validated in the context WP5.

Moreover, D2.1 will provide valuable inputs to subsequent deliverables, including:

Co-funded by the European Union

- **D2.2** will be the final version of the architecture shown in D2.1 together with an update of the application areas description and testing methodology.
- **D2.3** will identify key integration points between the components of the INCODE architecture and propose an integration plan.
- **D3.1** will provide the design and initial implementation of the Infrastructure Management Layer based on the design guidelines and requirements described in this document.
- **D4.1** will provide the design and initial implementation of the Application and Business programmability Stratum based on the design guidelines and requirements described in this document.
- **D5.1** will provide details of the deployed testing infrastructure to cover the Application Areas based on the description and requirements contained in this document.

1.3 Document structure

This document is organised as follows:

- **Section 2** defines the state of the art, terminology used in the document, and the methodology used to collect the requirements and architectural features.
- **Section 3** describes the initial version of the INCODE architecture, the layers of this architecture, along with a description of the logical components in each of these layers.
- **Section 4** describes the study and analysis of the collected requirements to facilitate their systematic mapping to the INCODE architecture. These requirements are classified into business requirements, functional requirements, and non-functional requirements.
- **Section 5** defines the INCODE Application Areas, relevant use cases in each AA, and the requirements needed to implement them.
- **Section 6** concludes this document and discusses the next steps in the project development process.

Co-funded by
the European Union

2 Definitions and prerequisites

This section presents the research axes around INCODE (see Section 2.1), introduces important terminology and acronyms that will be consistently used throughout the project's duration (see Section 2.2), and then highlights the methodology adopted for the definition of the INCODE architecture (see Section 2.3).

2.1 State of the art

To pursue the vision for true edge intelligence, INCODE aims to pioneer in six distinct – yet highly correlated areas, which are briefly presented in the following sections. Note that the state-of-the-art analysis shown in Sections 2.1.1 – 2.1.6 is an updated version of the INCODE proposal. Additional text and references are provided to capture the latest developments in these areas as well as relevant pointers to the logical components of the INCODE architecture described in Section 3.

2.1.1 Area #1 - Smart & scalable orchestration across the IoT-Edge-Core-Cloud continuum

State of the art and challenges: Orchestration refers to the many ways of managing application lifecycle, mostly within cloud environments. However, the gradual expansion of modern infrastructures towards the edge has significantly increased the complexity of orchestration. This is because the emerging edge deployments (i) consist of a large number of heterogeneous devices with an equally large number of management tools, while at the same time (ii) the performance of centralized orchestration mechanisms is challenged due to scalability issues. Today's resource orchestration on heterogeneous infrastructures is realized by multiple systems, each managing its own territories (e.g., OSM for NF management [15], OpenSlice for 5G slicing [16], SDN controllers for datacenter networks [20][21], Kubernetes for cloud [22]), often ignoring each other's presence, thus making suboptimal decisions. This becomes even more demanding at the IoT orchestration level due to the plethora of operating systems and solutions.

INCODE ambition: In the face of multiple geo-distributed private IoT-edge nodes, INCODE will address potential scalability pathologies of existing orchestration mechanisms - with respect to the increasing number of locations and devices - by taking orchestration to the next level. Specifically, the INCODE runtime orchestrator spreads orchestration tasks across a designated grid of infrastructure management instances (see the distributed resource orchestration layer in Section 3.1.2), which coordinate their efforts through an end-to-end service orchestrator (see Section 3.1.3) to complete the job more efficiently than if a single orchestration instance had been responsible for the task. INCODE will pay attention to the design of the infrastructure management distributed engine to amortize the complexity of maintaining such a distributed engine by dramatically reducing the time to make orchestration decisions/plans.

2.1.2 Area #2 - Network programmability across the entire IoT-to-edge-to-core continuum

State of the art and challenges: Today's cloud-dominated world mainly deals with ways to program compute and storage microservices within a single node/cluster or across geo-distributed nodes/clusters. In this context, the underlying network is often considered as a blackbox, despite the huge effort to expose its programmability through SDN [8][9][10][11][12] and NFV [13][14][15][16]. With the advent of 5G [18][19], smart IoT devices, and more

Co-funded by the European Union

advanced edge processors, a portion of the network's programmability is exploited (i.e., traditionally hardware-based network functions turned into software-based microservices [14]), however open problems still exist. First, a portion of the available IoT technologies still consider fixed radio access networks (RAN)s and legacy gateways, with little to no reconfigurability on RAN-specific parameters or traffic engineering. Secondly, the emergence of edge processors with various layers of programmability (i.e., ARM cores, GPUs, SmartNICs, P4) is partially only exploited by today's systems as most of the efforts deliberately focus on application-layer concepts (i.e., GPU application acceleration). Programming the underlying data path - used by applications - could bring enormous performance, security, and elasticity benefits, if SmartNIC adapters and programmable switches could be jointly exploited by applications.

INCODE ambition: INCODE is the first attempt to consider an entirely programmable resource stratum consisting of IoT nodes, connected to an open fully programmable RAN to mobile core to datacenter network fabric, towards powerful edge processors and commercial off-the-shelf (COTS) edge/core servers. This vision is made possible through the various controllers and drivers described in Section 3.3.2.2 and unlocks tremendous opportunities for future applications to exploit the underlying programmability for achieving unprecedented performance and augmented user experience, while at the same time allows the INCODE orchestration to dynamically create ad-hoc spatiotemporal communication paths formed by communities of smart agents/peers/clouds for solving time-critical IoT tasks.

2.1.3 Area #3 – Collaborative programming tools for efficient, decentralised intelligence applications on swarms

State of the art and challenges: Programming, deploying and maintaining applications in environments that include dynamic groups of nodes in the IoT device-edge domain is a complex task that can alienate adopters from properly updating and providing new features to their application. Industry practices such as CI and CD are considered an integral part of software development; still, this practice has not gained traction in the smart-edge field, with the development and deployment of new IoT application features for swarms of heterogeneous devices being a tidy and inefficient process. Existing platforms for the management of cloud-backed Edge IoT devices (i.e., AWS IoT [23], Azure IoT [24]) offer highly integrated toolchains that try to overcome this issue [25]. However, these tools are dependent on the available cloud services (i.e., storage, analytics, etc.) offered by each provider and resulting in vendor lock-in and limited options for IoT/Edge application developers. Finally, more recent trends, such as the Internal Developer Platform (IDP), are trying to resolve one of the issues with the CI/CD and DevOps approaches; the assumption that developers also manage, at least partly, the deployment of their applications and are familiar with an ever-changing set of tools for infrastructure management; this assumption is not usually true outside the IT world and can also quickly create bottlenecks in the CD process - especially when dealing with highly heterogeneous IoT devices and multitudes of vendor-specific services and tools.

INCODE ambition: The INCODE platform aims to create a set of configurable tools for the IoT-Edge domain to reduce the complexity of programming and maintenance, while also improving the efficiency of decentralised intelligence apps. The goal is to create advanced tools which can allow application development and deployment for large swarms of devices at the edge by (i) creating an open framework that can support multiple AI/ML frameworks, cloud providers and edge devices, thereby eliminating vendor lock-in and enhancing interoperability, (ii) building a multi-role IDP (see Section 3.3.1.4) to speed up the application development and deployment process on large swarms of devices, (by enabling developer self-service and eliminating the bottlenecks caused by current approaches) and (iii) allowing easier feature development by leveraging release distribution (with A/B or canary releases) and remote testing logging and debugging functionalities (see Section 3.3.1.4.3). Finally, efficiency will be improved by (iv) automated hardware acceleration for efficient use of the resources by Java programs (see Section 3.3.1.4.4), and (v) an innovative distributed reasoning engine based on

Co-funded by the European Union

the synergy between a service-level policy engine (see Section 3.3.1.3.3) and a resource-level policy engine (see Section 3.3.2.1.3). Developers are assisted through the use of (iv) and (v), both in terms of design time and runtime tools, with a dedicated integrated development environment (IDE) allowing for source-level decoration, and IDP support for defining inference rules needed to support dynamic grouping of nodes across the device-edge-cloud, which not only reduces the complexity of programming and maintenance but acts as an enabler for ad-hoc collaboration during rules evaluation.

2.1.4 Area #4 – Device management and unified hardware abstractions

State of the art and challenges: Existing research project approaches [26][27][28][29][30][31] in the area of IoT avoid managing the vast heterogeneity and the large number of IoT devices, which constantly grow over the last years. A safe path is typically selected which effectively "hides" the many IoTs behind statically configured gateways, which offer limited to no access to device-level information (e.g., hardware specifications, data models) as well as no visibility on device-level traffic, thus limited security whatsoever. As a result, managing IoTs involves simply accessing their gateways and sending traffic to a data analytics platform. To make things worse, each IoT standards organization and IoT platform vendor have created their own version of IoT data models, following a bespoke metamodel and representation language, thus further amplifying IoT management problems. Finally, IoT devices are only a portion of the emerging edge hardware. Considering the growing number of edge processors assembled out of FPGAs, ARM processors, and XPUs (i.e., data, graphics, vision, tensor, and neural processing units), it is essential to immediately address hardware abstraction.

INCODE ambition: INCODE takes a radical approach in solving the device management conundrum by introducing an infrastructure management layer (see Section 3.1.2 and Section 3.3.2) with a diverse set of drivers covering all sorts of IoT devices, network elements, and (edge) compute processors. Specifically, in the area of IoT, rather than introducing "locked" gateways and custom IoT data models, INCODE opens up the IoT RAN fostering the leading O-RAN architecture through 5G and beyond communications. This allows INCODE IoT drivers to programmatically access low level IoT signalling, acquire full control on both device and flow levels, but more importantly introduce open standards (e.g., OpenConfig [32] and YANG [33]) for encoding and managing IoT data models (see Section 3.3.2.2.1). In the area of edge processing, state-of the art smart NICs and edge processing nodes with embedded GPU capabilities are considered and combined with homogenized data models and interfaces for state-of-the-art DPUs under INCODE. An essential aspect involves developing a generic driver framework (in the form of an SDK) to enable future driver developers to adhere to compatible rules and seamlessly integrate any driver type into INCODE.

2.1.5 Area #5 – Trusted data sharing, secure processing, and device-level security

State of the art and challenges: Existing edge solutions are provided by various vendors atop heterogeneous infrastructures. This creates a scattered landscape and a diversification of the security mechanisms each vendor develops, with difficult, even impossible ways to coordinate all these security systems. Orchestration and resource management requires coordination between the various edges, so the security and privacy aspects are mandatory to be taken into account [34]. It is evident that security and privacy of such systems needs to be handled by a centralized authority that coordinates the authorization, access control and data sharing across the different edges [35].

INCODE ambition: INCODE addresses these challenges in two research directions: An IoT Device and Drivers Authentication (see the Identity Management component in Section 3.3.3.1) and Attestation (see the Attestation component in Section 3.3.3.2) service is

Co-funded by the European Union

developed, in which first the IoT device registers to INCODE obtaining a blockchain id through a distributed leger mechanisms after authentication and then verified through hardware attestation and a set of predefined attestable attributes (ABAC concept). The solution will examine the use of dynamically configured attestation attributes based on the type of devices and their origin, as well as novel hybrid (hardware and software) attestation solutions to reduce cost according to examined performance effects. The same authentication and attestation process occurs for device drivers ported to INCODE to ensure the integrity and trust of the software. Data and application provenance mechanisms are deployed through a permissioned blockchain for logging/auditing models, datasets and software artefacts (see the Data Provenance component in Section 3.3.3.3). The goal is to use these mechanisms to guarantee the traceability of those data and software artefacts when residing inside the INCODE platform. Complementary, the usage of smart contracts that audit the consistency of data reported by distributed data sources ensure trusted and reliable software and data inside the platform. Finally, INCODE also investigates efficient in-network attack mitigation mechanisms (see the Attack Mitigation component in Section 3.3.3.4) to be used as a service by the INCODE orchestration platform for preventing malicious activities at various layers of the ecosystem.

2.1.6 Area #6 - Cross-layer innovation through an open ecosystem

State of the art and challenges: Existing edge deployments typically rely on public or telco owned infrastructures, thus fail to provide their stakeholders full control over the underlying platform and especially the hardware. Specifically, in such edge deployments stakeholders may have limited choices when it comes to (i) what types of policies can be applied to the underlying infrastructure, (ii) how much hardware intervention is offered in terms of reconfiguration, and (iii) visibility of the underlying data path due to the existence of proprietary hardware or lack of permissions to deeply inspect its interfaces. Apart from these problems, such an approach also limits the number of stakeholders that can benefit from edge deployments, as hardware (i.e., IoT, whitebox, edge processors) incubators are left behind. To support the advent of Industry 4.0 as well as parallel advancements in the energy and waste management sectors, hardware innovation is key.

INCODE ambition: INCODE creates private infrastructure islands out of whiteboxes with an open framework to manage and innovate on top of them. This way, INCODE allows stakeholders to freely introduce applications and policies not bound by any low-level infrastructure restrictions, while programmable hardware vendors are first class citizens in this versatile ecosystem. The ultimate INCODE vision is to incrementally establish ad-hoc synergies among relevant European (initially) stakeholders that may fully regulate the way business actors and clients will be (i) accessing, (ii) exploiting, and (iii) benefited from the INCODE platform in tandem with the INCODE infrastructure.

2.2 Terminology

For the better understanding of INCODE and for the shake of consistency, it is important to highlight some terminology that is key around the INCODE concept and ecosystem. Although this terminology was initially defined in the INCODE proposal, during the INCODE runtime, the consortium identified the need to refine part of this terminology. This section clearly introduces this terminology and what updates have been done since the INCODE proposal, also summarized in Table 2-1. This terminology will be consistently used throughout the entire project.

Co-funded by
the European Union

Table 2-1: Updated INCODE terminology and acronyms.

Old Term and Acronym		Updated Term and Acronym	
Name	**Relation to WP**	**Name**	**Relation to WP**
Infrastructure Management Layer (IML)	WP3	Resource Management Layer (RM-L) – See Section 3.1.2	WP3
Application and Business Programmability Stratum (ABPS)	WP4	Service Management Layer (SM-L) – See Section 3.1.3	WP4
Runtime Orchestration (RO)	WP4	Service Orchestrator (SO) – See Section 3.3.1.3	Part of the orchestrator is in the SM-L (WP4)
		Resource Orchestrator (RO) – See Section 3.3.2.1	Part of the orchestrator is in the RM-L (WP3)
Telemetry	WP4	Service-level Telemetry – See Section 3.3.1.3.2	Part of the telemetry relates to SM-L (WP4)
		Resource-level Telemetry – See Section 3.3.2.1.5	Part of the telemetry relates to RM-L WP3
Resource Manager	WP3	Resource Orchestrator (RO) with Resource Manager as an internal sub-component – See Section 3.3.2.1	WP3

Co-funded by the European Union

Security & Trust Initially, this was a component of IML underneath the device drivers	WP3	Security and Trust Layer (ST-L) – See Section 3.1.4 This has now become an entire vertical layer that exposes 4 security & trust services (see Section 3.3.3) to both SM-L (WP4) and RM-L (WP3)	WP3 This layer still belongs to WP3 but serves also WP4

2.3 Methodology

To define the INCODE architecture in a way that benefits all the actors of the INCODE ecosystem, the methodology shown in Figure 2-1 is pursued. The process begins from - but also end at - the INCODE stakeholders, who are the main drivers of the workflow. First, the INCODE stakeholders shape the definition of the application areas that are relevant for INCODE by defining competent use cases in each application area. These use cases shall be able to stress key aspects of the INCODE to make it unique and useful to the global community. After the definition of the use cases, functional, non-functional, and business requirements are identified per use case, out of which the INCODE consortium will distil features of the architecture. During this process, feedback is provided to the technical WPs of INCODE with regards to the main technical components of the architecture, their interfaces, etc. When the features are extracted, an iterative CI/CD process begins for the design and implementation of these features into the INCODE platform. When enough features are implemented for a certain use case, a validation process begins to evaluate whether the use case functions as expected. This creates additional feedback towards (i) the design of the architecture and (ii) the use case itself. Upon a successful validation, the use case goes to production and its exploitation begins by creating value towards the stakeholders. This whole process repeats for all the use cases.

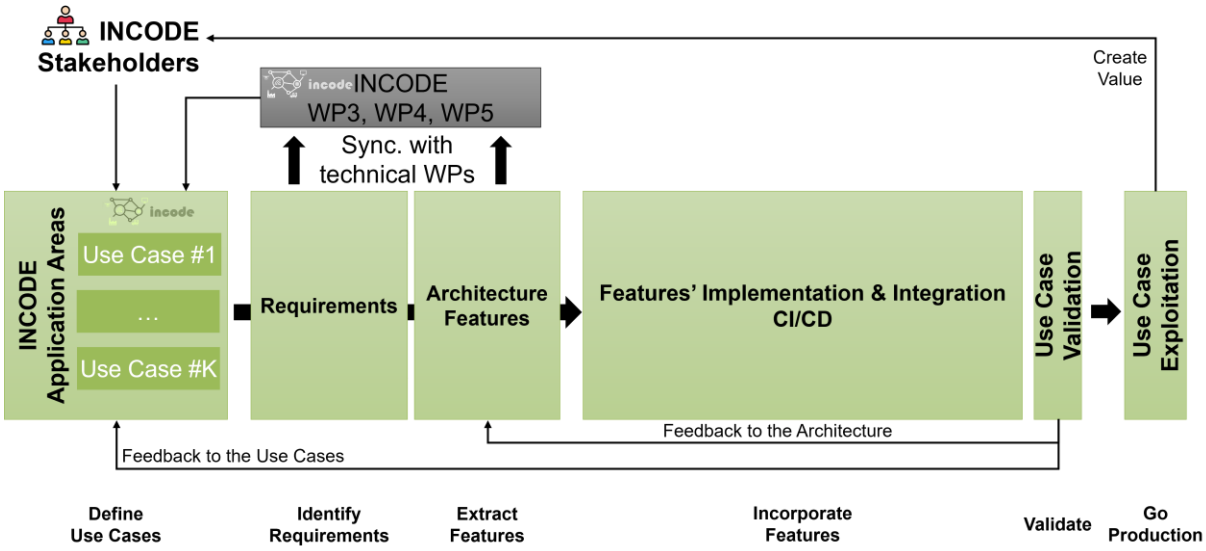

Figure 2-1: Methodology to define the INCODE architecture.

Co-funded by the European Union

In the context of this deliverable, the relevant part in Figure 2-1 concerns the iteration between use cases definition and validation as this is the set of steps that contributes to the initial definition of the INCODE architecture, but also to the evolution of the architecture throughout the project.

Important remark: Note that the identification of requirements always (chronologically) precedes the design of the architecture, despite the fact that in this deliverable they are presented in a reverse order (i.e., Section 3 about the architecture definition is placed earlier than Section 4 about the Requirements). We opted for this layout just to facilitate the readability of the document.

Co-funded by
the European Union

3 Initial version of the INCODE Architecture

This section introduces the first version of the INCODE architecture. Section 3.1 positions INCODE in a relevant ecosystem, managed by the INCODE architecture presented in Section 3.2. Section 3.3 details the logical components that comprise the INCODE architecture, while the preliminary interactions between these components are described the same section, by means of interfaces. The INCODE platform, its internal and external interfaces as well as the workflows will be refined at a later stage in WP2, in the context of D2.2 and D2.3.

3.1 Ecosystem: Layers and Domains

INCODE tames the complexity of a vastly heterogeneous infrastructure that spans across several geo-distributed domains. Because these domains may be owned and managed by different administrative entities, INCODE's orchestration stack introduces multiple layers that maintain a fine-grained separation of concerns with regards to (i) what is considered as infrastructure, (ii) who manages the infrastructure per domain, (iii) how do infrastructure resources get exposed to the overlay services, (iv) how services are managed within a domain as well as across domains, and (v) how vertical end users and other relevant stakeholders consume services offered by the INCODE platform.

The horizontal domains managed by INCODE are detailed in Section 3.1.1. Section 3.1.2 describes the distributed resource management layer of the INCODE architecture, while Section 3.1.3 introduces INCODE's end-to-end service management layer. Finally, Section 3.1.4 introduces a vertical layer that offers security and trust services to the resource and service management layers of the INCODE platform.

3.1.1 INCODE PHY-L: Vastly heterogeneous multi-domain physical layer

INCODE infrastructures comprise of geo-distributed and heterogeneous physical appliances that span across multiple interconnected domains as shown in Figure 3-1. We call this infrastructure "INCODE's physical layer", which is abbreviated as PHY-L.

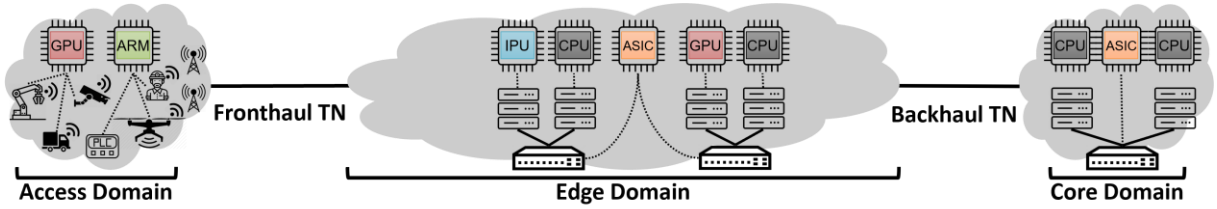

Figure 3-1: The heterogeneous multi-domain physical layer (PHY-L) of the INCODE architecture.

Access Domain: On the left-hand side in Figure 3-1. Vertical domains (e.g., logistics, critical infrastructures, smart factories, public protection and disaster recovery, smart farming, etc.) utilize IoT devices customized to meet the requirements of their specific stakeholders. The goal is to modernize their business operations, automate error-prone tasks, and use technology to assist their personnel. INCODE needs to associate and register these IoT devices with the platform as well as provide means to store their data and actuate upon them. In more extreme cases, these IoT devices do not only merely transfer data, but also possess local processing and communication capabilities that allow them to formulate IoT swarms. In this case, INCODE needs to exploit the local processing and communication capabilities of these devices, by offloading critical application logic there.

Edge Domain: The middle part in Figure 3-1 illustrates INCODE's edge domain(s). This domain interconnects with a vertical access domain through a fronthaul transport network offering: (i) key network services, such as 5G radio access network (RAN) network functions

Co-funded by the European Union

(NFs) for establishing a working RAN (i.e., a 5G gNodeB) to accommodate the traffic of the vertical domain and (ii) compute and storage resources - either on commodity or specialized edge processors - for hosting part of the vertical application that requires proximity with the IoT devices and the end-users.

Core domain: On the right-hand side in Figure 3-1, a backhaul transport network interconnects one or mode edge domains with a core domain. This domain offers (i) core network services by means of a 5G core (5GC) where the various INCODE RANs register themselves and get broader access to external networks and (ii) ample compute and storage resources that can be offered to demanding (in terms of compute and storage requirements) applications and/or application components which can tolerate higher latency.

End-to-end perspective: This segregation of domains is fully compliant with 3rd Generation Partnership Project (3GPP)'s definition of Radio and Core Network domains [1] [2] as well as IETF's Transport Networking [3]; the composition of such "Radio-Transport-Core" domains forms end-to-end service domains that INCODE is designed to manage [4].

3.1.2 INCODE RM-L: Distributed resource management across domains

Infrastructure owners neither disclose the internals of the underlying system, nor do they provide external users direct access to the physical or virtual devices of the infrastructure. To meet these hard requirements, infrastructure owners typically implement infrastructure services which are bound to the underlying physical or virtual resources, while exposing northbound APIs that allow external users/stakeholders to consume them without dealing with the complexity of the underlying hardware.

INCODE's resource management layer (RM-L) is built around this principle. To manage the resources of the physical layer, the INCODE architecture employs a distributed RM-L as shown in Figure 3-2. In short, this layer employs a resource orchestrator that exposes infrastructure resource and service APIs towards the upper layers of INCODE and controls the underlying infrastructure (PHY-L) through a set of controllers and drivers as shown in Figure 3-2.

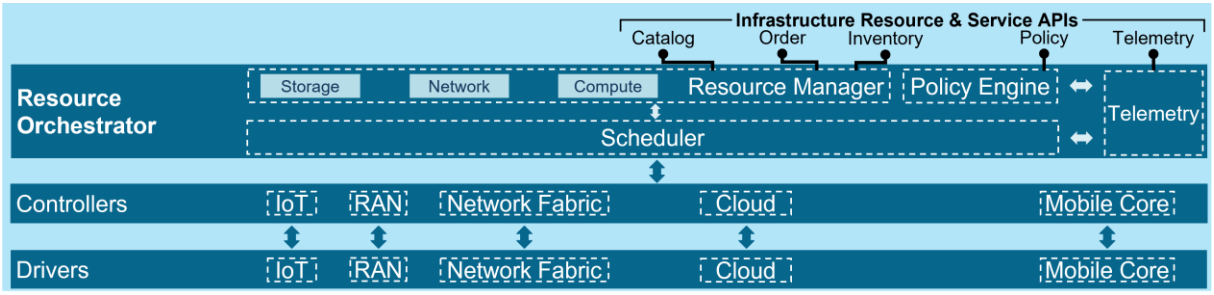

Figure 3-2: The distributed cross-domain resource management layer (RM-L) of the INCODE architecture.

Resource Orchestrator: The Resource Orchestrator (RO) oversees the entire RM-L as it manages all compute, storage, and network resources of the underlying infrastructure. On the one hand, the RO consumes the northbound interfaces of the underlying controllers (through the drivers associated to these controllers) to (i) observe the underlying infrastructure elements, their state, and resources, (ii) add/remove infrastructure elements, and (iii) allocate/release resources on/from infrastructure elements. To ensure that the state of the infrastructure is steady, telemetry information is stored for every part of the underlying infrastructure (i.e., IoT, RAN, network fabric, cloud, and mobile core). This information is used by the RO to make relevant decisions for e.g., scaling the infrastructure (thus the offered resources) up or down, dealing with failures and/or temporary unavailability of elements/resources, etc. On the other hand, the RO exposes technology-agnostic northbound APIs towards INCODE's service management layer, thus allowing the service orchestrator to consume infrastructure services that are linked to specific infrastructure resources. A policy engine component allows the RO

Co-funded by the European Union

to conditionally (based on the state obtained by the telemetry) apply certain actions to the underlying infrastructure during runtime. Further details about INCODE's RO are provided in Section 3.3.2.1.1.

Controllers: The RM-L controllers' interface with (i) the RM-L drivers towards the infrastructure side and (ii) the RO towards the upper layers of the INCODE platform. Five controller types are defined, each interfacing with a respective set of drivers as follows: (i) an IoT controller (IoT-C) manages the behaviour of one or more IoT drivers (depending on the number of IoT devices being used and the low-power protocol implementation used by these devices), (ii) a RAN controller (RAN-C) manages important RAN parameters and the behaviour of RAN-based network functions, either those being implemented into an integrated 5G NR solution (e.g., a commercial gNB) or as parts of a disaggregated open RAN solution, (iii) a network fabric controller (NetFabric-C) manages programmable network elements, such as switches, routers, and whiteboxes within a core or edge datacenter using state of the art software-defined networking (SDN) protocols, such as P4, (iv) a cloud controller (Cloud-C) manages compute, storage , and network resources within a server, a cluster of servers, or even across geo-distributed clusters, and finally (v) a mobile core controller (MobCore-C) manages the network functions of a mobile core network (CN)

Drivers: The RM-L drivers interface with key elements of the infrastructure (PHY-L), grouped into 5 categories: (i) IoT drivers (IoT-D) interfacing with various types of IoT devices, (ii) RAN drivers (RAN-D) interfacing with 5G new radio elements (e.g., entire 5G gNBs and/or open-RAN based components), (iii) network fabric drivers (NetFabric-D) interfacing with programmable network elements, such as SDN switches, whiteboxes, etc. (iv) Cloud drivers (Cloud-D) interfacing with virtual infrastructure management (VIM) and/or container infrastructure management (CIM) agents, and (vi) mobile core drivers (Mob-D) interfacing with 5G/4G core network functions. The behaviour of the drivers is managed by the respective set of controllers mentioned above, either through standardized protocols (e.g., P4 for network fabric controllers and drivers) or custom ones (e.g., for custom IoT devices tailored to a certain use case). A detailed description of the INCODE controllers and drivers is provided in Section 3.3.2.2.

3.1.3 INCODE SM-L: End-to-end service management across the IoT-Edge-Core compute continuum

Managing an infrastructure versus managing the services atop this infrastructure are two discrete operations in a real system. Different administrative entities are typically behind these operations, thus a typical way for these administrative entities to interact is through APIs. Section 3.1.2 already outlined how INCODE abstracts away the underlying hardware through an infrastructure "resource as a service" model.

At the service layer, service providers should focus on delivering high-level APIs to vertical end users/stakeholder, with the sheer purpose to facilitate service onboarding, end-to-end service deployment, runtime service monitoring, and service lifecycle management. This should be done in a seamless manner, while still offering end users the ability to associate end-to-end services with certain requirements, such as performance, locality, security, etc. To do so, INCODE introduces a service management layer (SM-L) as shown in Figure 3-3.

Co-funded by
the European Union

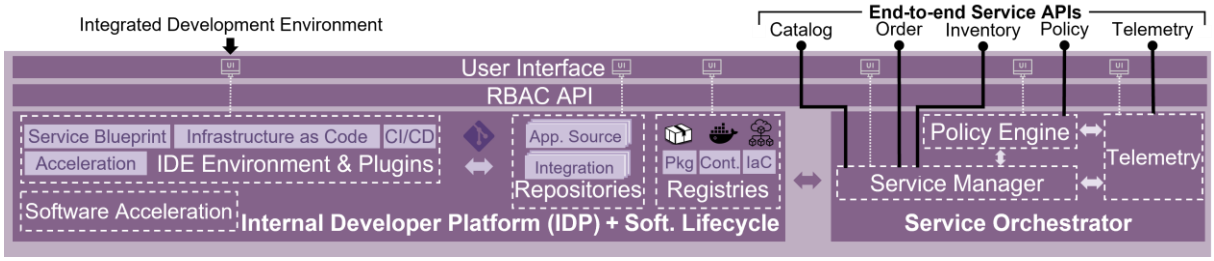

Figure 3-3: The end-to-end service management layer (SM-L) of the INCODE architecture across the IoT-Edge-Core compute continuum.

In short, the INCODE SM-L offers an internal developer platform (IDP) environment, coupled with state-of-the-art software lifecycle management tools, that allows developers and end users to programmatically turn their services' source code into deployable assets, which are automatically deployed upon a certain infrastructure using Infrastructure as Code (IaC) principles and standardized service and resource orchestration APIs. The INCODE SM-L consists of four logical blocks as shown in Figure 3-3; these blocks are briefly described next.

IDP and Software Lifecycle Management: INCODE envisions an environment that will embrace both end users with "ready-to-use" services and service developers (who wish to develop new services or extend existing ones) to seamlessly coexist and share abstractions that facilitate service development through an integrated development environment (IDE) as well as continuous integration & continuous delivery (CI/CD) of services atop an IaC-based environment. More details about this block are available in Section 3.3.1.4.

Service Orchestrator: INCODE couples the IDP and Software Lifecycle Management environment with a service orchestrator (SO) that takes requests for end-to-end service deployments through service blueprints and realizes these requests in collaboration with the RO (RM-L). First, IaC tools of the IDP prepare the environment prior to a service deployment; then the RO allocates infrastructure resources in this environment, which are in turn consumed by the SO for service deployment on behalf of the end users. Service-level telemetry is obtained by the SO during runtime, while a policy engine allows end users to associate services with service-level agreements (SLAs).

INCODE's Role-based access (RBAC) API acts as a centralized authority that allows different stakeholders (each with a different role) to access the corresponding INCODE services, either those exposed by the SM-L, RM-L, or the vertical security services to be described in the next section. More details about the RBAC API are provided in Section 3.3.1.2.

User Interfaces: At the top of the INCODE architecture, a set of user interfaces visualize key information for various stakeholders. Among the most important user interfaces, INCODE offers service catalog, service order, and service inventory UIs, a policy management UI, service and infrastructure telemetry portals, as well as third-party UIs for source code management, CI/CD, and an IDE for service developers. Additional information on the INCODE UIs is available in Section 3.3.1.1.

3.1.4 INCODE ST-L: Vertical security and trust services for the INCODE orchestration platform

Complex ecosystems such as those INCODE attempts to manage pose certain security and trust requirements. At the resource layer, a multi-vendor grid of devices appears across multiple edge sites, with IoT hardware heterogeneity being the most prevalent reason of increasing management complexity. Being able to register every IoT device in this complex ecosystem is of utmost importance. Moreover, attesting the firmware of each device adds another layer of security as it protects the platform from malicious attacks. At the service layer, applications need to consume data from the IoT domain as well as exchange data between

Co-funded by the European Union

each other. Auditing data and application components, while fortifying data exchange using encryption are also crucial security and trust services.

Because security and trust are important across all layers of the architecture, INCODE adopts a vertical stratum that offers security and trust services to both SM-L and RM-L. This vertical layer is formally defined as Security and Trust layer (ST-L). Four categories of security and trust services are deemed important for INCODE as shown in Figure 3-4 and briefly described below.

Figure 3-4: The vertical security and trust layer (SM-L) of the INCODE architecture.

Identity Management: Due to the vast heterogeneity of the underlying hardware, INCODE's RM-L requires physical and virtual device authentication, effectively protecting the platform from malicious attackers that may attempt to compromise certain devices. To effectively manage the identity of these devices, a distributed key-value store - organized as a blockchain - will be employed by this component. More details about the Identity Management (IM) service are provided in Section 3.3.3.1.

Attestation: To successfully register a device to the INCODE platform, this device must undergo an attestation process for effectively assuring the integrity of the device's firmware and configuration. The Attestation (AT) service works in tandem with the IM service to render the resources of a device available to the RO. More details about AT and its' synergy with the IM service are provided in Section 3.3.3.2.

Data Provenance: During service runtime, INCODE must ensure that data exchange between service components is secure and certain data never leak outside of a designated domain. More details about the Data Provenance service are provided in Section 3.3.3.3.

Attack Mitigation: At a post-onboarding phase, the INCODE RM-L employs microservices for detecting and mitigating adversarial attacks through data poisoning, and evasion attack mechanisms. More details about the Attack Mitigation service are provided in Section 3.3.3.4.

3.2 Overall Architecture

The combination of the service (SM-L) and resource (RM-L) management layers along with the vertical security and trust layer (ST-L) constitute the INCODE architecture as shown in Figure 3-5. This is an initial version of the INCODE ecosystem that captures all important aspects of the INCODE ecosystem, including the infrastructure, the resources exposed by the infrastructure, the overlay services, as well as auxiliary security and trust considerations. The proposed architecture takes a substantial leap with regards to the architecture described at the proposal time, both in terms of the number of internal components and exposed interfaces.

Co-funded by
the European Union

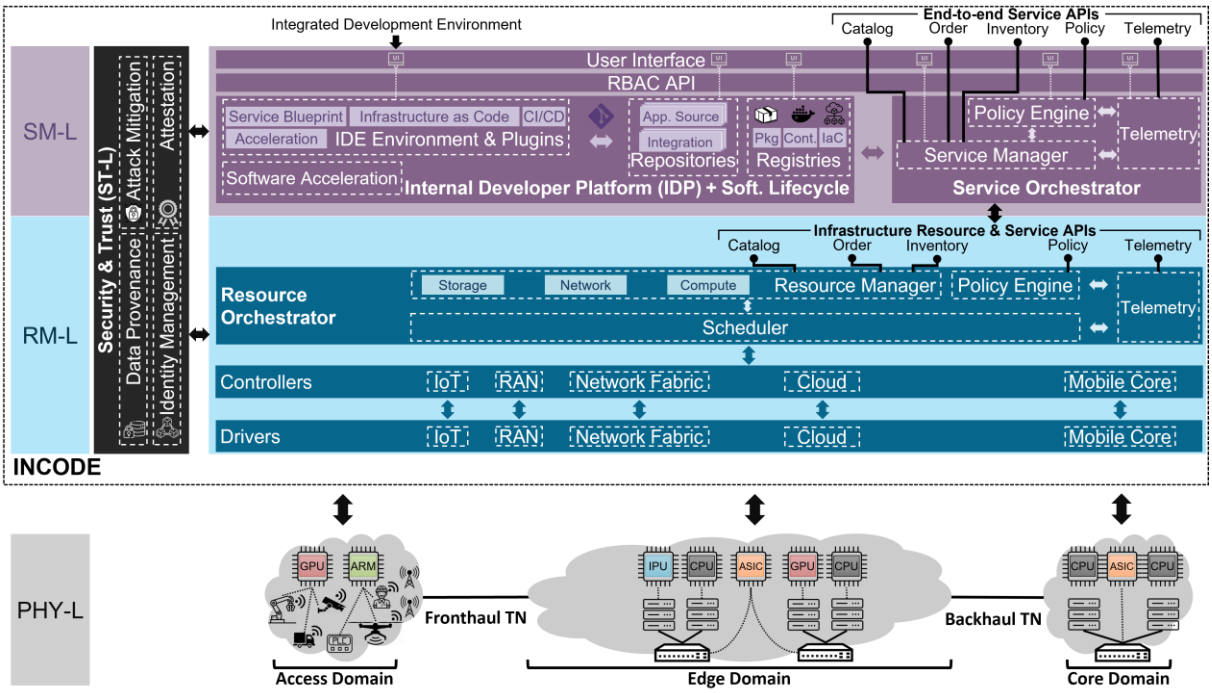

Figure 3-5: The initial version of the INCODE architecture.

In brief, RM-L is the singular layer directly interfacing with the underlying hardware by exposing high-level infrastructure resources and services to the overlay service layer. The service layer, in turn, utilizes RM-L services to streamline the onboarding, ordering, and provisioning of end-user applications via high-level end-to-end service APIs. This clear distinction between resource and service management enables diverse stakeholders to engage with specific components of the system without navigating unnecessary software layers. For instance, an infrastructure owner's primary interaction revolves around RM-L, focusing on efficient management of the underlying devices and resources. Conversely, a vertical end user, service provider, or developer may exclusively interact with the SM-L. Both RM-L and SM-L have the potential to leverage security and trust services provided by the vertical ST-L, based on their respective requirements. For example, the Service Orchestrator (SO), a component of the SM-L, may need to verify the software of a specific service component before deployment. On the other hand, the Resource Orchestrator (RO) may need to validate the firmware of an IoT device before onboarding it onto the platform.

3.2.1 Mapping of the INCODE architecture with technical WPs and Tasks

To break down the responsibilities for realizing this architecture, a mapping of the INCODE architecture with the key technical work packages and tasks is necessary. This mapping is visualized in Figure 3-6.

Co-funded by the European Union

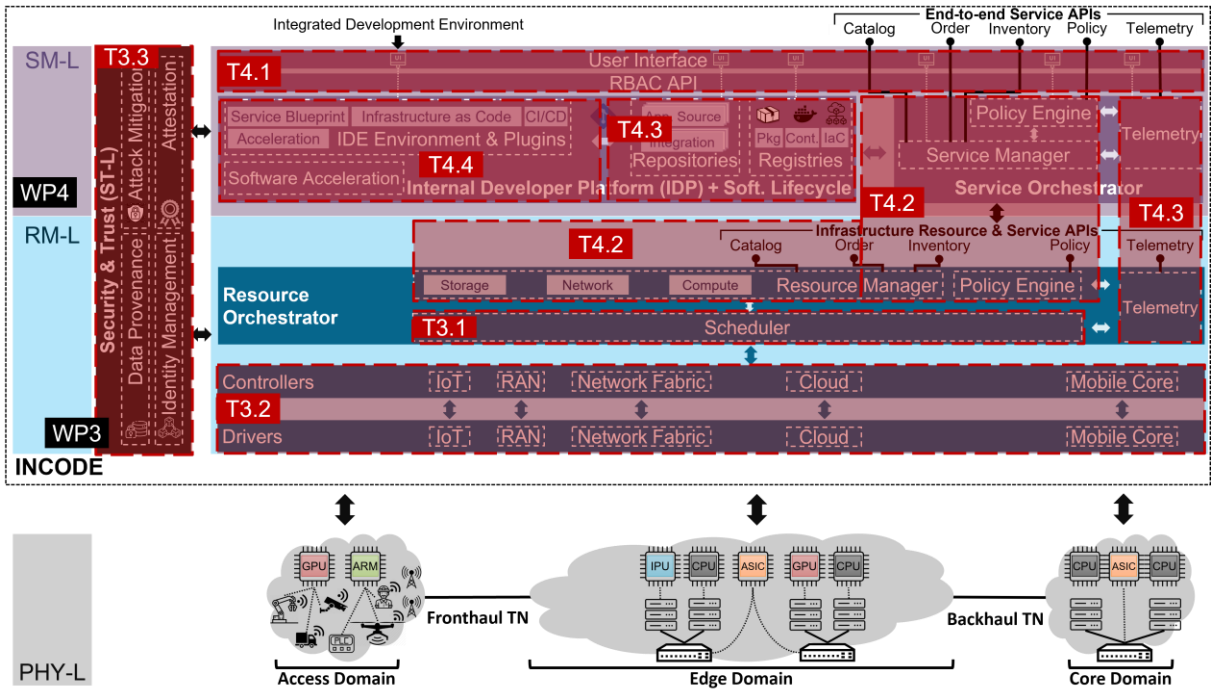

Figure 3-6: Mapping of technical work packages and tasks to the various parts of the initial version of the INCODE architecture.

In short, WP3 is responsible for the RM-L and ST-L layers, while WP4 is responsible for the SM-L. In WP3, Task 3.1 is responsible for the Scheduler logical block, while Task 3.2 oversees the design and development of the Controllers & Drivers. Task 3.3 is in charge of the vertical ST-L, despite the fact that services of this layer may also be consumed by the SM-L (WP4). On the other hand, Task 4.1 is responsible for the UI and RBAC API logical blocks of the SM-L, while Task 4.2 is responsible for the two orchestrators of the platform, each residing in a different layer (SO is part of SM-L and RO is part of RM-L). Task 4.3 oversees both the Software Lifecycle Management part of the IDP as well as the Telemetry blocks of the two orchestrators (service-level telemetry in the SO and resource-level telemetry in the RO). Finally, Task 4.4 deals with the IDP, the IDE environment & plugins, software acceleration and how these blocks interact with the Software Lifecycle Management (Task 4.3).

This way, the main technical WPs of the INCODE project fully cover the extent of the INCODE architecture. In the coming months, interactions between blocks of the architecture (and Tasks) will be specified by means on interfaces between the blocks. A preliminary list of interfaces for the INCODE architecture's logical components is already provided in the various sub-sections in Section 3.3.

3.3 Logical Components

This section outlines the logical components of the INCODE architecture, split across the three main layers of the architecture. Section 3.3.1 outlines the logical components of the INCODE Service Management Layer (SM-L), Section 3.3.2 those of the Resource Management Layer (RM-L), and finally, Section 3.3.3 those of the Security and Trust Layer (ST-L).

Co-funded by the European Union

3.3.1 Logical Components in the Service Management Layer

The SM-L constitutes the top part of the INCODE architecture (Figure 3-5). This service-oriented layer consists of four (4) logical components, detailed in the following four subsections 3.3.1.1-3.3.1.4.

3.3.1.1 User Interface

The User Interface (UI) is a key logical component in the Service Management Layer. It acts as the front-end interface, enabling users to interact with the INCODE platform's services. The UI provides a user-friendly interface for various stakeholders, including application providers and operations teams, facilitating access and management of platform services. It handles user authentication and access control to ensure users only access relevant functionalities based on their roles.

Table 3-1: List of UI views provided by the INCODE User Interface.

UI View Name	UI View Objective	Integration with Backend Service
Service Catalog	Manage the services stored in INCODE's service catalog. Management operations include Create, Read, Update, and Delete (CRUD) of service specifications which are the main items of the service catalog.	**Logical Block:** Service Orchestrator **Component of logical block:** Service Manager **API:** Service Catalog API
Service Order	Users, such as operations teams, can use this view to create, update, and manage service orders related to the INCODE platform's services. This might include tasks like initiating service deployments, tracking service orders, and resolving issues related to service provisioning.	**Logical Block:** Service Orchestrator **Component of logical block:** Service Manager **API:** Service Order API
Service Inventory	The service inventory view provides the front-end interface for users to explore, configure, and work with services efficiently within the INCODE platform. It allows to retrieve information about the available services, their configurations, versions, and related data. It can also use the API to perform configurations and adjustments as needed.	**Logical Block:** Service Orchestrator **Component of logical block:** Service Manager **API:** Service Inventory API
Service Policy	The service policy view provides the front-end interface for users to configure and manage policies effectively within the INCODE platform. It allows to perform actions such as creating, updating, and deleting policies, associating policies with services, retrieving policy details, and monitoring policy enforcement.	**Logical Block:** Service Orchestrator **Component of logical block:** Policy Engine **API:** Policy API
Service Telemetry	The primary objective is to provide users with a user-friendly interface for visualizing telemetry data generated by services within the INCODE platform. Telemetry data can include various metrics, logs, and monitoring information.	**Logical Block:** Service Orchestrator **Component of logical block:** Telemetry **API:** Telemetry API

Co-funded by the European Union

Resource Telemetry	The primary objective is to provide users with a user-friendly interface to monitor the status, performance, and utilization of various resources within the INCODE platform. These resources may include computing resources, networking components, and storage systems.	**Logical Block:** Service Orchestrator **Component of logical block:** Telemetry **API:** Telemetry API
Artefacts Registries	The Artifacts Registries view interacts with TM Forum APIs to perform actions related to artifact discovery, management, and retrieval. These artifacts include network configurations, service templates, device profiles, and other resources.	**Logical Block:** IDP and Soft. Lifecycle **Component of logical block:** Registries **API:** Registry API

Stakeholders

Vertical end-users, as a critical stakeholder group for the User Interface (UI), encompass individuals or entities operating within specific industry sectors. They possess distinct and specialized needs, workflows, and objectives tied to their respective industries, requiring a UI tailored to address these unique requirements. User-centric design principles are paramount in crafting interfaces that streamline their industry-specific tasks, offering customization options, industry compliance, and seamless integration with relevant tools. Ensuring the UI's performance, reliability, and compliance with industry regulations is vital, while continuous feedback and adaptation to evolving industry trends are key to enhancing user satisfaction and productivity within their vertical contexts.

Vertical end-users are expected to interact mostly with the **Service Catalog** and **Service Inventory** views. The Service Catalog provides them with an accessible repository of industry-specific services and offerings, aiding in efficient discovery and selection of services that align with their sector's needs. Meanwhile, the Service Inventory view offers real-time visibility into the status and performance of services, helping vertical end-users to monitor and manage services crucial for their operations, ensuring optimal performance and resource allocation. These views empower vertical end-users with the tools they need to navigate and harness industry-specific services effectively, ultimately enhancing their productivity and enabling them to better serve their vertical's unique requirements.

Vertical service providers are a crucial stakeholder group for the User Interface (UI), including organizations or individuals responsible for offering and managing services within a given platform or ecosystem. They rely on the UI to streamline service provisioning, configuration, and monitoring processes. Service providers require user-friendly interfaces to efficiently manage service catalog offerings, define and enforce service policies, and access telemetry data to ensure optimal service delivery. The UI should facilitate service lifecycle management, from creation and customization to real-time monitoring and reporting. Furthermore, it should support role-based access control to enable service providers to fulfil their responsibilities effectively while maintaining security and compliance standards. In summary, service providers rely on the UI as a central tool to facilitate service management, ensuring quality, performance, and adherence to service policies within the platform or ecosystem they operate in.

The **Service Catalog** view enables service providers to efficiently manage and showcase their service offerings, simplifying the process of adding, modifying, and presenting services to end-users. The **Service Order** view aids in the organized provisioning of services, ensuring smooth service delivery to customers. The **Service Policy** view allows providers to enforce and manage policies related to service performance and security. **Service Telemetry** and **Resource Telemetry** views provide real-time insights into service and resource performance,

Co-funded by the European Union

facilitating proactive monitoring and issue resolution. Lastly, the **Artefacts Registries** view assists in managing and accessing essential artifacts and configurations, streamlining the deployment of services. Together, these views empower service providers to deliver high-quality services, maintain security, and optimize resource allocation, ultimately enhancing their effectiveness and the overall customer experience.

Vertical service developers are a stakeholder group for the User Interface (UI) representing individuals or teams responsible for creating, customizing, and deploying services within the platform. The UI serves as their primary workspace, providing essential tools and views to streamline the development lifecycle. Service developers rely on the UI to access the **Service Catalog** for selecting and configuring service components, the **Service Policy** view for defining and implementing performance and security policies, and the **Artifacts Registries** view for managing and integrating reusable components into their services. Additionally, access to **Service Telemetry** and **Resource Telemetry** views is essential for monitoring and optimizing service performance. The UI should support collaboration and version control, ensuring efficient development, testing, and deployment of services while maintaining quality and compliance standards. Overall, the UI plays a pivotal role in enabling service developers to create and deliver innovative, reliable, and industry-specific services within the platform.

Infrastructure providers are a stakeholder group comprising organizations or individuals responsible for managing the underlying hardware and resources within the platform's infrastructure. The UI is essential for infrastructure providers as it facilitates resource allocation, monitoring, and maintenance. Infrastructure providers rely on the UI to optimize resource utilization and allocate computing, networking, and storage resources efficiently. The **Resource Telemetry** view provides real-time insights into resource performance and usage, aiding in proactive monitoring and capacity planning. Additionally, access to the **Service Policy** view allows infrastructure providers to enforce resource-related policies and security measures. The UI should support seamless integration with infrastructure management tools and offer comprehensive control over resources, ensuring robust and reliable infrastructure operations to meet the diverse needs of other stakeholders in the ecosystem.

Infrastructure owners are a stakeholder group representing entities or individuals responsible for the ownership, maintenance, and strategic management of the physical infrastructure that underpins the platform. The UI plays a critical role in supporting infrastructure owners' objectives by providing tools and insights necessary for efficient infrastructure management. Infrastructure owners rely on the UI to gain visibility into resource utilization through the **Resource Telemetry** view, ensuring optimal performance, scalability, and cost-effectiveness. They use the **Service Policy** view to enforce governance and compliance standards across the infrastructure. The UI should facilitate effective decision-making by offering comprehensive reporting and analytics, helping infrastructure owners optimize resource allocation, reduce operational costs, and ensure the long-term sustainability and resilience of the platform's underlying infrastructure.

Hardware and software vendors, representing companies or individuals who provide the technology components and solutions that integrate with or support the platform, interact with the UI as a gateway to showcase their offerings, validate compatibility with the platform through the **Artifacts Registries** view, and gain insights into the utilization of their hardware or software through the **Resource Telemetry** and **Service Telemetry** views. The UI should offer easy integration and certification processes, enabling vendors to seamlessly integrate their products and solutions with the platform. Additionally, it should provide mechanisms for vendors to access user feedback and analytics, facilitating product refinement and alignment with the evolving needs of other stakeholders within the ecosystem. Ultimately, the UI plays a pivotal

Co-funded by
the European Union

role in fostering collaboration and innovation between vendors and the platform's user community.

System integrators, representing entities or individuals responsible for integrating and configuring various components and technologies within the platform to create comprehensive solutions for end-users, use the UI as their primary tool for orchestrating the integration process efficiently. System integrators rely on the UI to access the **Service Catalog** and **Artifacts Registries** views to select and incorporate services, components, and configurations that align with the specific needs of their clients. The **Service Policy** view helps them implement and customize policies to ensure that the integrated solutions meet performance and security requirements. The UI should provide collaboration features, version control, and documentation access, enabling system integrators to streamline their integration projects while maintaining quality and compliance standards. Overall, the UI empowers system integrators to deliver seamlessly integrated solutions that fulfil the unique requirements of their clients within the platform ecosystem.

The UI seamlessly interfaces with the API Management Service Hub, providing a centralized entry point for users to consume and manage APIs. It collaborates with other logical components, such as the Runtime Orchestration Service and Software Lifecycle Service, to deliver a comprehensive user experience within the Service Management Layer (SM-L).

3.3.1.2 RBAC API

The API Management Service Hub/RBAC-based API gateway acts as a central platform that manages the entire API lifecycle, focusing on role-based access control (RBAC) for securing and regulating API access. It acts as an entry point for external users and third-party developers to access APIs exposed by the INCODE platform. It enforces RBAC policies to control access to APIs, ensuring that only authorized users with appropriate roles can interact with specific API resources.

The hub provides authentication and authorization mechanisms, such as OAuth 2.0 or JWT, to validate users' identities and grant access based on their assigned roles. It also manages API documentation, versioning, and analytics, providing a holistic solution for API consumers and providers.

By serving as an RBAC-based API gateway, the API Management Service Hub enhances security, allows for fine-grained access control, and ensures that only authenticated and authorized users can interact with the APIs exposed by the INCODE platform.

Table 3-2: List of APIs provided through the INCODE RBAC API hub.

API name	API Objective	Provided by
Authentication and Authorization	*Offer authentication and authorization APIs to validate the identity of external users and third-party developers. These APIs should enable the enforcement of RBAC policies, ensuring that only authorized users with the appropriate roles can access specific SM-L and RM-L APIs. The objective is to secure and regulate API access, safeguarding sensitive data and resources.*	*Security & Trust Layer (ST-L)*
RBAC Management	*Allowing administrators to define and configure role-based access control policies. These APIs enable the setup and modification of RBAC rules, ensuring that API access aligns with the*	*Security & Trust Layer (ST-L)*

Co-funded by the European Union

	organization's security and compliance requirements.	
Security and Threat Protection	*Security APIs to protect against common security threats, such as API security vulnerabilities and attacks. These APIs aim to enhance the overall security posture of API access to the SM-L and RM-L APIs.*	*Security & Trust Layer (ST-L)*
Developer Onboarding and Key Management	*APIs to facilitate the registration of external developers, provide them with API keys, and manage developer profiles. The objective is to streamline the onboarding process and ensure secure API access.*	*Security & Trust Layer (ST-L)*
Usage Monitoring and Analytics	*APIs to enable external users and third-party developers to track their API usage, gain insights into performance, and identify areas for optimization. The objective is to improve the quality of integrations and user experiences.*	*Service Management Layer (SM-L) Resource Management Layer (RM-L)*
Discovery and Documentation	*APIs for API discovery and documentation, allowing external users and developers to explore available TM Forum compliant APIs, understand their functionality, and access comprehensive documentation, to facilitate easier API consumption and integration.*	*Service Management Layer (SM-L) Resource Management Layer (RM-L)*
Rate Limiting and Throttling	*APIs to prevent abuse and ensure fair usage. The objective is to maintain API performance and availability.*	*Service Management Layer (SM-L) Resource Management Layer (RM-L)*
Versioning and Lifecycle Management	*APIs for API versioning and lifecycle management provided to external users and developers. Allowing to keep track of API versions, ensuring that their integrations remain compatible. The objective is to support the smooth evolution of API integrations.*	*Service Management Layer (SM-L) Resource Management Layer (RM-L)*

3.3.1.3 Service Orchestrator

The Service Orchestrator is a central component in INCODE's SM-L as it provides the ability to manage end-to-end services atop 5G resources across heterogeneous and distributed domains in the IoT-to-edge-to-cloud continuum.

The northbound API of the SO allows vertical end-users and service providers to express end-to-end services using an open and standardized service specification API. The services created using this API are stored into a Service Catalog (service onboarding), where the INCODE stakeholders can select a service of their choice and order it for deployment (service ordering). More importantly, with a proper integration of the SO's APIs with the INCODE IDP (see Section 3.3.1.4), service onboarding and service ordering can be completely automated.

In the INCODE project, the SO is divided into three logical subcomponents each one being responsible for separate parts of the orchestration procedure, such as service management (Section 3.3.1.3.1), service telemetry (Section 3.3.1.3.2) and service lifecycle management using policies (Section 3.3.1.3.3). These components are explained in the rest of this section.

Co-funded by the European Union

3.3.1.3.1 Service Manager

The SM is responsible for the entire lifecycle management of end-to-end vertical services atop INCODE's 5G-based infrastructures across the IoT-to-edge-to-cloud computing continuum. The service lifecycle entails three main operations described in the following paragraphs.

Service onboarding on the SO's catalog is a process that requires an already packaged service to be associated with a service onboarding descriptor that guides the SO on how to incorporate this service into its service catalog. The data models for the service specification, category, and catalog will follow the TMF633 Service Catalog Management API [38] model schema and will be accessible through standardized API endpoints as shown in Table 3-3.

Service ordering from the SO's catalog.is a process that entails the selection of an already onboarded service from the SO's service catalog and the ordering of this service using important runtime information. This information may contain e.g., the number of resources required by the selected service, potential area of service for service components, etc. The data models for the service order will follow the TMF641 Service Ordering API [39] model schema and will be accessible through standardized API endpoints as shown in Table 3-3.

Service inventory for runtime management once an instance is deployed on a given (set of) domain(s). The data models for the service inventory will follow the TMF638 Service Inventory Management API [40] model schema and will be accessible through standardized API endpoints as shown in Table 3-3.

A key integration point of the Service Manager to achieve a successful service order is through interaction with the Resource Orchestrator (see Section 3.3.2.1). This is important because resource allocation is a process strictly performed by the INCODE RM-L, where the RO plays a central role. Therefore, to deploy a service with enough resources, the SO parses a service order request and attempts to identify three important pieces of information: (i) the amount of compute resources required by the service to operate, (ii) potential networking requirements for ensuring certain QoS (e.g., certain delay budget between a 5G end user and a specific service component), and (iii) location-specific requirements, such as a desired 5G area of service or the placement of a service component into a specific domain for security purposes (i.e., because the data of this service is not allowed to flow outside of this domain). These 3 pieces of information formulate two concomitant service order requests that the SO dispatches to the RO, asking for compute and network resources. This interaction between the two INCODE orchestrators is done through the same TMF-based APIs (service catalog, order, inventory) and comprises an internal process that the end user is totally unaware of. The detailed workflow for service onboarding, end-to-end service ordering, and service inventory will be provided in future deliverables within the context of WP3 and WP4.

Table 3-3 lists all the APIs that will be exposed by the SM and can be used to enable its integration with (i) the INCODE stakeholders towards the northbound and (ii) the underlying resource orchestration layer (i.e., RM-L) for coordinated resource allocation for end-user services. All these APIs comply with TMForum's open and standardized APIs.

Table 3-3: List of TMF-based northbound APIs exposed by the INCODE SM-L.

Endpoint	Title	Description	Version
/tmf-api/service-catalog-management/v4	633 Service Catalog Management	Provides a catalog of services.	4.0.0
/tmf-api/service-ordering-management/v4	641 Service Ordering	Provides the ability to manage service orders.	4.0.0

Co-funded by the European Union

/tmf-api/service-inventory-management/v4	638 Service Inventory Management	Provides the ability to query and manipulate the service inventory.	4.0.0
/tmf-api/service-quality-management/v2	657 Service Quality Management	Provides the ability to manage quality of service.	2.0.0
/tmf-api/service-activation-management/v3	640 Service Activation Management	Provides the ability to activate and configure services.	3.0.0
/tmf-api/service-test-management/v4	653 Service Test Management	Provides the ability to manage tests of provisioned services.	4.0.0
/tmf-api/resource-catalog-management/v4	634 Resource Catalog Management	Provides a catalog of resources.	4.0.0
/tmf-api/resource-order-management/v4	652 Resource Order Management	Provides the ability to order resources.	4.0.0
/tmf-api/resource-inventory-management/v4	639 Resource Inventory Management	Provides the ability to query and manipulate the resource inventory.	4.0.0
/tmf-api/resource-pool-management/v1	685 Resource Pool Management	Provides the ability to manage resource reservation in a pool of resources.	1.0.0
/tmf-api/sla-management/v2	625 SLA Management API	Provides the ability to manage service-level agreements.	4.0.0
/tmf-api/agreement-management/v2	651 Agreement Management	Provides the ability to manage agreements, especially in the context of partnerships between partners.	2.0.0
/tmf-api/alarm-management/v4	642 Alarm Management	Provides the ability to manage alarms.	4.0.1
/tmf-api/lcm-rules-management/v1	LCM Rules	Custom API for rule-based service LCM.	1.0.0

3.3.1.3.2 Telemetry

Modern services require not only neat orchestration but also runtime tools that facilitate service observability and the potential detection of failures. To this end, INCODE provides service-level telemetry as part of the Service Orchestrator (SO). For every service component that gets deployed by INCODE on a given infrastructure, telemetry agents are installed at designated endpoints to facilitate the collection of relevant monitoring data for this specific service component.

In the context of modern cloud-native deployments, which are relevant for INCODE, every component of a service may be packaged as a standalone container image following the trend of microservice-based services. In this case, the telemetry module of the SO places a dedicated monitoring agent per container which collects component specific monitoring data and reports this data to a metrics collector. The collector persists this information into a monitoring database and makes this available to external components through APIs.

A relevant external component could be the Policy Engine who may wish to consult this information for affecting a service's lifetime through specific actions as described in the next section. Other components may also consume telemetry information for data visualization purposes (e.g., a Grafana dashboard), analytics, etc. More information of service-level telemetry will be provided in the context of WP4 deliverables.

Co-funded by the European Union

3.3.1.3.3 Policy Engine

An important aspect of a service's runtime management is the ability to intervene in its lifecycle management (LCM) states through, for example, runtime policies that may affect the way the service behaves based on dynamic events or specific conditions. The INCODE SO makes this possible through a dedicated Policy Engine component. This component one the one hand integrates with the Telemetry block of the SO to register for certain service-level alarms according to the needs of the service operator. For example, a service operator may want to receive an alarm when the CPU utilization of a certain service component exceeds a predefined threshold. When such a conditional alarm is raised, the Policy Engine integrates with the Service Manager to perform certain actions to the service. For example, when the previous alarm is raised for a service, a possible action could be to scale the service component out by creating another replica. The INCODE SO provides additional endpoints for LCM management as shown at the bottom of Table 3-3. Further details about this component of the SO will be provided in WP4 deliverables.

3.3.1.4 Internal Development Platform & Software Lifecycle Management

An Internal Developer Platform (IDP) is a comprehensive platform that empowers development teams to efficiently build, deploy, and manage applications. It encompasses a range of logical components within the Software Development Lifecycle (SDLC). The IDP provides self-service capabilities and a curated set of tools, services, and infrastructure, enabling developers to focus on software development instead of dealing with the complexities of underlying infrastructure. By leveraging the principles of Infrastructure as Code (IaC) and Continuous Integration/Continuous Deployment (CI/CD), the IDP automates various processes and accelerates service delivery.

In the example depicted in Figure 3-7, the IDP supports the software lifecycle of four applications: Application Area 1 (AA1), Application Area 2 (AA2), Application Area 3 (AA3), and Application Area 4 (AA4). For each application, there should be a dedicated source repository, as well as a repository for integration. Additionally, there should be package registries, container registries, and Infrastructure as Code (IaC) registries. These components work together to facilitate the CI/CD processes for each application.

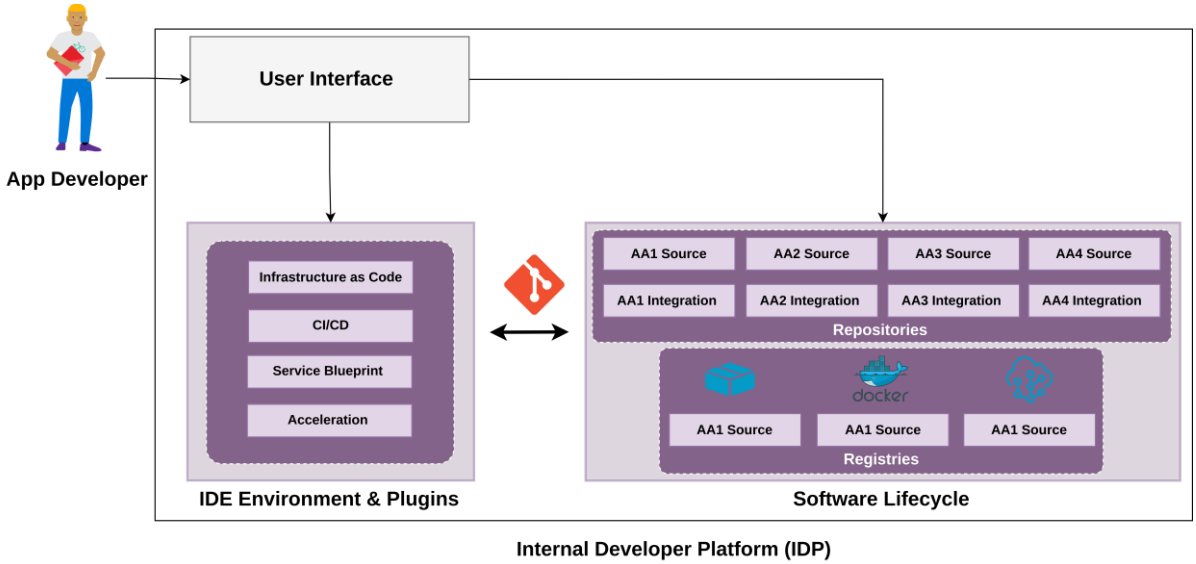

Figure 3-7: INCODE's *Internal Development Platform overview.*

The IDP embraces the principles of Infrastructure as Code (IaC), empowering teams to programmatically define and manage infrastructure resources. By leveraging tools, such as OpenTofu [41], the IDP enables efficient provisioning and management of infrastructure, significantly reducing manual configuration efforts.

In the IaC workflow illustrated in Figure 3-8, it is essential to have dedicated resources in the infrastructure for each application area in INCODE. Taking the example of the Application Area 4 (AA4) use case, when triggered, the CI/CD process for AA4 will result in an integration repository containing all the necessary information and configurations in which will be sent to the CI/CD Infrastructure Deployment Pipeline.

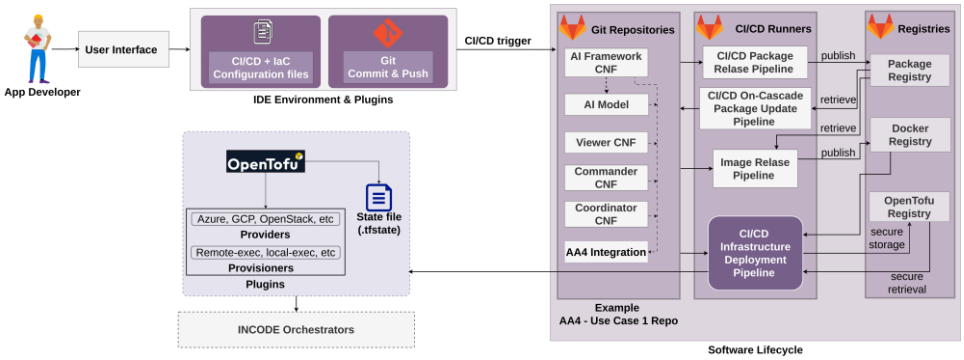

Figure 3-8: INCODE's IDP using Infrastructure as Code principles.

Simultaneously, the deployment of the infrastructure should include AA4's Docker image. Additionally, the CI/CD Infrastructure Deployment Pipeline should interact with the Terraform registry to store and retrieve the status of AA4's infrastructure. This allows for efficient monitoring and updating of the infrastructure as needed. OpenTofu will interact with the underlying infrastructure specially INCODE orchestrators to deploy the same infrastructure indicated in the code.

3.3.1.4.1 Repositories

Throughout the software development lifecycle, repositories provide an efficient way for collaboration, version control and automation. These repositories enable the storage and management of source code, infrastructure-as-code configurations, as well as other project assets. An integration component ensures code quality and seamless interactions across various software components. Integration tests are triggered within GitLab's CI/CD pipelines to verify the compatibility between different services within each application. SIEMENS is the key partner to provide the INCODE repositories, which will be extensively discussed in D2.2.

3.3.1.4.2 Registries

The package registry component is a public or private software package registry. Multiple package managers are supported by this registry, and several functionalities are made available for these packages, such as publishing and sharing from within a CI/CD pipeline. Moreover, request forwarding, and duplicate management is possible, as well as authentication depending on the package manager being used.

The container registry component facilitates the storage and management of container images. Every GitLab project has its own container registry. The container images are associated with a project and any group that project has been added to. These container images are built and published to the registry from a GitLab pipeline. The infrastructure registry supports Terraform modules. A Terraform module is built and published from a GitLab CI/CD pipeline. This module provides a reusable way to define and manage cloud infrastructure resources. SIEMENS is the key partner to provide the INCODE registries, which will be extensively discussed in D2.2.

Co-funded by the European Union

3.3.1.4.3 IDE Environment & Plugins

The IDE environment provides the developer with a set of features and plugins designed to empower the software development process. Among its core components are Infrastructure as Code (IaC), Continuous Integration/Continuous Deployment (CI/CD), Self-Service Tooling, and Acceleration. These components work together to not only simplify but also help with the development work.

- **Infrastructure as Code (IaC)** is a transformative approach to managing infrastructure resources by defining and deploying them through code. Instead of the conventional manual processes, IaC allows teams to manage infrastructures using code, which can be version-controlled and automated. In our context, IaC is embraced with the principles of IaC withing the IDE. Popular tools like OpenTofu [41] can be leveraged to achieve this goal. OpenTofu is an open-source tool used for building, changing and versioning infrastructure efficiently. The IDE simplifies the process of defining infrastructure code, with suggestions or corrections, making it easier for developers to work with OpenTofu or similar tools to create and manage infrastructures.
- **Continuous Integration/Continuous Deployment (CI/CD):** The IDE allows developers to define CI/CD pipelines within the IDP. These pipelines seek to automate essential steps in the software development lifecycle such as building code, running test and deploying applications. As shown in Figure [33], the IDE incorporates a CI/CD plugin that makes easier the definition of the software lifecycle within the SM-L.
- **Self-Service Tooling** capabilities in the IDP empower developers to perform certain tasks autonomously without requiring extensive manual intervention. Examples the tools include container registries for managing container images, package registries for managing software dependencies, and other potential services that developers need. With self-service capabilities, bottlenecks are reduced as well as productivity is improved.
- **Software Acceleration:** The IDE includes a plugin designed to assist developers in optimizing their code for the best hardware acceleration. The plugin likely provides hints or suggestions to help developers leverage hardware acceleration effectively, ensuring that their code runs as efficient as possible. It is further explained in the next section.

3.3.1.4.4 Software Acceleration

The INCODE Software Acceleration block comprises of four main components, (i) a *task-based API*, (ii) the *Software Acceleration Runtime*, (iii) the *Software Acceleration Execution Engine,* and (iv) the *Software Acceleration Compiler Extensions*. The Software Acceleration module API offers a gateway to the transparent generation of heterogeneous code, optimized for their target device, with no user intervention. Internally, upon the invocation of the function to be accelerated, the Software Acceleration Runtime creates a data-flow graph, by analysing the data dependencies in the Task-Graph, and it generates a set of bytecodes to orchestrate the execution. These bytecodes are interpreted by the Software Acceleration Execution Engine, which then offloads the bytecode representation of each method to its corresponding target backend. Finally, the optimized code produced is cached by the Software Acceleration Runtime to be reused in case of multiple launches. Figure 3-9 illustrates the software stack of the Software Acceleration module.

Co-funded by
the European Union

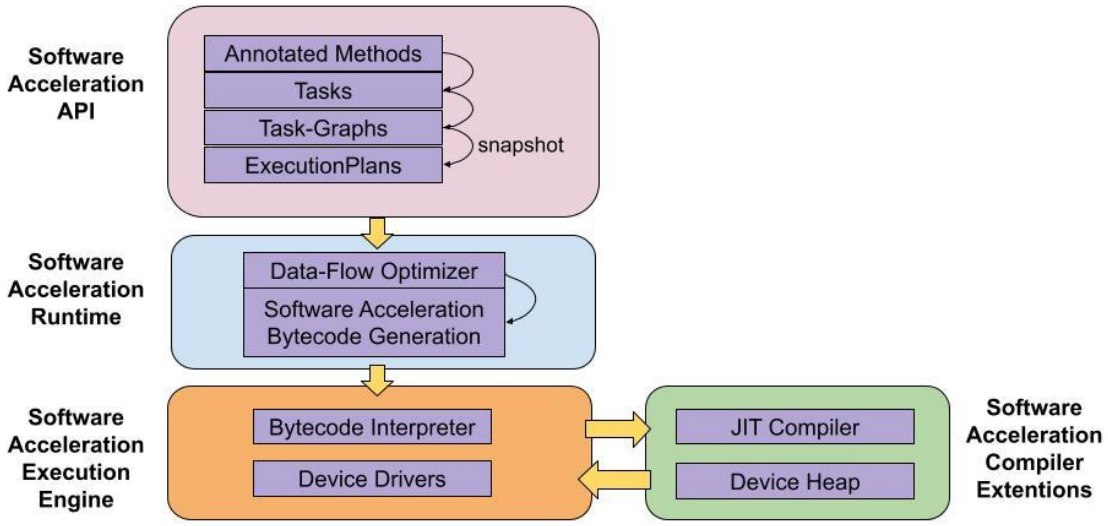

Figure 3-9: Software Acceleration Architecture.

Software Acceleration in the INCODE Project

The INCODE methods that will be candidates for acceleration with the Software Acceleration block will use the @Parallel annotation to mark the data-parallel loops of the methods. Table 3-4 lists the API that will be provided by the Software Acceleration module to facilitate the acceleration of the INCODE candidate use cases.

Table 3-4: List of Software Acceleration APIs.

Endpoint	Title	Description
uk/ac/manchester/tornado/api	Task Graph	Groups tasks to orchestrate and optimize the data transfers from the host CPU to the heterogeneous devices.
uk/ac/manchester/tornado/api	Immutable Task Graph	Used to create an immutable version of a Task Graph.
uk/ac/manchester/tornado/api	Execution Plan	Triggers the execution and can be used to enable profiling and to fine tune the execution.

3.3.2 Logical Components in the Resource Management Layer

This section introduces the logical components of INCODE's RM-L. Section 3.3.2.1 describes the Resource Orchestrator (RO) logical components and its blocks, while Section 3.3.2.2 lists the set of infrastructure controllers and drivers that the RO exploits to manage the underlying infrastructure.

3.3.2.1 Resource Orchestrator

The Resource Orchestrator (RO) is a software component expressing the notions of management and orchestration of any kind of resources in the spectrum of RM-L. The RO is responsible to provide an end-to-end slice by combining various types of computes, network and storage infrastructure components in order to satisfy the specifications provided by the services or the upper layers of management.

Co-funded by the European Union

In terms of interoperability with the rest of the platform, RO is equipped with a northbound interface which advertises the available resources to the Service Orchestrator and provides runtime slice information after its creation. It is also equipped with a southbound interface to communicate with the Controllers of each domain. Finally, a westbound interface is also deployed to communicate with the security blocks of the platform, while an eastbound interface is used to publish the telemetry metrics to the central telemetry module.

In the INCODE project, the RO is divided in four logical subcomponents each one being responsible for separate parts of the orchestration procedure, such as resource allocation, dynamic scaling and load balancing. Those components are:

- Resource Manager (RM), which is mainly responsible for holding the inventory and the catalogue of the available resources and bundles of resources. It is also responsible for the creation of resource orders.
- Policy Engine, which is responsible for the enforcement of policies inherited by the service orchestrator or directly indicated to the RO.
- Scheduler, which is responsible for the deployment and the runtime management of the resource orders provided by the RM over the Controllers and the Drivers on the underlying infrastructure.
- Telemetry module, which is responsible to provide the performance metrics of the infrastructure occupied by the RO.

The detailed description of those components follows in the sections below.

3.3.2.1.1 Resource Manager

The implementation of the RM is going to support a mechanism through which a developer or infrastructure manager will be able to define resource specifications for underlying resources. These specifications are going to be able to be categorized accordingly and get exposed to a service catalog. The data models for the resource specification, category and catalog will follow the TMF634 Resource Catalog Management API model schema and will be accessible through standardized API endpoints. During the instantiation of a resource the mechanisms of TMF639 Resource Inventory API will be triggered to create a new resource according to its resource specification and add it to a resource inventory. To accommodate external services, resources will be handled as services and will be used to offer Resources as a Service (RaaS). To achieve this goal Resource Facing Service Specifications (RFS) will be created based on the TMF model for service specifications (TMF633 Service Catalog). The service specifications can have a direct link to all the underlying resource specifications that need to be bundled in order to design an elaborate resource. The service specifications will, similarly to the resource specification, be instantiated to services through the mechanisms of the TMF 638 - Service Inventory Management API. All the resources and services created in the RM will be exposed to an internal orchestrator. The running resource facing services will be orchestrated and managed through the same orchestrator.

Table 3-5 lists all the northbound APIs that will be exposed by the RM and can be used to enable its integration with other parts of the INCODE platform. All these APIs are compliant with TMForum's open and standardized APIs.

Table 3-5: List of TMF-based northbound APIs exposed by the INCODE RM-L.

Endpoint	Title	Description	Version
/tmf-api/serviceCatalogManagement/v4	633 Service Catalog Management	Provides a catalog of services.	4.0.0
/tmf-api/serviceOrdering/v4	641 Service Ordering	Provides a standardized mechanism for managing Service Order.	4.0.0

Co-funded by the European Union

/tmf-api/serviceInventory/v4/	638 Service Inventory Management	Provides a consistent/standardized mechanism to query and manipulate the Service inventory.	4.0.0
/tmf-api/serviceQualityManagement/v2	657 Service Quality Management	This is Swagger UI environment generated for the TMF Service Quality Management specification.	2.0.0
/tmf-api/ServiceActivationAndConfiguration/v3	640 Service Activation and Configuration	Provides the ability to activate and configure Services.	3.0.0
/tmf-api/serviceTestManagement/v4	653 Service Test Management	Provides the ability to manage tests of provisioned Services.	4.0.0
/tmf-api/resourceCatalogManagement/v4	634 Resource Catalog Management	This is Swagger UI environment generated for the TMF Resource Catalog Management specification.	4.0.0
/tmf-api/resourceOrderingManagement/v4	652 Resource Order Management	This is Swagger UI environment generated for the TMF 652-Resource Order Management-v4.0.0 specification.	4.0.0
/tmf-api/resourceInventoryManagement/v4	639 Resource Inventory Management	Provides the ability to manage Resources.	4.0.0
/tmf-api/resourcePoolManagement/v1	685 Resource Pool Management	Resources that can be reserved are only in one pool.	1.0.0
/tmf-api/productCatalogManagement/v4	620 Product Catalog Management	Provides a catalog of products.	4.0.0
/tmf-api/productOrderingManagement/v4	622 Product Ordering	Provides a standardized mechanism for placing a product order.	4.0.0
/tmf-api/customerManagement/v4	629 Customer Management	TMF Customer Management	4.0.0
/tmf-api/partyRoleManagement/v4/	669 Party Role Management	This is Swagger UI environment generated for the TMF Party Role Management specification.	4.0.0
/tmf-api/party/v4/organization	632 Party	Provides standardized mechanisms for party management such as creation, update, retrieval, deletion and notification of events.	4.0.0
/tmf-api/agreementManagement/v2	651 Agreement Management	This is Swagger UI environment generated for the TMF Agreement Management specification.	2.0.0
/tmf-api/accountManagement/v4	666 Account Management	This is Swagger UI environment generated for the TMF Account Management specification.	4.0.0

Co-funded by the European Union

/tmf-api/userinfo	691 Federated ID	TMF Federated ID	1.0.0
/tmf-api/alarmManagement/v4	642 Alarm		4.0.0
/tmf-api/lcmrulesmanagement/v1	LCM Rules	Custom API environment for LCM Rules	1.0.0

3.3.2.1.2 Distributed Storage

For distributed storage, INCODE will use Ceph [42] - an open-source project that provides a scalable, highly available and high-performance storage solution. Ceph addressed all needs for distributed storage: file, block and object storage. It provides POSIX-compliant semantics for file access, as well as S3- and Swift-compliant APIs for object stores. In addition, it integrates easily as a storage provider for Kubernetes persistent volumes. The resource orchestrator may allocate compute and distributed storage resources through Kubernetes, on which Ceph will be assigned as a storage subsystem.

3.3.2.1.3 Policy Engine

At the resource management level, the INCODE platform offers a policy engine block as shown in Figure 3-10. Policies are sets of rules, guidelines, or principles that dictate how certain actions or behaviours should be handled. These policies pertain to security, access control, compliance, resource allocation, and more. The intention of this engine will be to schedule a set of event-based actions performed during the lifecycle of a service. The way this will be implemented, is through an internal to the developer portal, where the maintainer and/or designer of a service and/or a resource will be able to write a set of rules, that will be checked for their compliance during 5 stages of the service's lifecycle. The stages will be: 1) the pre-provisioning stage, 2) the creation stage, 3) the after-activation stage, 4) the supervision of the operation stage, and 5) the after-deactivation stage. These rules will not be available to be designed by any user but only by authorised users through the role-based access control (RBAC) module so that the need for rules evaluation will be minimised.

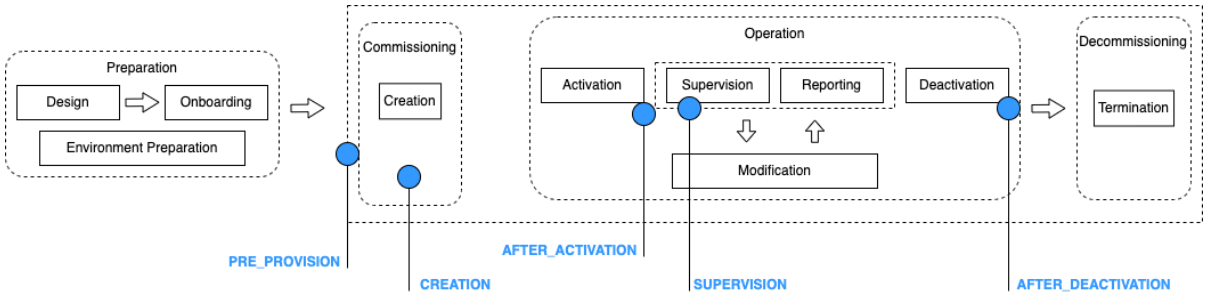

Figure 3-10: RM-L policy engine overview.

3.3.2.1.4 Scheduler

We can view the compute infrastructure of INCODE as hierarchy of Kubernetes clusters across the continuum, somewhat similar as the architecture used in the hypershift open-source project [43] where there is a management cluster that hosts the k8s control plane for other clusters; these managed clusters, have only k8s worker nodes and can be potentially anywhere as shown in Figure 3-11.

Co-funded by the European Union

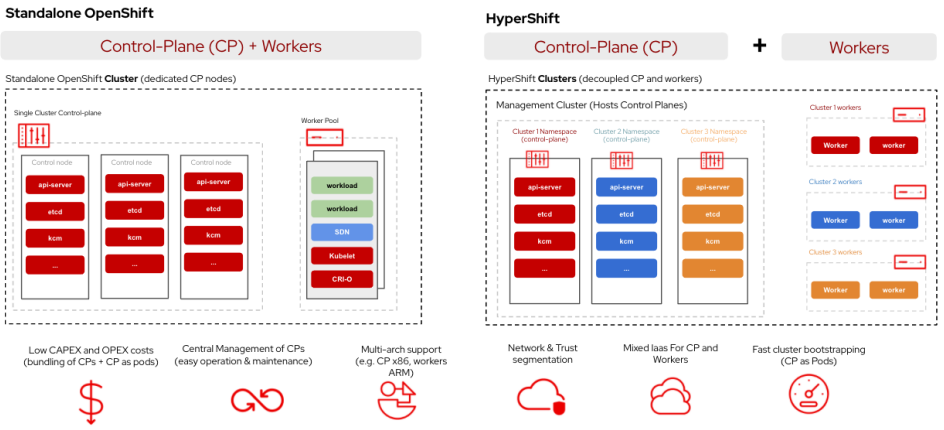

Figure 3-11: Hypershift cluster middleware for hosting multiple OpenShift control planes.

Within a single managed cluster, the standard Kubernetes scheduler can select where to execute a particular task based on the resources required by the task (CPU, memory), as well by special needs of the task (e.g., SRIOV or GPU). In those cases where the built-in k8s scheduler does not satisfy the INCODE needs, then a special purpose scheduler can be easily added to k8s [44].

Where INCODE differs from Hypershift is that in INCODE a single workload can spread across several managed clusters. However, given that a management cluster is also k8s, then we can use the same mechanism described above to develop and add a meta-scheduler that will break the INCODE workload into components and match the needs of each component to the capabilities available in each of the remotely managed clusters. This INCODE meta-scheduler will follow the same principles as other k8s schedulers, that is match required capacity (as expressed in the limits section of the component deployment descriptor) to available capacity in each of the potential target managed clusters; and it will use, as in standard k8s, labels and annotations to express special requirements.

3.3.2.1.5 Telemetry

Within the RM-L, it is mainly the RO that interacts with the telemetry service to verify resource usage in order to adequately scale out onto similar resources or to circumvent complete physical site or network failures. It is envisioned that the telemetry service will consist of three main parts. First, a metadata service which will serve as an index of that metrics and logs are collected, to be used by the telemetry UI service for dashboard authoring, or by an analytics service to automatically filter out specific signals belonging to an area on interest in order to discover a systemic issue. Second, the actual telemetry sink service, which will scale out alongside the INCODE resource layer. Third, the telemetry analytics which will allow to correlate signals and to perform stream-based machine learning in order to discover complex undesired or suboptimal conditions. The telemetry sink service will not allow telemetry from resources not annotated as observable in a policy within the policy engine or which are not present in the service catalog.

With regards to interfaces, the telemetry service will accept telemetry signals via its telemetry data API, will accept telemetry register requests from the service manager leading to a corresponding entry within the telemetry metadata registry and will allow analytic flows to be created based via the telemetry UI.

Co-funded by the European Union

3.3.2.2 Controllers and Drivers

INCODE's set of infrastructure controllers and drivers spans across 5 areas, namely IoT, RAN, network fabric, Cloud, and mobile core. The controllers and drivers in each of these five areas are presented in this section.

3.3.2.2.1 IoT Controller and Drivers

The IoT Controller (IoT-C) is a decentralized storage system responsible for maintaining the context information state among different elements or components present in every domain. Hence, this component allows the provision and consume of context information. Additionally, the IoT-C allows the subscription to context information to receive any notification about the change on the data.

Context information is compounded by a series of attributes associated with the entities to present the status and behaviour of the real world. This concept is also referred to as "digital twins". Additionally, IoT-C has the capacity to exchange this contextual information based on the implementation of the ETSI NGSI-LD API. To understand this API, it is needed to define some of its principles. In NGSI-LD, the world consists of a set of entities, which are composed of:

- **an entity identifier** that uniquely identifies an entity,
- **an entity type**, a description of which kind of information is provided by an entity and associated to the corresponding defined data model to describe this information,
- **properties** which represent the information expected to be found in these entities. These properties have values and other sub-attributes, pieces of metadata that describe an attribute. The definitions of these properties are made in the corresponding data model.
- **relationships** which represent the relations between different entities.
- **values** that each property has defined.

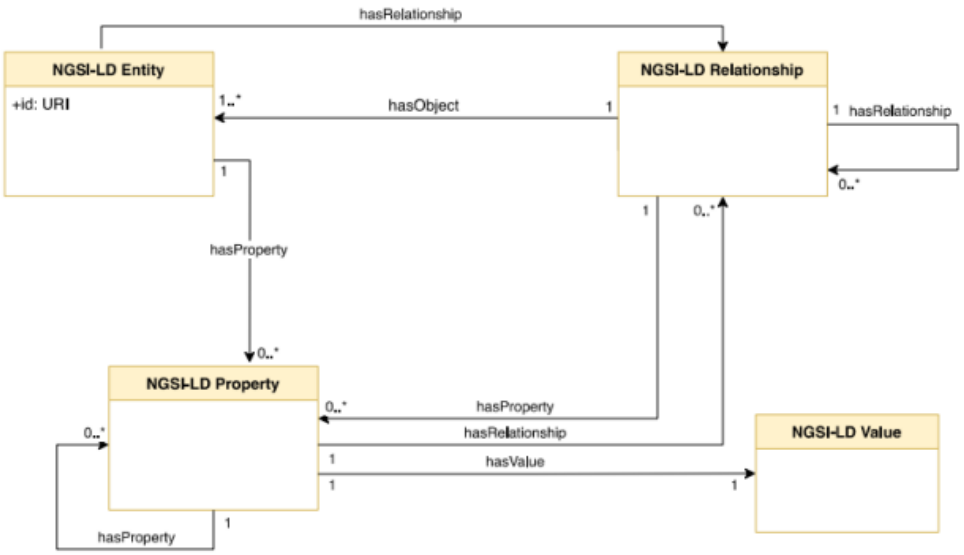

Figure 3-12: NGSI-LD information model as UML.

Hence, a wide range of objects can be represented as entities, such as drones, robots, RAN-C, etc. depending on the availability of the corresponding data model to describe that entity. ETSI NGSI-LD provides a standardisation approach to facilitate the modelling of these entities following the JSON-LD data format representation and JSON-Schema for describing the structure of this JSON data.

Besides, JSON-LD @context is used to expand the different properties defined for an entity. This @context includes a URI or set of URIs where we can find the correct semantic definition of the properties as well the expected type of value of them. Additionally, a description of the property is provided as well as the URL to the model definition. Therefore, the semantic definition of the data is separate from the entity properties facilitating the interoperability and management of the data.

Figure 3-13: Example of JSON-Schema definition of a property

Additionally, IoT-C allows the temporal representation of entities using sub-properties createdAt, modifiedAt, deletedAt and/or observedAt. ETSI NGSI-LD defines a set of operations for provision or management of temporal evolution of entities and attributes as well as the corresponding operations for consuming them. IoT-C use a historical database (TimescaleDB) to save and request this historical information of the entities. The temporal operations data consume (GET operations) are provided by Mintaka and the temporal operations data provision (POST, PATCH, DELETE) are provided by other Orion-LD to separate the reads and writes operations over the temporal data.

Co-funded by
the European Union

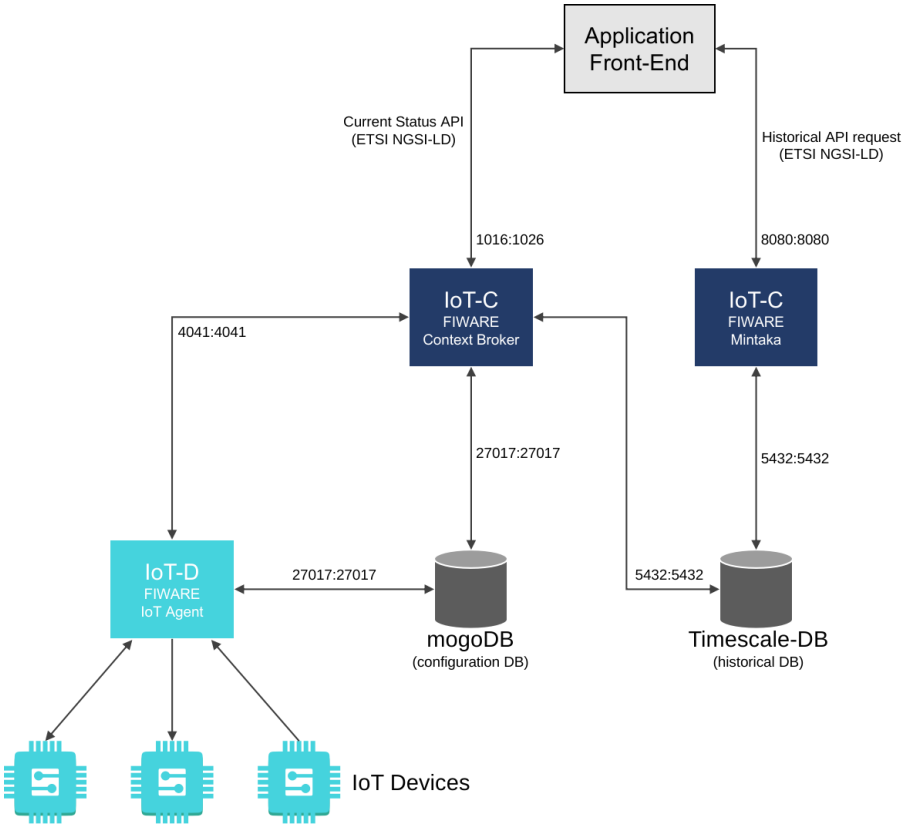

Figure 3-14: IoT Controller and IoT Drivers of the INCODE architecture.

Moreover, one of the main keys of the IoT-C is the capacity to access the state of the continuum in a decentralized way. This means that, despite the status of the ecosystem, the SM-L and RM-L will monitor the status of the continuum as well the readjustment of the configuration in the Edge domain. This concept of distributed state concept can be described as a decentralized storage system responsible to keep and maintain the state information between different elements of the INCODE architecture. IoT-Cs have the capacity to synchronise this contextual information based on defined mechanisms in the ETSI NGSI-LD between them (Distributed Operations). Therefore, IoT-C will store a number of entities with their attributes living in a distributed network.

The implementation of the Distributed Operations, called Context Source Registration operations in the ETSI NGSI-LD API, is something on going in the implementation of the IoT-C instance based on Orion-LD.

A Context Source Registration (CSR) is an operation to inform an IoT-C (NGSI-LD Broker) where it can be found non-local entities. Therefore, a local query to an IoT-C is not only retrieving the local entities but also the remote entities through a distributed request to the IoT-Cs behind the matching CSR and their entities are also appended to the final response. Consequently, IoT-C can query any of the IoT-Cs in the distributed network and get the same response. For this to work, every IoT-C needs to have (at least) one CSR for each and every other IoT-C connected to this distributed network.

Co-funded by
the European Union

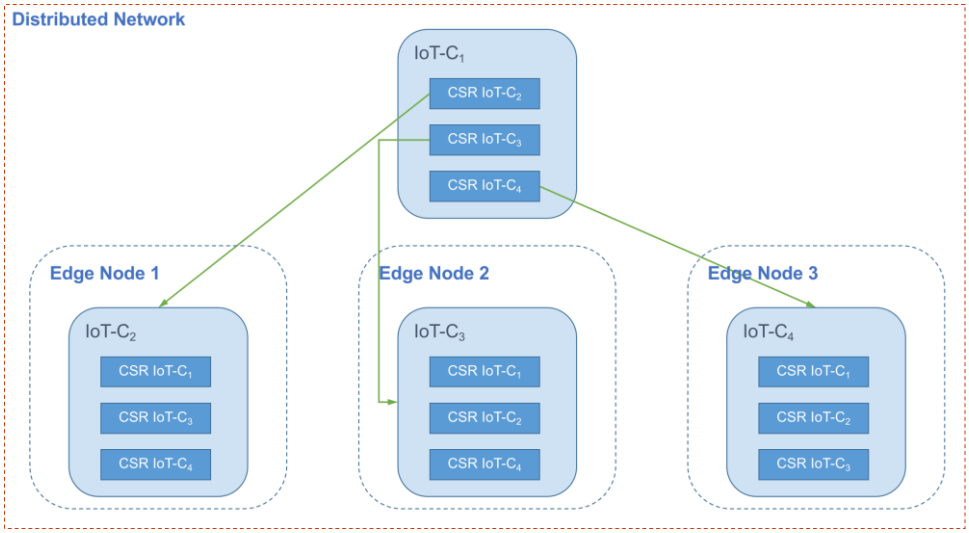

Figure 3-15: Distributed operations of IoT Controllers.

The ETSI NGSI-LD defines different ways to configure the CSRs for this distributed state, but we are interested in the following two approaches:

- The first approach consists of every IoT-C replicates the entire state. This is obtained using Inclusive CSRs. This offers the capacity of faster queries and resolve any possible issue in the communication of the edge nodes.
- The second approach consists of each IoT-C taking only exclusively their own data. Therefore, during queries, each IoT-C should forward requests to the other brokers. This would be done using Exclusive CSRs. The advantage of this approach is that the updates operation is executed faster but queries would be slower.

Finally, IoT-C offers a simple HTTP REST API which provides common functionalities to access, provision, and subscription operations over entities. The next table summarise all these operations that will be exposed by the RM-L and can be used to enable the use of the IoT-C with other parts of the INCODE platform.

Table 3-6: List of IoT-C operations exposed by the INCODE RM-L.

Endpoint	Title	Description
/ngsi-ld/v1/entities	Context Information Provision	Operations for providing or managing Entities and Attributes
/ngsi-ld/v1/entities /ngsi-ld/v1/entityOperations	Context Information Consumption	Operations for consuming Entities and checking for which Entity Types and Attributes Entities are available in the system
/ngsi-ld/v1/subscriptions	Context Information Subscription	Operations for subscribing to Entities, receiving notifications and managing subscriptions
/ngsi-ld/v1/temporal/entities	Temporal Context Information Provision	Operations for providing or managing the Temporal Evolution of Entities and Attributes

Co-funded by the European Union

/ngsi-ld/v1/temporal/entities /ngsi-ld/v1/temporal/entityOperations	Temporal Context Information Consumption	Operations for consuming the Temporal Evolution of Entities
/ngsi-ld/v1/csourceRegistrations	Context Source Registration	Operations for registering Context Sources and managing Context Source Registrations (CSRs)
/ngsi-ld/v1/csourceRegistrations	Context Source Discovery	Operations for retrieving and discovering CSRs

Regarding the IoT Drivers, RM-L involves the need to resolve a common set of issues in the IoT world, the difference data representation formats and the heterogeneous protocol communication. Here is where the IoT Driver (IoT-D) provides a solution on this issue. An IoT-D is a component that lets a group of devices sends their context information to and be managed from a IoT-C using their own native protocols. IoT-D should also be able to deal with security aspects, authentication and authorization processes, to stablish secure communication between the IoT Devices and Robots and the IoT-C. This security mechanism is developed based on standardized OAuth2 token management in the header.

The IoT-D components serve as a practical implementation of the Interfaces with IoT Devices, Robots, and third systems with the purpose to facilitate the retrieval of valuable contextual information (Entities) and enabling the triggering of actions in response to updates in the fore, the main purpose of the IoT-D is to make a gateway translation from the payload and transport protocol towards JSON-LD format compliant with ETSI NGSI-LD API. This allows the effective query or subscription to changes that can be produced in the real world, which is particularly useful for monitoring of the status of various parameters presented in NGSI-LD entities attributes at the IoT-C level.

Consequently, IoT-Ds act as intermediaries to handle the complexity and heterogeneity of data integration and protocol transformation, ensuring that the contextual information, from diverse sources, is represented and manage in a standardized way and made available through the RM-L using the ETSI NGSI-LD.

The design of the IoT-Ds has been developed to facilitate the future creation of any new component with different protocol and/or data format representation. This is achieved through the encapsulation of all ETSI NGSI-LD API operations into a specific library, the IoT-D node library. The main purpose of this IoT-D node lib is to provide a common framework for provisioning IoT-D, allowing each individual IoT-D to access standardized mapping data for devices and to offer series of common utility functions towards the Northbound communications, usually the connectivity with the IoT-C. Therefore, each IoT-D is just the implementation of the Ad-hoc code to deal with the proprietary communication protocol and data format representation of the Devices and Robots.

Co-funded by the European Union

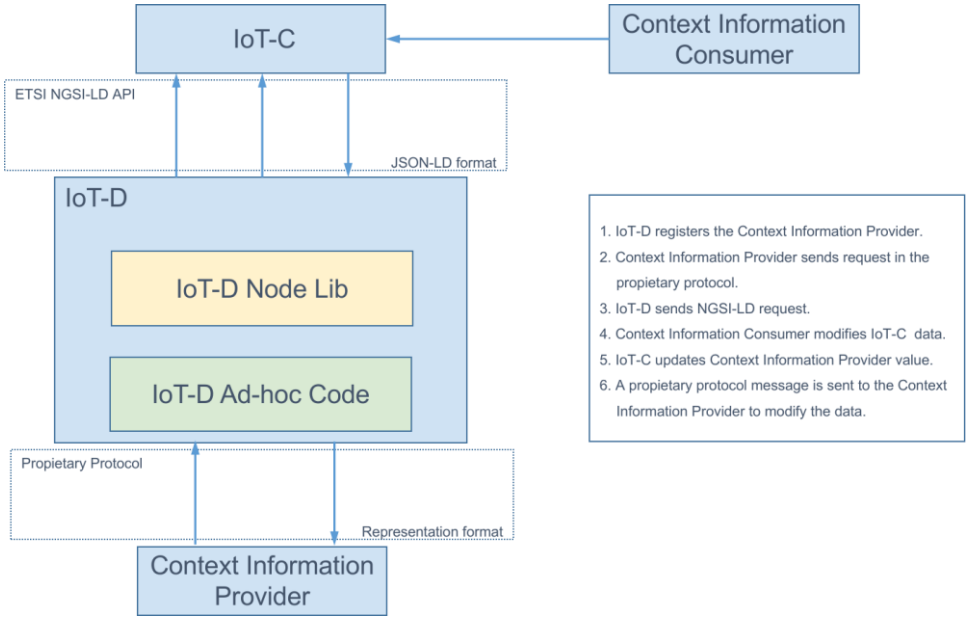

Figure 3-16: Architecture of and IoT-D.

As we can see in the figure, the IoT-D are divided into northbound and southbound communications.

- **For southbound communications**, the IoT-D Ad-hoc Code listens to any change in context information entities and raises specific callbacks towards the IoT-D Node Lib in order to process the information obtained from Devices and Robots and process it.
- **For northbound communications**, the IoT-D Node Lib offers an interface which accepts structured input data, following the corresponding Data Model defined for the captured Context Information Entities and send the data to the corresponding IoT-C.

Currently, the IoT-D manage JSON, UL, and XML data representation format and HTTP, MQTT, LWM2M, LoRaWAN, SigFox, and OPC-UA communication protocols under the hood offering a facade pattern to handle this complexity.

Additionally, IoT-D offers a simple HTTP REST API which provides common functionalities to access, provision and shut down context information providers as well as configuration of a group of devices with the same properties. The next table lists all these operations that will be exposed by the RM-L and can be used to enable the use of the IoT-D with other parts of the INCODE platform.

Table 3-7: List of IoT-D provision operations exposed by the INCODE RM-L.

Endpoint	Title	Description
/iot/about	IoT-D Service Health	Check the IoT-D Service Health status with information about IoT-D Node Lib version, port used, base root and version of the IoT-D.
/iot/services	Service Group Management	Provide the operations to manage the group management of IoT-Ds. A group management is a set of IoT-Ds that share the same properties (apikey, IoT-C url, Entity type, resource used in the base url, and the list of attributes).
/iot/devices	Device or actuator management	Provides the operations related to the management of a specific device or actuator.

Co-funded by the European Union

At this stage of the project, the use of the IoT-D with JSON payload format and MQTT, HTTP, ROS2, and Modbus transport protocol support seems relevant for the INCODE project. The activity to be developed in the project is the provision and support of the corresponding IoT-Ds to be adopted for each of the Application Areas. The first two are already available to use, the IoT-D Ad-hoc Code for ROS2 and Modbus need to be developed under the scope of the INCODE project.

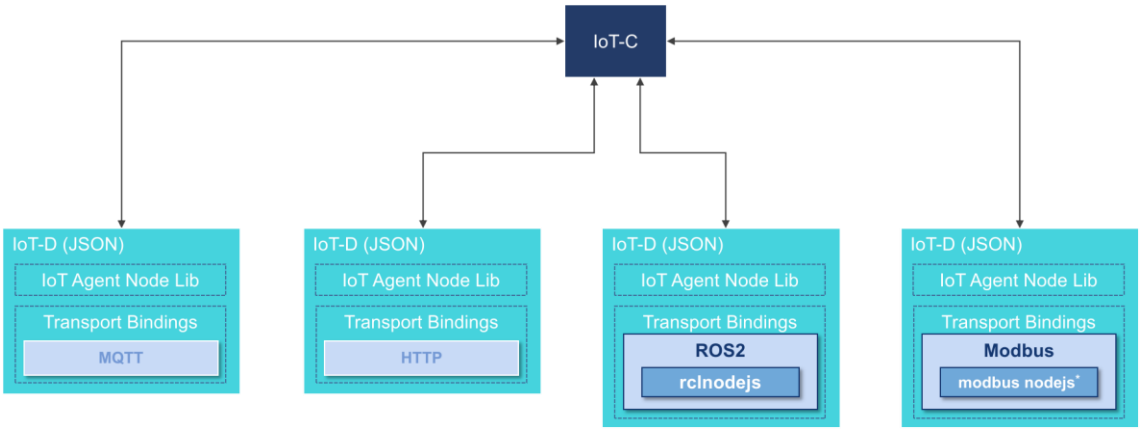

Figure 3-17: Composition of the different IoT Drivers.

3.3.2.2.2 RAN Controller and Drivers

The Radio Access Network Controller (RAN-C) configures and manages the 5G RAN equipment based on O-RAN Alliance standards [5]. It manages multi-vendor equipment as white boxes. RAN-C inherits Radio Resource Management (RRM) functionalities from two separate controllers:

- The near-Real-Time RAN Intelligent Controller (near-RT-RIC) handles lifecycle management for all network elements, configuration, and other essential functions. It optimizes RAN functions with policy-based guidance, model management, and enrichment information to near-RT-RIC function.
- The non-Real-Time RAN Intelligent Controller (non-RT-RIC) optimizes and controls O-CU and O-DU nodes. It provides intelligent functionalities like dynamic spectrum sharing, traffic steering, load balancing, and handover provisioning.

RAN-C uses a southbound E2 interface with white-box radio equipment and exposes the northbound APIs A1 and O1 to the RM-L platform. A1 comes from non-RT-RIC components, while O1 comes from near-RT-RIC.

Co-funded by
the European Union

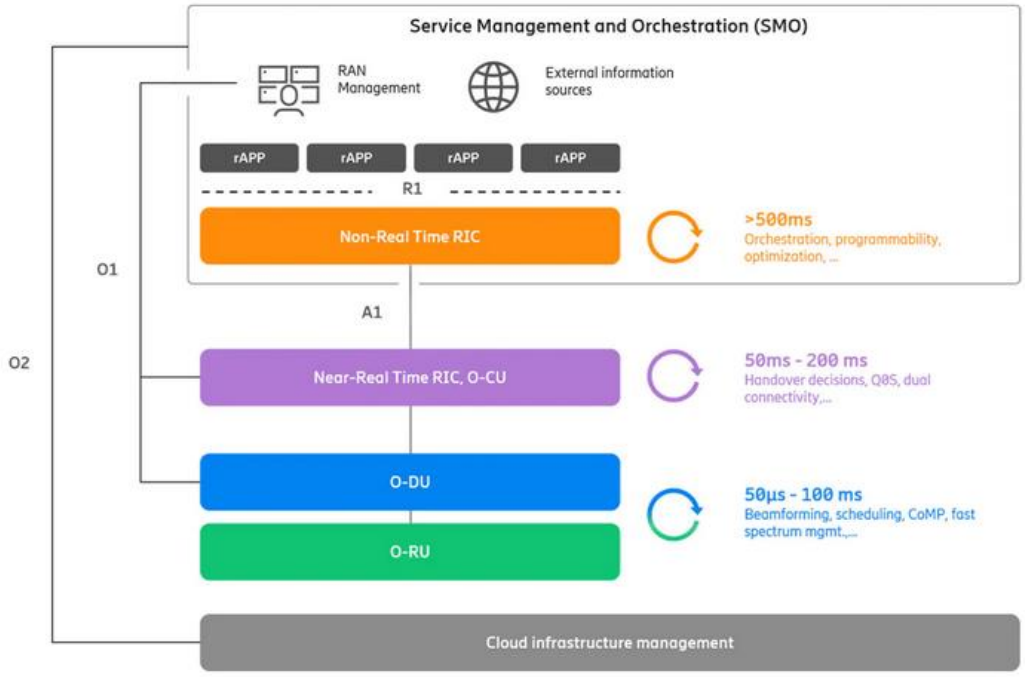

Figure 3-18: RAN Controllers and interfaces as dictated by O-RAN Alliance.

Regarding the RAN drivers (RAN-D), they implement deployment, management, and configuration actions (see Table 3-8) of the controllers over software-based O-CUs, O-DUs, and O-RUs on commodity hardware. rApps and xApps are part of the RAN-D mechanism.

Table 3-8: List of RAN-D operations exposed by the INCODE RM-L.

Endpoint	Title	Description
A1	A1 - Interface	This interface interconnects Non-RT-RIC function in the Service Management & Orchestration (SMO) layer with a new logical near-RT-RIC function in the RAN. This new A1 interface allows the Non-RT-RIC to provide Policy Guidance to the RAN (near-RT-RIC) to steer its operation.
O1	O1 - Interface	Interface between management entities in Service Management and Orchestration Framework and O-RAN managed elements, for operation and management.
E2	E2 - Interface	This interface interconnects the near-RT RIC and the E2 nodes, i.e., DUs, CUs in 5G and O-RAN compliant eNBs in 4G. E2 allows the RIC to control procedures and functionalities of the E2 nodes. E2 also enables the collection of metrics from the RAN to the near-RT RIC, either periodically or after pre-defined trigger events.

3.3.2.2.3 Network Fabric Controller and Drivers

INCODE promotes a fully programmable network fabric of whiteboxes using state-of-the-art SDN protocols, such as the Programming Protocol-independent Packet Processors (P4) [8]. To do so, INCODE will rely on an open standardized solution for managing heterogeneous -

Co-funded by the European Union

yet programmable - network fabric. Specifically, INCODE will leverage the TeraFlowSDN controller [10]; an ETSI Open Source Group that develops an open source cloud native SDN controller enabling smart connectivity services for future networks beyond 5G.

Among many other SDN drivers, the TeraFlowSDN controller implements a novel SDN device driver with native P4 support. A P4Runtime client has been natively incorporated in the TeraFlowSDN device driver and an overlay P4 Manager module abstracts low-level client interactions with the whiteboxes to offer key abstractions for key P4 entities (e.g., tables, actions, etc.). Figure 3-19 shows a high-level overview of how INCODE plans to integrate the ETSI TeraFlowSDN controller and P4 device driver into the INCODE architecture and how it interacts with P4 devices through P4Runtime and ONF's Stratum [11][12].

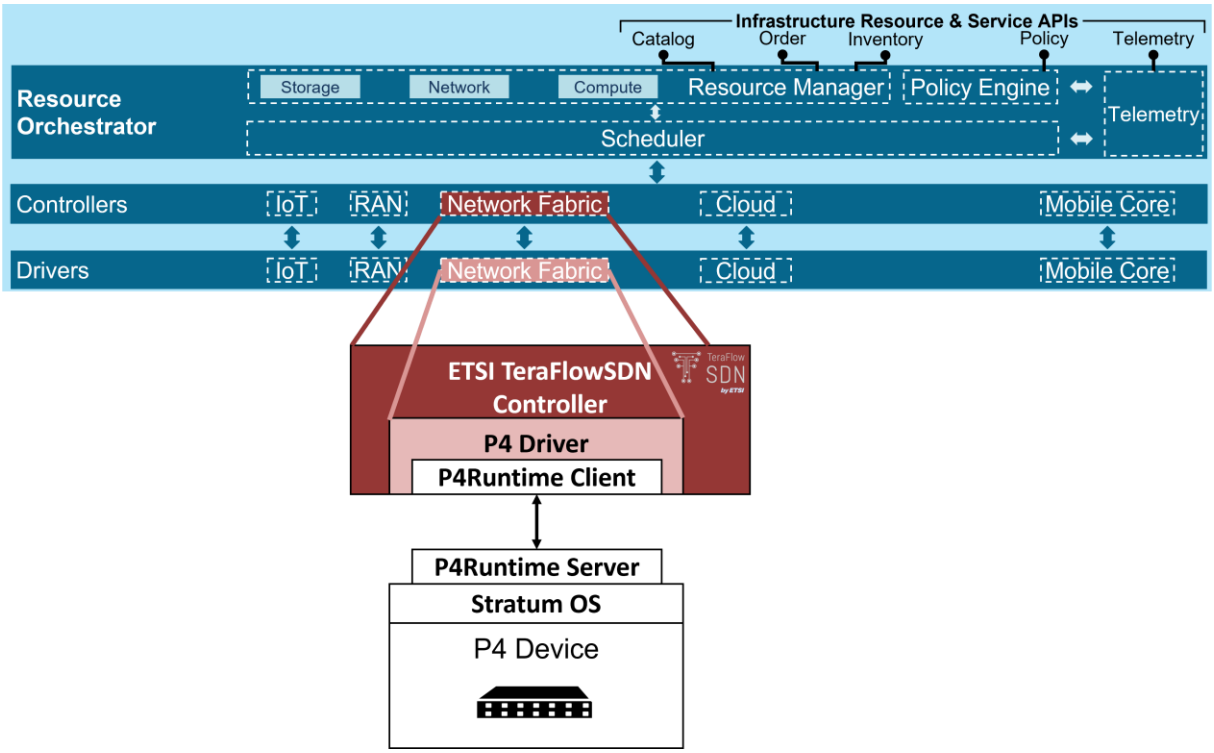

Figure 3-19: Architecture of INCODE's network fabric controller and driver.

The ETSI TeraFlowSDN controller and P4 driver is a cloud native platform that could be easily integrated into the INCODE ecosystem, which is also based on Kubernetes.

API

The complete list of RPCs supported by the ETSI TeraFlowSDN network fabric driver is depicted in Table 3-9. These RPCs are currently available as part of the ETSI TeraFlowSDN release v2.1[1] (July 2023). INCODE plans to use this version as a baseline; potential extensions on this baseline release or open-source contributions to future releases for the needs of INCODE will be considered in the context of WP3 T3.2.

Table 3-9: List of RPCs supported by the INCODE P4 network fabric device driver.

RPC Name	RPC Description
Connect	Initiates a connection with a given P4 device on a given IP. The Connect RPC also offers custom settings which allow to pass the P4 binary file and the P4 info file to be installed on a given P4 device to realize a target forwarding logic.
Disconnect	Tears a connection to a P4 device down.

[1] *ETSI TeraFlowSDN controller release v2.1 (July 2023), Available: https://labs.etsi.org/rep/tfs/controller/-/tags*

Co-funded by the European Union

GetConfig	Retrieves either the entire device configuration or part of it, based on an input list of resource keys to be looked up.
GetInitialConfig	Retrieves a basic initial configuration for a P4 device to allow automated onboarding of P4 devices through the Automation component.
SetConfig	Installs a set of entries to the table(s) of the P4 pipeline, based on an input map of resource keys to resource values.
DeleteConfig	Removes a set of entries from the table(s) of the P4 pipeline based on an input list of resource keys to be removed.

3.3.2.2.4 Mobile Core Controller and Drivers

The mobile core controller (Mob-C) comprises the total of functionalities implemented by 5G Core (5GC). This controller is a software-based entity that runs on general purpose hardware, typically on datacenters and cloud environments, allowing for scalability and flexibility of deployment. Part of the controller concerning the user plane, could be deployed at the edge to facilitate local breakout to the remote user equipment (UE).
Mob-C is implemented as a set of network functions that follow the service-based architecture (SBA), shown in Figure 3-20, and they are responsible for clearly defined and specific tasks, such as session management (Session Management Function - SMF), mobility management (Access and Mobility Management Function – AMF), data traffic management (User Plane Function – UPF), authentication (Authentication Server Function - AUSF), and policy control (Policy Control Function - PCF).

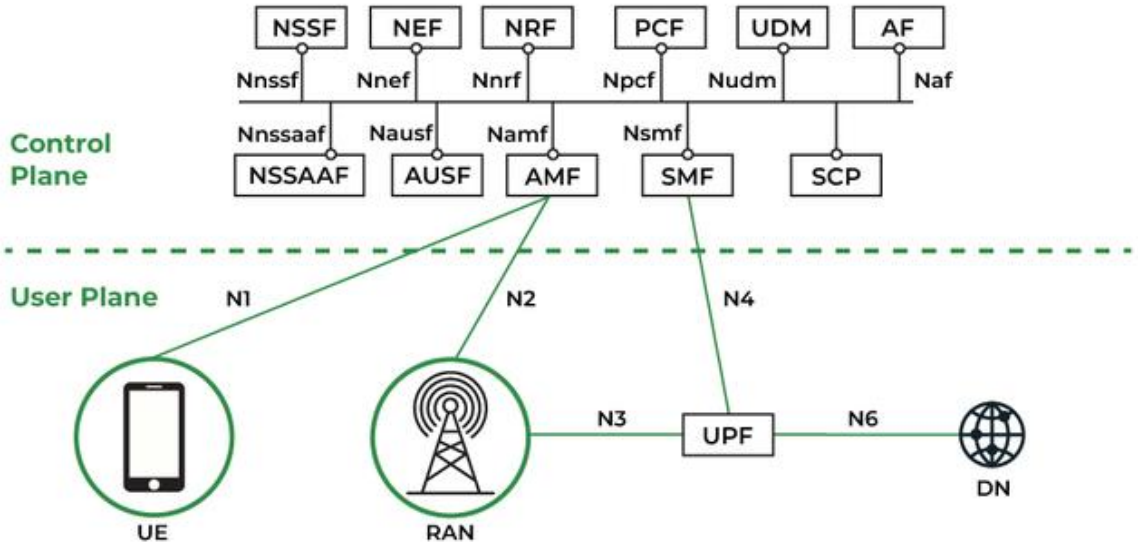

Figure 3-20: Service Based Architecture of 5G Core.

Regarding the Mobile core drivers (Mob-D), they concern the software components that facilitate the deployment of Mob-C, as well as its communication with the underlying hardware and the overlying orchestrators. In the scope of INCODE, it is aimed that Mob-D will be able to deploy Mob-C as a service (5GaaS) and implement the necessary connectivity with the respective components ad-hoc. Finally, Mob-D is responsible for the deployment of performance-specific UPFs, according to user profiles. Table 3-10 showcases the APIs of the 5G Core with the underlying infrastructure.

Co-funded by the European Union

Table 3-10: List of APIs supported by the INCODE Mobile Core controller.

Endpoint	Title	Description
N1	AMF-UE	Reference point between the UE and the AMF.
N2	AMF-RAN	Reference point between the RAN and the AMF.
N3	Device or actuator management	Reference point between the RAN and the UPF.
N6	PF-DN	Reference point between the UPF and a Data Network.

3.3.2.2.5 Cloud Controller and Drivers

The Cloud Controller (Cloud-C) is basically a Kubernetes controller with extensions where necessary. The reasons behind choosing Kubernetes are mainly:

1. Kubernetes already provides a developer-friendly abstraction of underlying compute resources as required by INCODE; and,
2. Kubernetes was designed from the beginning with extensibility in mind.

For any device not natively supported by Kubernetes, we will develop the appropriate Cloud driver (Cloud-D) as a Kubernetes Operator [45], that is a custom resource that extends the Kubernetes API to support any application or devices. For example, if a particular INCODE workload cannot be containerized and requires a virtual machine, we can use the kubevirt [46] to treat (deploy, scale, upgrade, uninstall, etc.) virtual machines as any other Kubernetes resources, and manage them through the Kubernetes API (or the standalone kubectl CLI tool). Similarly, we can extend the API (and kubectl) with any device needed to be controlled by the Cloud-C.

API

INCODE will leverage the Kubernetes API server - the core of Kubernetes' control plane - to perform cloud service orchestration and cloud resource allocation respectively. The Kubernetes API server allows the manipulation of the state of various Kubernetes API objects, such as namespaces, pods, secrets, deployments, services, etc. The Kubernetes API server exposes a REST API that lets overlay orchestration components (i.e., the RO and the SO) and end users (e.g., through kubectl) to communicate with a Kubernetes Cluster [22].

The open API specification of the Kubernetes controller is available at [50]. Table 3-11 summarizes an indicative list of interfaces that INCODE plans to leverage to manage relevant Kubernetes objects. On these interfaces, the INCODE orchestrators can apply Create, Read, Update, and Delete (CRUD) operations; we use the term “Manage” to refer to these operations altogether. These interfaces appear in Kubernetes version 1.26. By the time of the INCODE deployment and validation activities, more recent Kubernetes versions may be exploited, where these APIs may potentially change (and likely be augmented).

Co-funded by the European Union

Table 3-11: List of interfaces supported by the INCODE Cloud controller based on Kubernetes version 1.26.

Endpoint	Description
/api/v1/namespaces	Manage a namespace. This is the environment where a Kubernetes application “lives”.
/api/v1/namespaces/{namespace}/resourcequotas	Manage a ResourceQuota object in a target namespace. The ResourceQuota object defines the constraints of the target namespace in terms of usable resources (CPU, RAM, Storage).
/api/v1/namespaces/{namespace}/serviceaccounts	Manage a service account object in a target namespace.
/api/v1/namespaces/{namespace}/secrets	Manage secret objects in a namespace.
/apis/rbac.authorization.k8s.io/v1/clusterroles	Manage a ClusterRole object in a Kubernetes cluster to define a set of permissions.
/apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/rolebindings	Manage a RoleBinding object in a target namespace to associate a ServiceAccount to a ClusterRole (set of permissions).
/api/v1/pods	Manage the smallest, most basic deployable objects in k8s. This is the API that the SM-L uses to prepare the environment for each application container image.
/api/v1/deployments	Guide k8s on how to create or modify instances of the pods that comprise a containerized application deployment.
/api/v1/services	Expose an application running on a set of Pods. The VAO exploits both basic application exposure via ClusterIP, as well as more advanced schemes, such as NodePort and LoadBalancer.

3.3.3 Logical Components in the Vertical Security and Trust Layer

This section describes the ST-L, a vertical layer offering security and trust services to both SM-L and RM-L, as shown in Figure 3-5. The ST-L comprises four logical components, which we describe in detail below. Specifically, we introduce Identity Management (IM) in Section 3.3.3.1, Attestation (AT) in Section 3.3.3.2, Data Provenance (DP) in Section 3.3.3.3, and Attack Mitigation (AM) in Section 3.3.3.4.

3.3.3.1 Identity Management

The Identity Management (IM) component manages identities of devices at the RM-L and of application/services at the SM-L. This is done via a Decentralized Identity and Access

Co-funded by the European Union

Management component of the INCODE Platform. This section will dive into the various regulations and compliances needed to be considered for the Identity and access management.

1. **EIDAS Regulation** - The Regulation on electronic identification and trust services for electronic transactions in the internal market (eIDAS Regulation) for electronic transactions internal market (eIDAS Regulation) states that in order to contribute to their general cross-border use, it should be possible to use trust services as evidence in legal proceedings in all Member States. INCODE project should be fully aligned with the objectives of the eIDAS regulation, specifically Article 17 of the eIDAS Regulation, which says that Member States should encourage the private sector to use electronic identification voluntarily under a notified scheme for identification purposes when needed for online services or electronic transactions. The possibility to use such electronic identification enables the private sector to rely on electronic identification and authentication already largely used in many Member States at least for public services and to make it easier for businesses and citizens to access their online services across borders.
2. Usage of Verifiable Credentials signed with eIDAS digital certificates issued by EU Trust Service Providers (TSPs) and an associated Identity and Access Management Framework using Verifiable Credentials that is compatible with the Gaia-X Trust Framework and, therefore also with the EBSI initiative and aligned with the recently published European Digital Identity Wallet Architecture and Reference Framework.

General Overview of Components and Interaction in Decentralized IM

Considering the regulation, all the IMs participating in the platform should be certified by a Universal Trust Registry. This would also enable the possibility of using Self-issued OpenID Providers (SSOP). The self-issued credentials (or Verified Credentials) issues by the SSOP should be signed by the certificate issued by the Universal Trust Registry issued to the SSOP. Figure 3-21 shows how all the components are working in Decentralized IM.

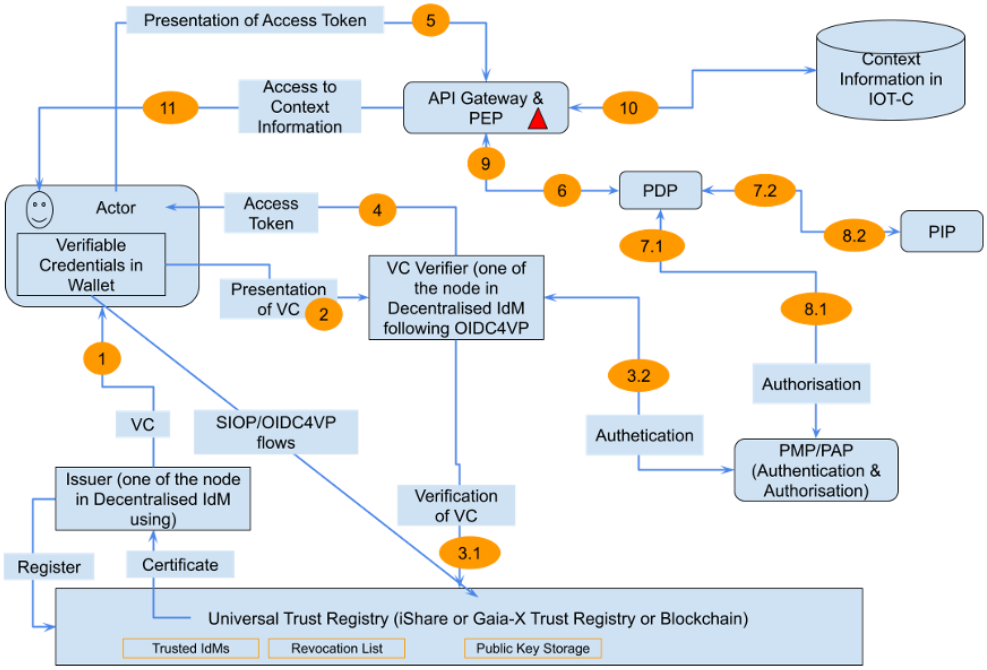

Figure 3-21: Decentralized Identity Management overview.

Co-funded by the European Union

Universal Trust Registry

The Universal Trust Registry (UTR) defines and enforces a set of rules in a way that all services and organizations can use their digital identities and attributes in a consistent and trusted manner. This makes it easier for organizations and users to complete interactions and transactions or share information with other participants. The UTR addresses the following issues.

ID Binding

At the root of any decentralized IM there is the requirement to verify the identity of an entity in the real world and the assignment of some identifier that can be used later in representation of the real entity in the online processes. This association between an identifier (including some metadata) and the real identity of an entity is what we call ID Binding. We propose to rely on identifiers already used in digital certificates issued by the Trust Service Providers (TSPs) authorized by the relevant European laws. The combination of digital certificates issued by TSPs, and Verifiable Credentials contributes to the legal validity and interoperability of the cross-border data-related transactions in the European Union facilitating the cross-border validation of eSignatures, eSeals, and more. Essentially, Verifiable Credentials and Presentations (including Product Specifications and Offerings) used in the ecosystem will be signed using digital certificates using the JAdES format as defined in ETSI TS 119 182-1 [47]

In addition, we propose the use of a specialized type of Verifiable Credential that we will call VerifiableID, in line with the terminology used by EBSI. A Verifiable ID is a special form of a Verifiable Credential that a Natural Person or Legal Entity can put forward as evidence of who he/she/it is, and that can be used for identification and authentication purposes as described later in this document. Identifiers are required for IoT devices, Gateways and IoT Hubs respectively in the application context. Such components require identification on a similar basis. From an organizational perspective the application context must be linked with legal and/or natural persons identifiers to clarify the delegation of power to the application context. Such identifiers could be realized as X.509 certificates or as Digital Identifiers (DID).

A valid identifier should contain at least:

1. Issuer distinguished name
2. Subject distinguished name
3. Serial number
4. Version information
5. Validity information

Proof of participation in decentralized IDM group

We propose to use a Trusted Participant List including the identities and associated metadata of all legal persons/IoT Devices participating in the concrete ecosystem. The Trusted Participant List is updated during the onboarding process of an entity and is managed by one or more collaborating trusted participants in the INCODE platform. Please note that this list is different from the EU Trusted List with the identities of TSPs authorized to issue digital certificates/seals in the EU.

Co-funded by the European Union

There are different ways to implement the Trusted Participant List. No matter which way we choose, the users of the Trusted Participant List should not be aware of the technology used to implement it. The users of the Trusted Participant List just use an API to query the list on verification, and the maintainers use a different API to register and update the list.

This way, it is completely possible to use a mix of centralized and decentralized technology without the users noticing it. Or to migrate transparently from one technology to another depending on the requirements of the specific ecosystem.

Having said that, we propose that one of the implementations uses a federated set of interoperable EBSI-compatible blockchain networks for the maintenance of the Trusted Participant List, providing a decentralized, hyper-replicated, efficient and resilient mechanism for querying the list. Anyone can create a replica of the information using centralized systems if they wish.

We propose to base the API in the one defined by EBSI for Trusted Lists of different types. For example:

GET /participants and GET /participants/{did} to get the list of participants or to check a given participant if we have its DID, respectively.

There are several other APIs to get attributes/metadata associated with the participants, and APIs to maintain the list, used by the entity or entities responsible for the list. The full specification is described in the EBSI documentation.

Proof of Issuing Authority

Given that anyone can have access to the technology needed to create Verifiable Credentials and anybody can issue credentials and digitally sign them with their eIDAS digital certificate, the problem is how a verifier knows that the Verifiable Credentials received from the subject have been issued by an entity which is entitled or authorized to issue that type of credential.

The primary mechanism to solve this problem is the use of Trusted Issuer Lists (there may be several lists, one per domain or type of credential). A Trusted Issuers List is a register of trusted public entities which can issue Verifiable Credentials belonging to a given domain or of a given type. It is assumed that an entity must be first in the Trusted Participant List before it appears in the Trusted Issuers List. This list includes the identifiers, public keys for verification of signatures and their accreditations in the form of Verifiable Credentials/Presentations from third parties, enabling the entity to issue credentials of a given type. All information in the registry is validated and signed by trusted legal entities of the corresponding domain (Conformity Assessment Bodies and third-party auditors).

Using Trusted Issuers Lists (there may be several lists, one per domain or type of credential) is the simplest mechanism. However, in very complex ecosystems with many entities issuing credentials of different types, the management of Trusted Issuer Lists can be difficult to scale. For those ecosystems we can use the combination of Trusted Issuer Lists with the “chaining” of Verifiable Credentials, like the certificate chaining used with traditional X.509 digital certificates:

Co-funded by the European Union

1. At the root of the trust hierarchy there is a set of Trusted Issuers Lists as described above, containing the primary trusted entities in the ecosystem.
2. The entities in those Trusted Issuers Lists can issue special Verifiable Credentials to other entities, authorizing them to be also Trusted Issuers, even if they are not included in a Trusted Issuer List. The signature of the special Verifiable Credential attests that the subject of the credential is explicitly authorized by the signer to issue a given type of credentials (usually a subtype of the parent type, but not necessarily; the specific rules must be defined in the corresponding governance model for the domain/ecosystem). This mechanism can be also used by those Trusted Issuers not in Trusted Lists if we need several levels in the hierarchy, though usually two or three layers (including the root Trusted Issuers List) should be enough to handle large ecosystems.

Identification and Authorization

To enable transactions, the Decentralized Identity and Access Management Framework leverages on the above and provides an Identity Management system addressing additional requirements. For authentication we propose to use the same mechanism as in EBSI and the EUDI Wallet for online flows, namely using OpenID Connect for Verifiable Presentations (OID4VP) and Self-Issued OpenID Provider v2 (SIOPv2), which leverages the proven, robust and secure standards of OpenID Connect protocols to:

1. Transport Verifiable Credentials/Presentations in the flows of OpenID Connect so Relying Parties can use well known mechanisms to issue and receive Verifiable Credentials.
2. Enable all participants (via SIOPv2) to send identity data and Verifiable Credentials to other participants without the requirement for big and centralized Identity Providers as it is unfortunately common in implementations of standard OpenID Connect.

The authentication phase corresponds to interactions (1) to (4) in Figure 3-21. This way we implement a distributed, fault-tolerant, trustful and efficient IAM system avoiding the existence of centralized Identity Providers (IdPs). Using widely implemented standards like OIDC and W3C Verifiable Credentials provides a very low barrier of entry to participants implementing IAM.

Using OIDC for transporting Verifiable Credentials enables integration of the attested data inside the credential for sophisticated and flexible Authorization schemes. Participants implementing this Decentralized Identity and Access Management Framework can use credential data for advanced RBAC/ABAC access control and policy enforcement. The authorization phase corresponds to interactions (5) to (11) in Figure 3-21.

Furthermore, the IAM Framework can be used by participants not just to interact with the data space and marketplaces, but they can adopt it and use it for peer-to-peer interactions between participants in the ecosystem without the involvement of central entities.

Co-funded by the European Union

Usage and Access Control

A policy or contract negotiation (CN) involves two parties, a provider that offers one or more assets under a usage contract and a consumer that requests assets. A CN is uniquely identified through IRI. Each CN requires a newly generated IRI, which may not be used in a CN after a terminal state has been reached. A CN progresses through a series of states, which are tracked by the provider and consumer using messages. A CN transitions to a state in response to an acknowledged message from the counterparty. Both parties have the same state of the CN. In case the states differ, the CN is terminated, and a new CN must be initiated.

Components Involved

1. PDP - The **Policy Decision Point (PDP)** makes the decision based on the data sent by the PEP and the deposited policies. The PDP also interprets the policies in terms of context information and instructions. This means the policy decision may also depend on additional information that is not present in the intercepted system action itself. This includes information about the context, such as data flows or the geographical location of an entity. It is also possible to specify pre- or post-conditions that must be held before (e.g., integrity check of the environment) and after (e.g., data item is deleted after usage) decision-making. In addition, it is possible to define on-conditions that must be held during usage (e.g., only during business hours). The PDP functionality is summarized in Figure 3-22.

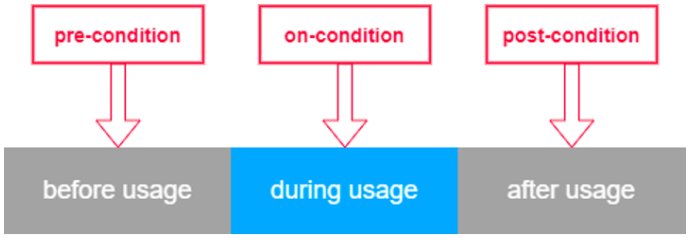

Figure 3-22: PDP Functionality with an example of access control during business hours

2. PEP has two main tasks as shown in Figure 3-23. First, it is the entry point for enforcement, meaning it is the point where data or metadata is stopped and transferred to the PDP, the PDP decides and returns it to the PEP. Secondly, the PEP will subsequently manipulate or lock the data according to the decision.
3. PAP is used to support the creation and specification of usage policies often via a user-friendly graphical interface. PAP is not a component that is directly needed for enforcement but should be briefly mentioned here. PAP is important for specification and management of usage policies.

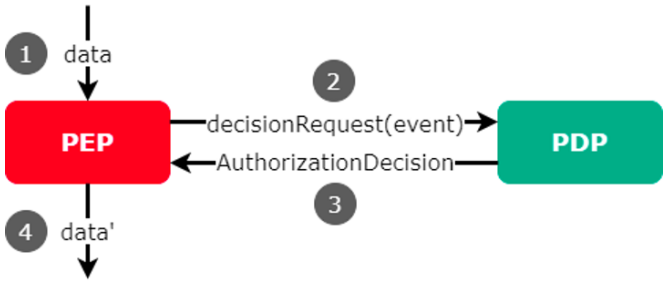

Figure 3-23: Simplified PEP and PDP interaction.

Co-funded by the European Union

4. PIP - **The Policy Information Point (PIP)** is the component to determine information such as context information during policy evaluation. This information can then be used in the PDP for decision making.

Policy Enforcement

Once the authentication phase has been performed based on Verifiable Credentials, the system generates an access token which can be used to access protected resources and allows efficient enforcement of usage policies. This phase corresponds to interactions (5) to (11) in Figure 3-21.

In the description we use concepts from the standard XACML architecture, and the most relevant ones are explained below, using a simplified diagram of the main logical components. To be clear, we are not limited to an XACML implementation, but use some of their architectural concepts for our explanation.

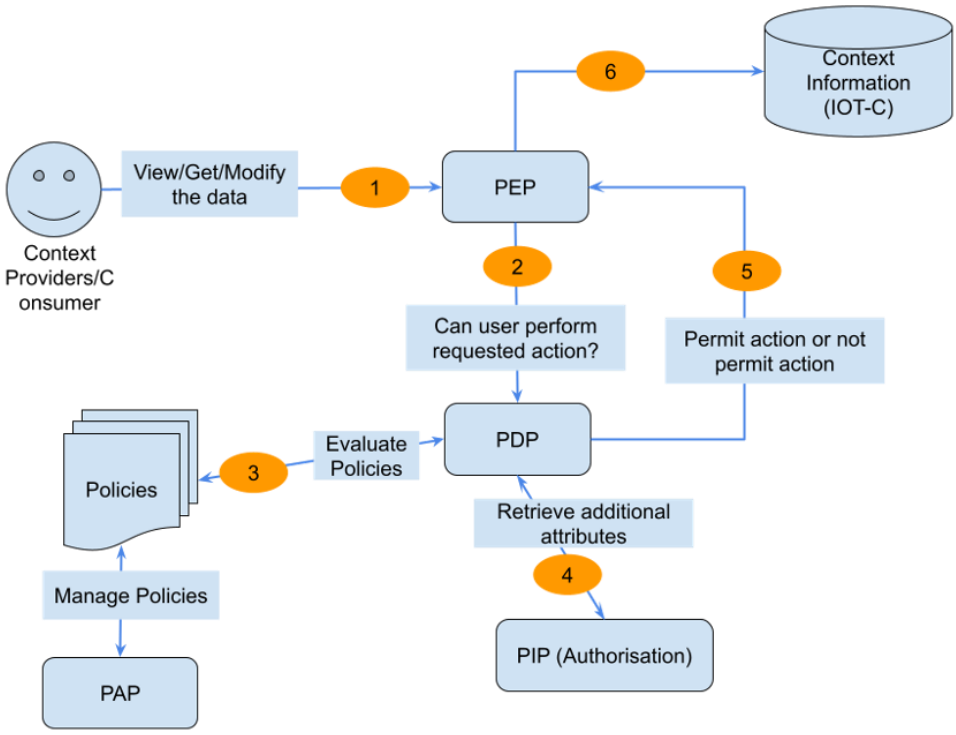

Figure 3-24: Authorisation Flow

We assume that the authentication phase has been already performed, and that an access token has been derived from the VerifiableID that was used for authentication. Briefly, we can assume that the access token contains the essential information from the credential (e.g., claims) so the policy enforcement can be performed.

The general flow is the following:

1. A Context Provider or Consumer (actor) sends a request which is intercepted by the Policy Enforcement Point (PEP). If the request is authenticated (contains the access token), the flow continues.
2. The PEP converts the request (including data from the access token) into an authorization request in a given format.

Co-funded by the European Union

3. The PEP forwards the authorization request to the Policy Decision Point (PDP).
4. The PDP evaluates the authorization request against the policies it is configured with. The policies are acquired via the Policy Retrieval Point (PRP) and managed by the Policy Administration Point (PAP). If needed it also retrieves attribute values from underlying Policy Information Points (PIP).
5. The PDP reaches a decision (Permit / Deny / Not Applicable / Indeterminate) and returns it to the PEP.
6. PEP allows access to context information based on the actor's access policies defined.

Generally, the PEP is combined with the API Gateway (or Proxy).

Policy Definition Language

A Policy Definition Language is required to define and agree access and usage policies. The defined and agreed policies can be used directly or translated into an executable language. We propose to use ODRL as an interoperable standard for the negotiation and acceptance of Access and Usage Policies. There is a well-defined profile for ORDL for Access Control as defined in [48].

APIs

Table 3-12: summarizes the APIs exposed by the Identity Management Component.

Endpoint	Title	Description
/login	Presents a credential as starting point for the auth process	Returns a rendered html with a QR encoding the login-starting point for the siop flow - e.g., 'openid://?scope=somethign&response_type=rt&response_mode=rm&client_id=ci&redirect_uri=uri&state=state&nonce=nonce'
/samedevice	Starts the SIOP flow for credentials hold by the same device	When the credential is already present in the requesting browser, the same-device flow can be used. It creates the login information and then redirects to the /authenticationresponse path.
/startsiop	Initiates the SIOP flow	Initiates the SIOP flow and returns the 'openid://...' connection string
/authentication_response	API to authenticate response	Stores the credential for the given session.
/token	Token endpoint	Token endpoint to exchange the authorization code with the actual JWT.
/well-known/jwks	Retrieve public key	Provides the list of public keys for the given verifier, to be used for verifying the JWTs.
/info	Get the DID	Returns the issuer and verifier DID, generated on startup

Co-funded by the European Union

/identifiers/{did}	Get DID Document	Gets the DID document corresponding to the DID.

Prior to onboarding, a device makes a registration request to the IM, as depicted in Figure 3-25. Upon receiving the registration request, the IM triggers an attestation request to the attestation (AT) service on behalf of the requesting device; the AT service proceeds with attesting the device (following a suitable attestation mechanism, as described in the following subsection) and provides the attestation result as response; the IM inspects the attestation result and only if the latter is acceptable according to the relevant (use-case defined) policies, the device is granted access to the platform.

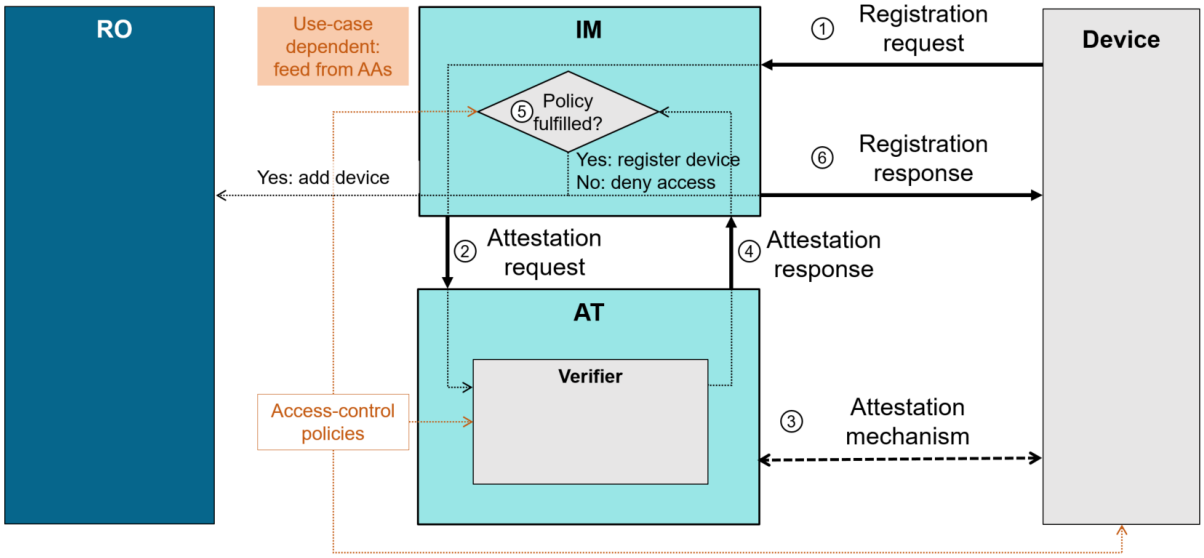

Figure 3-25: Registration and authentication of IoT devices in INCODE.

For the sake of modularity, the AT is tasked to only produce attestation results; it is the IM that evaluates them against access-control policies to establish whether and which device should be authorized to join INCODE. Moreover, as the attestation result is stored, along with the device ID, on the INCODE blockchain, users can later verify that the device in question has been attested and can thus be considered trustworthy. Table 3-13 describes the APIs for device registration exposed by the IM.

Table 3-13: Devise registration APIs exposed by the IM.

Endpoint	Title	Description
POST/devices/{id}	Registration of new devices	Request registration of new device *id*.
DELETE/devices/{id}	Deletion of existing devices	Request deletion of device *id* from catalog of registered devices.
GET/devices/{id}	Read device information	Retrieve information about device *id.*
PUT/devices/{id}	Update device information	Update information about device *id*.

Co-funded by the European Union

3.3.3.2 Attestation

The Attestation (AT) component is responsible for access control at the RM-L, to ensure that only verified devices are granted access to INCODE. It comprises different methodologies to validate the integrity and trustworthiness of IoT devices, to accommodate different security requirements depending on how each device is meant to serve its application domain. The AT component plays a crucial role in the authentication and registration process of devices; therefore, it needs to work in concert with the IM (ST-L) and RO (RM-L) components. The AT component in INCODE leverages different attestation mechanisms.

Remote attestation

Remote attestation provides a strong access-control service leveraging cryptographic techniques to boost trust from trusted components (e.g., trusted hardware). A remote attestation protocol lets a verifier interact with an attesting device (referred to as an *agent* below) to determine whether the device is in a trustworthy state. The trustworthiness of the device's state is determined based on reference values provided by an endorser (e.g., a device manufacturer) and pre-defined policies specified by the party relying on the attestation result (e.g., a service provider). An attestation protocol comprises two main phases:

1. *Challenge:* upon receiving an attestation request, the verifier generates a "challenge" for the attesting device.
2. *Response:* upon receiving a challenge, the attesting device generates a response, based on the challenge and on a fresh measurement of its internal state, as evidence of being in a trustworthy state. The verifier evaluates the supplied response by comparing it with suitable reference values, producing an attestation result.

The attestation result is finally supplied to the relying party for evaluation based on desired security policies. Figure 3-26 provides a high-level illustration of a remote attestation protocol.

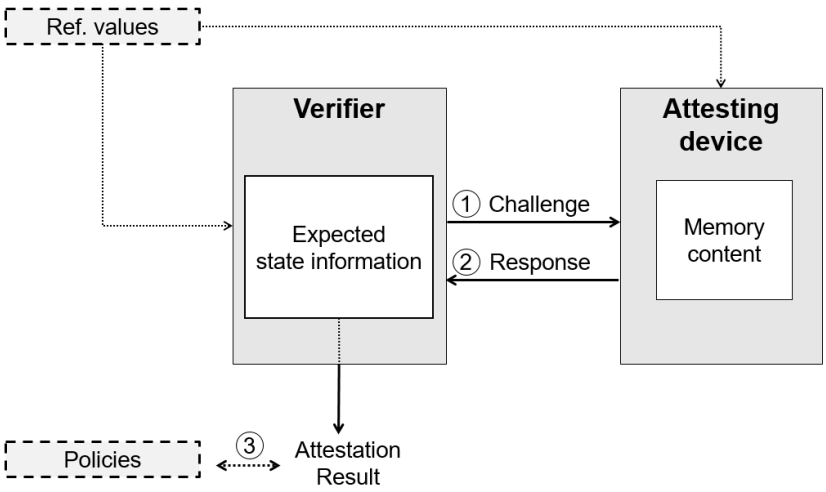

Figure 3-26: Hardware-based attestation: high-level description of a remote-attestation protocol.

In INCODE, the IM acts as the relying party on behalf of service providers/users who need to assess the trustworthiness of the device. Reference values depend on the device and are supplied by the device manufacturer; attestation policies are related to the desired security requirements and are therefore use-case dependent. Examples of said reference values and policies will be described in greater detail when analysing attestation scenarios for concrete use cases. Table 3-14 describes the APIs providing remote-attestation functionalities exposed by the AT component.

Co-funded by the European Union

Table 3-14: Remote attestation APIs exposed by the AT.

Endpoint	Title	Description
POST/agents/{id}	Attestation agent creation	Create attestation agent for device *id.*
DELETE/agents/{id}	Attestation agent deletion	Delete attestation agent for device *id*.
GET/agents/{id}	Remote attestation request	Request attestation status for device *id*.
PUT/agents/{id}/reactivate	Start remote attestation	(Re)start attestation agent for device *id* (continuous attestation).
PUT/agents/{id}/stop	Stop remote attestation	Stop attestation agent for device *id* (continuous attestation).

Software-based attestation

As a software attestation mechanism, INCODE will provide malware detection in terms of verifying the integrity and security of binary files and, more specifically, of software containers. In particular, malware detection focuses on recognising the presence of malicious software (malware). Malware includes viruses, worms, trojans, ransomware, spyware, and other malicious programs that can harm or compromise the integrity of a computing system. Based on the aforementioned remarks, the architecture design of the INCODE Malware Detection Components (MDC) is depicted in Figure 3-27. MDC consists of four modules: (a) Binary Receiving Module (BRM), (b) Visual Representation Generation Module (VRGM), (c) Malware Detection Module (MDM) and (d) Notification Module (NM). BRM represents an input interface that is responsible for receiving the software containers as binary files. Next, VGRM will convert a binary file into a visual representation. For this purpose, each byte of the binary file is converted into a pixel, utilising a pre-defined colour scheme, such as (a) Black: 00, (b) White: FF, (c) Blue: printable characters and (d) Red: everything else. As a result, each pixel is then placed on a two-dimensional visual representation, considering the locality of each byte. In particular, the bytes that are close to each other should also be placed as near as possible on the two-dimensional representation. To this end, clustering techniques will be adopted by VRGM. Then, the MDM receives the two-dimensional representations and undertakes to recognise the presence of malware, utilising transfer learning and pre-trained Convolutional Neural Networks (CNNs), such as Visual Geometric Group (VGG) and ResNet architectures. Finally, based on the detection outcomes, the Notification Module (NM) is responsible for generating the corresponding security events. For this purpose, the Malware Information Sharing Platform (MISP) will be used.

Co-funded by the European Union

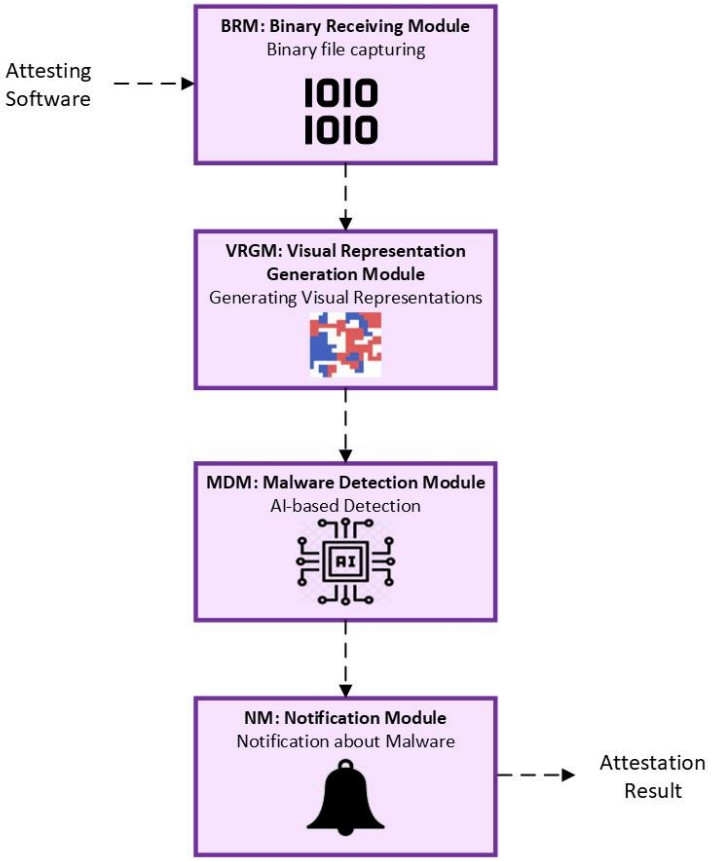

Figure 3-27: Overview of the Malware Detection Component

Table 3-15 describes the software attestation APIs exposed by the INCODE ST-L Attestation component.

Table 3-15: Software attestation APIs exposed by the AT.

Endpoint	Title	Description
POST/attest	Initiate software attestation	Request attestation of new software.
GET/attest/status	Check status	Check the status of a previously initiated attestation request.
GET/attest/report	Attestation report	Once the attestation process is complete retrieve the attestation report.
GET/attest/history	Attestation history	Retrieve a list of previous attestation requests and their statuses.

3.3.3.3 Data Provenance

Data Provenance refers to a documented trail that accounts for the origin of data and where it has moved from to where it is presently. The data coming from the various IoT devices and consumed for different services in INCODE project makes the data provenance and its traceability a challenge.

Co-funded by the European Union

Many existing solutions incorporate third-party auditor (TPA) programs to maintain data integrity based on log files. This process increases the required storage size as well as communication and computation overheads. At the same time, it also brings many security concerns. Various solutions and practices have been explored to preserve data confidentiality and provide information security. Blockchain technology has a promising solution to provide security and privacy in IoT Big Data systems. Blockchain offers distributed data storage. It can be deployed to provide data provenance for various applications by recording data operations from blockchain transactions. Embedding the data provenance (enriched by blockchain technology) into Big Data applications enhances system security and privacy while ensures data availability. The blockchain-enabled data provenance mechanism for Big Data applications in IoT systems guarantees data verifiability and integrity. This is because the data operations are recorded in the form of the transaction by every block in the blockchain network. Different devices within the IoT edge cloud architecture impose various trust concerns on the systems. Hence, a provenance mechanism is applicable to record the origin of multiple sensor data to meet these concerns. The DLT (Distributed Ledger Technology) characteristics of the Blockchain can be used to implement the Data Provenance and Traceability in the INCODE project. The Data Provenance Flow is as follows:

1. In the INCODE platform, Context Producers (IoT Hub or IoT devices or Drones, etc.) send updates or create entities to the IOT-C broker and, at the same time, send the wallet information (in the form of Verified Credentials for example) is supplied to the DLT Adapter.
2. It is assumed that Identity Manager verifies the wallet information and credentials even before the API call reach IOT-C and wallet information is reached DLT Adapter.
3. DLT Adapter gets the signature from the wallet.
4. Once the signature is available and the Entity updated/created and been supplied with the details, it pushes the transaction record with Entity information to ETH Compatible Node.
5. The same steps 1 to 4 for also are applied to the Context Consumers except that for Get and Subscribe API calls.

So, every transaction over the Entity (update or create or get or subscribe) has a record in ETC compatible DLT. The Flow is explained in Figure 3-28.

Pre-condition is that Content Producers and Context Consumers are identified with the wallet (or Verified Credentials) and the Identity Manager is compatible to validate the Wallet information.

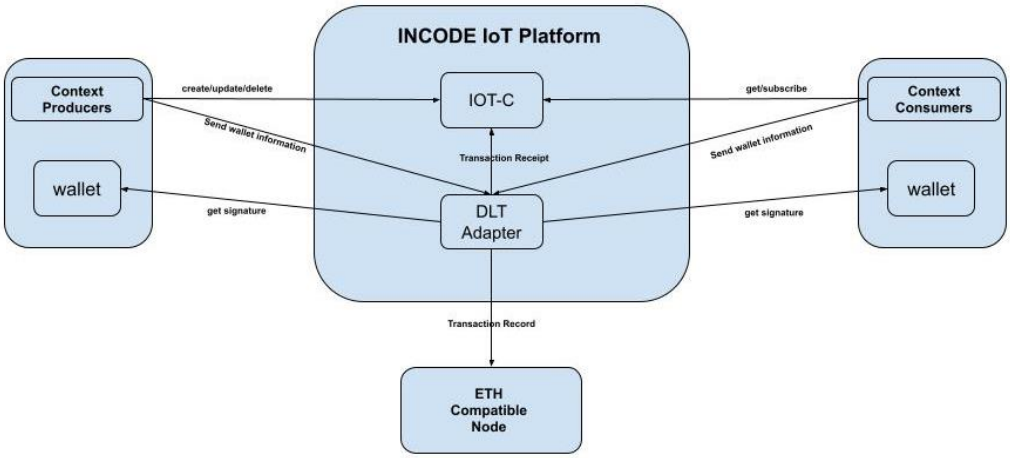

Figure 3-28: Flow of DLT Adapter component used for Data Provenance.

Finally, Data Provenance Tool provides a subset of the ETSI NGSI-LD which provides the functionalities to persist an operation represented by an Entity into the Ethereum compatible node. Table 3-16 summarises all these operations that will be exposed by the RM-L and can be used to enable the use of the Data Provenance Tool with other parts of the INCODE platform.

Table 3-16: List of APIs supported by the DLT adapter for Data Provenance.

Endpoint	Title	Description
/ngsi-ld/v1/entities	Create an Entity	Persist Entity creation into the Ethereum compatible Node and store the result of the transaction into an IoT-C
/ngsi-ld/v1/entities	Query an Entity	Request the information about one operation from Ethereum compatible node and store the result of the transaction into an IoT-C
/ngsi-ld/v1/entities/{entityId}/attrs	Update the information of an Entity	Update some of the attributes of an entity into the Ethereum compatible node and store the result of the transaction into an IoT-C
/ngsi-ld/v1/entityOperations/upsert	Create a batch of Entities	Create a batch of Entities, updating each of them if already exist into the Ethereum compatible node and store the result of the transaction into an IoT-C
/ngsi-ld/v1/entities/{entityId}	Retrieve the information from one Entity	Get the information associated to an Entity from the Ethereum compatible node and store the result of the transaction into an IoT-C

3.3.3.4 Attack Mitigation

As illustrated in Figure 3-29, the Attack Mitigation Component is composed of four main modules: (a) Traffic Capturing Module (TCM), (b) Flow Generation Module (FGM), (c) Intrusion Detection Module (IDM) and (d) Attack Mitigation Module (AMM). The first module is responsible for capturing the network traffic data that will be analysed for detecting and mitigating potential network-based attacks. This process usually involves intercepting and analysing the network traffic data exchanged between various entities. For this purpose, TCM will take full advantage of open-source packet sniffers that are able to capture and decode the various packets. Next, FGM will receive and analyse the network traffic data (i.e., pcap files) in order to generate bidirectional Transmission Control Protocol/ Internet Protocol (TCP/IP) flow statistics. Next, this kind of statistics will be used by the IDM to detect the presence of potential attacks. For this purpose, IDM will adopt pre-trained AI models that will be capable of discriminating benign and malicious network flows. Finally, considering the detection results from IDM, AMM will next take over to identify the appropriate mitigation actions based on a cost-benefit analysis, guiding the P4 controller to mitigate the cyberattack in a timely and effective manner. To this end, AMM will use offline Reinforcement Learning (RL) agents, considering a finite set of mitigation actions that can be executed by the P4 controller(s) in each P4 environment. Then, after identifying the appropriate mitigation action, AMM will inform the P4 controller, using its communication APIs.

Co-funded by the European Union

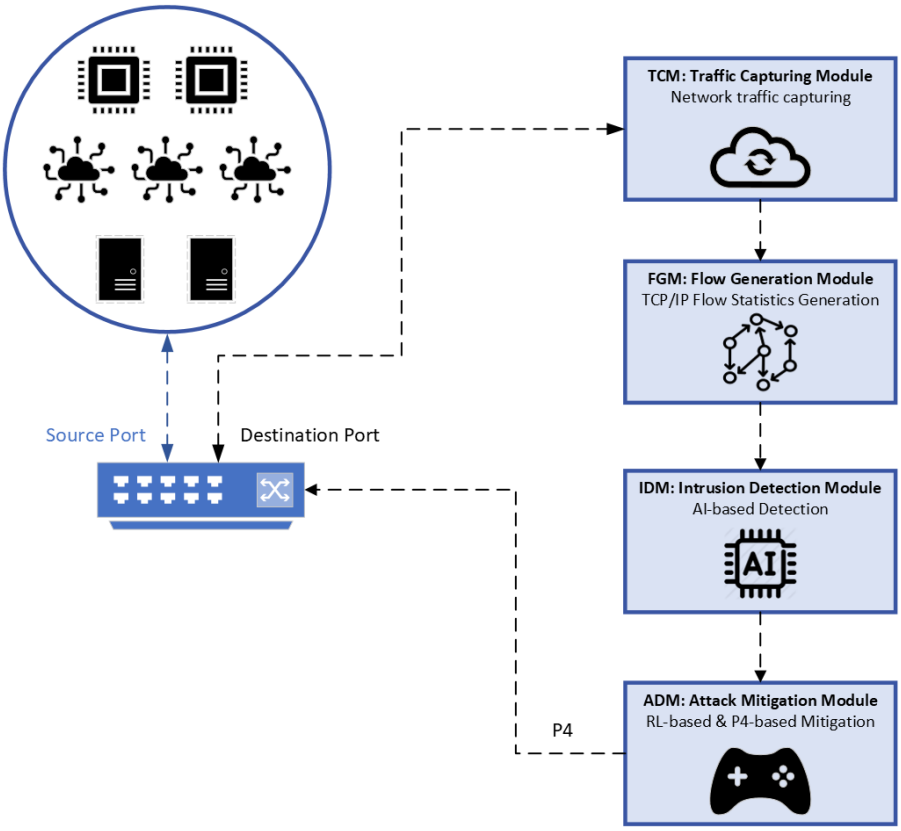

Figure 3-29: Overview of the Attack Mitigation Component

API

Attack mitigation is implemented as a network function that is applied to the data plane through the Network Fabric controller and drivers described in Section 3.3.2.2.3. To this end, a dedicated attack mitigation program will be implemented in e.g., the P4 programming language and the API of the P4 Network Fabric Controller (see Section 3.3.2.2.3 and Table 3-9) will be used to enforce the P4 rules to the designated switches.

Co-funded by
the European Union

4 Requirements

The objective of this section is to provide a consolidated view of the INCODE requirements and provide a formal coding to facilitate the tracking of updates up to the project's lifespan. The requirements are divided in three categories:

- **Business requirements (BSR)** refer to the specific needs, objectives, and expectations of a business or organization that must be fulfilled by the INCODE platform. They are divided in seven subcategories:
  - **Business Requirements for Vertical End Users (USER)**.
  - **Business Requirements for Vertical Service Providers (SP).**
  - **Business Requirements for Infrastructure Service Providers (ISP).**
  - **Business Requirements for Infrastructure Owners (IO).**
  - **Business Requirements for Vendors (VEN).**
  - **Business Requirements for System Integrators (SYS).**
- **Functional requirements (FNR)** describe the desire behaviour, features, and capabilities of the INCODE platform, outlines what the system should do or how to do it to meet the requirements of the users and achieve the business objectives. They are divided in four subcategories:
  - **Generic Functional requirements**, which includes Generic Platform Requirements (GPR), Generic RM-L Implementation Requirements (GRML), Generic SM-L Implementation Requirements (GSML), Generic User Interfacing Requirements and Information sharing (GUIR), Generic Programming Requirements (GPROG), and Generic Security Requirements (GSEC).
  - **Functional SM-L requirements (SML)**, which includes User Interfaces Functional Requirements (SML.UI), RBAC API Functional Requirements (SML.RBAC), Service Orchestrator Functional Requirements (SML.SO), Telemetry Functional Requirements (SML.TM), Software Lifecycle Functional Requirements (SML.SL), and IPD Functional Requirements (SML.IDP).
  - **Functional RM-L requirements (RML)**, which includes Scheduler Functional Requirements (RML.SCH), Resource Manager Functional Requirements (RML.RM), IoT Controller Functional Requirements (RML.CTL.IOT), IoT Driver Functional Requirements (RML.DRV.IOT), RAN Controller Functional Requirements (RML.CTL.RAN), RAN Driver Functional Requirements (RML.DRV.RAN), Network Fabric Controller Functional Requirements (RML.CTL.NF), Network Fabric Driver Functional Requirements (RML.DRV.NF), Mobile Core Controller Functional Requirements (RML.CTL.MC), Mobile Core Driver Functional Requirements (RML.DRV.MC), Cloud Controller Functional Requirements (RML.CTL.CL), Cloud Driver Functional Requirements (RML.DRV.CL)
  - **Functional ST-L requirements (STL)**, which includes Identity Management Functional Requirements (STL.IM), Attestation Functional Requirements (STL.AT), Data Provenance Functional Requirements (STL.DP), Attack Mitigation Functional Requirements (STL.AM).
- **Non-functional requirements (NFR)** describe the properties that the INCODE final product must have to judge the operation of the system.

Moreover, INCODE has adopted the MoSCoW technique to prioritize the creation of requirements. This method is used in project management and requirements gathering. The intention is to prioritise and categorize these requirements based on their importance and urgency. MoSCoW method classifies requirements in four different premises:

- **Must have**: requirements that are critical to the success of the INCODE project or the core functionality of the product. They are not negotiable and cannot be compromised.
- **Should have**: requirements are important but not critical, therefore can be prioritized lower that must-have requirements.

Co-funded by the European Union

- **Could have**: not mandatory requirements. These requirements are desirable but not essential for the project success or core functionality.
- **Won't have**: requirements that has been dismissed or have low priority and can be excluded intentionally from the current project.

This method was applied separately to all requirements for each of the three topics previously mentioned. The following sections list only the first three of them because we were not considered to include low priority requirements in this analysis.

4.1 Business Requirements

This section provides business requirements for key INCODE stakeholders.

4.1.1 Business Requirements for Vertical End Users

Req. Id	Requirement Description	
BSR.USER.1	The INCODE platform **MUST** provide security measures and data privacy controls to safeguard sensitive information of vertical end users, which includes encryption, access control, data backup, and compliance with data protection regulations.	
	Affected components	ALL
	Contributing Partner	ALL
	Comment	//

Table 4-1: BSR.USER.1 requirement

Req. Id	Requirement Description	
BSR.USER.2	The INCODE platform **MUST** provide adequate training and support resources towards the vertical end users, including documentation and online tutorials.	
	Affected components	ALL
	Contributing Partner	ALL
	Comment	//

Table 4-2: BSR.USER.2 requirement

Req. Id	Requirement Description	
BSR.USER.3	The UI **SHOULD** include a usable API and preferably an UI/GUI/IDE that could have the information and the services available	
	Affected components	IDE/API development
	Contributing Partner	ALL
	Comment	//

Table 4-3: BSR.USER.3 requirement

Req. Id	Requirement Description	
BSR.USER.4	INCODE **SHOULD** provide details about the overall license schema and the pricing model. Also providing details about the way that can interact with the service (through API, as library, pay per hit, pay per month, pay one off)	
	Affected components	ALL
	Contributing Partner	ALL
	Comment	//

Table 4-4: BSR.USER.4 requirement

Co-funded by the European Union

Req. Id	Requirement Description	
BSR.USER.5	The INCODE platform **SHOULD** support smooth integration with industry specific data sources or platforms commonly used in the cloud-edge environment by vertical end users.	
	Affected components	ALL
	Contributing Partner	ALL
	Comment	//

Table 4-5: BSR.USER.5 requirement

Req. Id	Requirement Description	
BSR.USER.6	The INCODE platform **SHOULD** support features and functionalities that facilitate compliance monitoring, audit trails and reporting capacities to the vertical end users.	
	Affected components	ALL
	Contributing Partner	ALL
	Comment	//

Table 4-6: BSR.USER.6 requirement

Req. Id	Requirement Description	
BSR.USER.7	The INCODE platform **SHOULD** support the provision of specific reports and key performance indicators to support informed decision-making about the deployment in the cloud-edge.	
	Affected components	ALL
	Contributing Partner	ALL
	Comment	//

Table 4-7: BSR.USER.7 requirement

Req. Id	Requirement Description	
BSR.USER.6	User friendly interface for software(container) deployment **SHOULD** include a usable API and preferably an UI/GUI/IDE that could have the information and the services available.	
	Affected components	IDE/API development
	Contributing Partner	ALL
	Comment	//

Table 4-8: BSR.USER.6 requirement

Req. Id	Requirement Description	
BSR.USER.8	INCODE **SHOULD** provide details about the overall license schema and the pricing model. Also provide details about the way that can interact with the service (through API, as library, pay per hit, pay per month, pay one off)	
	Affected components	INCODE components
	Contributing Partner	ALL
	Comment	//

Table 4-9: BSR.USER.8 requirement

Co-funded by the European Union

Req. Id	Requirement Description	
BSR.USER.9	INCODE platform **SHOULD** be compliant with standards.	
	Affected components	ALL
	Contributing Partner	ALL
	Comment	//

Table 4-10: BSR.USER.9 requirement

Req. Id	Requirement Description	
BSR.USER.10	INCODE platform MUST ensure end-to-end secure data transactions in order to protect sensitive data.	
	Affected components	ST-L Components
	Contributing Partner	ALL
	Comment	//

Table 4-11: BSR.USER.10 requirement

4.1.2 Business Requirements for Vertical Service Providers

Req. Id	Requirement Description	
BSR.SP.1	The INCODE platform **MUST** ensure secure data transmission and storage for sensitive vertical service data.	
	Affected components	Data transmission and storage systems.
	Contributing Partner	NEC
	Comment	Security of data is of utmost importance in this context.

Table 4-12: BSR.SP.01 requirement

Req. Id	Requirement Description	
BSR.SP.2	The INCODE platform **MUST** enable seamless integration of vertical services.	
	Affected components	UI, IDP, and SO
	Contributing Partner	Contributors to UI, IDP, and SO
	Comment	Seamless integration enhances overall user experience.

Table 4-13: BSR.SP.02 requirement

Req. Id	Requirement Description	
BSR.SP.3	The INCODE platform **MUST** provide customizable user interfaces for each vertical service to enhance usability	
	Affected components	User interface and experience modules.
	Contributing Partner	UI development team.
	Comment	Customizable UIs cater to diverse user preferences.

Table 4-14: BSR.SP.03 requirement

Co-funded by the European Union

Req. Id	Requirement Description	
BSR.SP.4	The INCODE platform **MUST** ensure compliance with industry-specific regulations and standards for vertical services.	
	Affected components	SO and RO
	Contributing Partner	UBI, UoP
	Comment	//

Table 4-15: BSR.SP.04 requirement

Req. Id	Requirement Description	
BSR.SP.5	The INCODE platform **MUST** enable real-time monitoring and alerts for critical vertical service parameters.	
	Affected components	Telemetry
	Contributing Partner	SIEMENS
	Comment	Real-time monitoring enhances operational efficiency.

Table 4-16: BSR.SP.05 requirement

Req. Id	Requirement Description	
BSR.SP.6	The INCODE platform **MUST** ensure high availability and reliability of vertical service operations.	
	Affected components	IDP, SO, and RO
	Contributing Partner	Contributors to IDP, SO, and RO
	Comment	//

Table 4-17: BSR.SP.06 requirement

Req. Id	Requirement Description	
BSR.SP.7	The INCODE platform **MUST** facilitate easy configuration and customization of vertical service parameters.	
	Affected components	SO and RO
	Contributing Partner	UBI and UoP
	Comment	//

Table 4-18: BSR.SP.07 requirement

Req. Id	Requirement Description	
BSR.SP.8	The INCODE platform **COULD** provide vertical service-specific billing and subscription management capabilities.	
	Affected components	SO and RO
	Contributing Partner	UBI and UoP
	Comment	//

Table 4-19: BSR.SP.08 requirement

Req. Id	Requirement Description	
BSR.SP.9	The INCODE platform **COULD** support scalability to accommodate varying demand for vertical services.	
	Affected components	IDP

Co-funded by the European Union

	Contributing Partner	IDP contributors
	Comment	Scalability is vital for meeting fluctuating service demands.

Table 4-20: BSR.SP.09 requirement

Req. Id	Requirement Description	
BSR.SP.10	INCODE **SHOULD** be able to offer a platform that can be scaled up and down pedending of the fluctuations in demand and adjust their resources accordingly	
	Affected components	INCODE components
	Contributing Partner	All
	Comment	//

Table 4-21: BSR.SP.10 requirement

4.1.3 Business Requirements for Infrastructure Service Providers

Req. Id	Requirement Description	
BSR.ISP.1	INCODE platform **MUST** be highly scalable to meet varying demands from different clients. It should efficiently allocate cloud resources to support the growing number of IoT devices and edge infrastructure.	
	Affected components	PHY-L, RM-L
	Contributing Partner	UNIS
	Comment	//

Table 4-22: BSR.ISP.1 requirement

Req. Id	Requirement Description	
BSR.ISP.2	INCODE **MUST** support Centralized Infrastructure Lifecyle Management (i.e., the lifecycle of all the INCODE infrastructure components across the continuum will be managed from a single location).	
	Affected components	RM-L
	Contributing Partner	REDHAT
	Comment	//

Table 4-23: BSR.ISP.2 requirement

Req. Id	Requirement Description	
BSR.ISP.3	INCODE **MUST** support Centralized Application Development (I.e., applications are deployed at a single location, and it is up to INCODE infrastructure management layer to deploy the application components to the location in the continuum).	
	Affected components	RM-L
	Contributing Partner	REDHAT
	Comment	//

Table 4-24: BSR.ISP.3 requirement

Co-funded by the European Union

Req. Id	Requirement Description	
BSR.ISP.4	INCODE **MUST** provide extensibility capabilities on its platform, facilitating easily addition of new resource/device types	
	Affected components	RM-L
	Contributing Partner	REDHAT
	Comment	//

Table 4-25: BSR.ISP.4 requirement

Req. Id	Requirement Description	
BSR.ISP.5	INCODE platform **SHOULD** seamlessly integrate with existing cloud infrastructure and services offered by cloud providers. This ensures that cloud service providers can offer complementary services and solutions to their clients.	
	Affected components	PHY-L, RM-L
	Contributing Partner	UNIS
	Comment	//

Table 4-26: BSR.ISP.5 requirement

Req. Id	Requirement Description	
BSR.ISP.6	INCODE **SHOULD** robust security mechanisms, including authentication, attestation, and data protection, which are crucial for cloud service providers. Compliance with industry-specific regulations (e.g., GDPR, HIPAA) is essential when handling sensitive data.	
	Affected components	ST-L
	Contributing Partner	K3Y, NEC, FIWARE, UNIS
	Comment	//

Table 4-27: BSR.ISP.6 requirement

Req. Id	Requirement Description	
BSR.ISP.7	As the platform spans both edge and cloud environments, cloud providers **SHOULD** be provided with seamless integration with edge devices and infrastructure. This allows them to offer comprehensive solutions that encompass both edge and cloud resources.	
	Affected components	PHY-L
	Contributing Partner	ALL
	Comment	//

Table 4-28: BSR.ISP.7 requirement

Req. Id	Requirement Description	
BSR.ISP.8	INCODE platform **SHOULD** maintain high availability and reliability to minimize downtime and ensure uninterrupted service to cloud provider's clients.	
	Affected components	PHY-L, RM-L
	Contributing Partner	ALL
	Comment	//

Co-funded by the European Union

Table 4-29: BSR.ISP.8 requirement

Req. Id	Requirement Description	
BSR.ISP.9	Cloud providers **SHOULD** need mechanisms to manage and track data provenance and lineage, especially when dealing with large datasets and diverse data sources to provide a map of the data journey.	
	Affected components	ST-L
	Contributing Partner	FIWARE, UNIS
	Comment	//

Table 4-30: BSR.ISP.9 requirement

Req. Id	Requirement Description	
BSR.ISP.10	The platform **SHOULD** optimize resource usage efficiently to control operational costs for cloud providers. Automated hardware acceleration and resource allocation could play a role in cost management.	
	Affected components	RM-L
	Contributing Partner	UNIS
	Comment	//

Table 4-31: BSR.ISP.10 requirement

Req. Id	Requirement Description	
BSR.ISP.11	INCODE **SHOULD** offer robust monitoring and reporting capabilities, allowing cloud providers to keep track of the performance and health of their INCODE-based services. This is essential for proactive maintenance and issue resolution for their clients.	
	Affected components	RM-L
	Contributing Partner	ALL
	Comment	//

Table 4-32: BSR.ISP.11 requirement

Req. Id	Requirement Description	
BSR.ISP.12	INCODE platform **COULD** adhere to SLAs defined with cloud providers clients, ensuring that performance, uptime, and other agreed-upon metrics are met.	
	Affected components	RM-L
	Contributing Partner	UNIS, FIWARE, UBI
	Comment	//

Table 4-33: BSR.ISP.12 requirement

4.1.4 Business Requirements for Infrastructure Owners

Req. Id	Requirement Description	
BSR.IO.1	The availability and reliability of the INCODE platform's underlying infrastructure **MUST** be ensured.	
	Affected components	...
	Contributing Partner	...
	Comment	//

Co-funded by the European Union

Table 4-34: BSR.IO.1 requirement

Req. Id	Requirement Description	
BSR.IO.2	The resource manager **MUST** optimize resource allocation to efficiently handle varying workloads and usage patterns.	
	Affected components	RO
	Contributing Partner	UoP
	Comment	//

Table 4-35: BSR.IO.2 requirement

Req. Id	Requirement Description	
BSR.IO.3	The resource management layer **MUST** ensure compliance with industry standards and regulations.	
	Affected components	RM-L
	Contributing Partner	RM-L contributors
	Comment	//

Table 4-36: BSR.IO.3 requirement

Req. Id	Requirement Description	
BSR.IO.4	The security and privacy of the infrastructure and data hosted on the INCODE platform **MUST** be ensured.	
	Affected components	ST-L and RM-L
	Contributing Partner	ST-L and RM-L contributors
	Comment	//

Table 4-37: BSR.IO.4 requirement

Req. Id	Requirement Description	
BSR.IO.5	Efficient and flexible data storage solutions **MUST** be provided for the INCODE platform's data requirements.	
	Affected components	RO storage
	Contributing Partner	RedHat
	Comment	//

Table 4-38: BSR.IO.5 requirement

Req. Id	Requirement Description	
BSR.IO.6	The infrastructure owner **MUST** facilitate seamless integration and compatibility with other components and systems within the INCODE platform.	
	Affected components	RM-L
	Contributing Partner	RM-L contributors
	Comment	//

Table 4-39: BSR.IO.6 requirement

Co-funded by the European Union

Req. Id	Requirement Description	
BSR.IO.7	The infrastructure owner **MUST** collaborate with other stakeholders to plan and implement continuous improvements and upgrades to the infrastructure.	
	Affected components	RM-L
	Contributing Partner	RM-L contributors
	Comment	//

Table 4-40: BSR.IO.7 requirement

Req. Id	Requirement Description	
BSR.IO.8	The INCODE platform **MUST** support different communication protocols so as to be compatible with existing infrastructure.	
	Affected components	INCODE Drivers, INCODE Controllers
	Contributing Partner	FIWARE
	Comment	//

Table 4-41: BSR.IO.8 requirement

Req. Id	Requirement Description	
BSR.IO.9	The INCODE platform **MUST** be easy to be deployed so as to minimize tampering with the existing infrastructure.	
	Affected components	INCODE Platform
	Contributing Partner	Platform developers
	Comment	//

Table 4-42: BSR.IO.9 requirement

Req. Id	Requirement Description	
BSR.IO.10	The INCODE platform **MUST** be designed with minimum installation & deployment costs in mind so as to be an attractive solution for businesses	
	Affected components	INCODE Platform
	Contributing Partner	Platform developers
	Comment	//

Table 4-43: BSR.IO.10 requirement

4.1.5 Business Requirements for Hardware and Software Vendors

Req. Id	Requirement Description	
BSR.VEN.1	The INCODE platform **MUST** support open management APIs to a broad range of RAN, network fabric, Mobile Core, and Cloud devices	
	Affected components	RM-L controller & drivers
	Contributing Partner	Controllers and drivers' contributors
	Comment	//

Table 4-44: BSR.VEN.1 requirement

Co-funded by the European Union

Req. Id	Requirement Description	
BSR.VEN.2	The INCODE platform **MUST** support open telemetry APIs to a broad range of RAN, network fabric, Mobile Core, and Cloud devices	
	Affected components	RM-L controller & drivers, Telemetry
	Contributing Partner	Controllers and drivers' and Telemetry contributors
	Comment	//

Table 4-45: BSR.VEN.2 requirement

Req. Id	Requirement Description	
BSR.VEN.3	The INCODE platform **MUST** support open APIs for integration with popular software vendors for cloud orchestration, telemetry, storage, and 5G.	
	Affected components	RM-L controller & drivers, Telemetry
	Contributing Partner	Controllers and drivers' and Telemetry contributors
	Comment	//

Table 4-46: BSR.VEN.3 requirement

Req. Id	Requirement Description	
BSR.VEN.4	Hardware and Software Vendors **SHOULD** offer scalable solutions that can accommodate varying customer needs and growth.	
	Affected components	INCODE Platform
	Contributing Partner	All
	Comment	//

Table 4-47: BSR.VEN.4 requirement

Req. Id	Requirement Description	
BSR.VEN.5	Hardware and Software Vendors **SHOULD** provide comprehensive technical support and maintenance services.	
	Affected components	INCODE Platform
	Contributing Partner	All
	Comment	//

Table 4-48: BSR.VEN.5 requirement

4.1.6 Business Requirements for System Integrators

Req. Id	Requirement Description	
BSR.SYS.1	INCODE platform **SHOULD** scale seamlessly to accommodate a diverse range of IoT devices and edge infrastructure. It should also allow for flexible integration of various IoT nodes, edge processors, and servers to meet different client needs.	
	Affected components	PHY-L, RM-L
	Contributing Partner	…All
	Comment	//

Table 4-49: BSR.SYS.1 requirement

Co-funded by the European Union

Req. Id	Requirement Description	
BSR.SYS.2	Integrators **SHOULD** be provided with robust developer support, including tools and resources for application development, testing, and deployment. A multi-role Integrated Development Platform (IDP) should streamline the development process and enable developer self-service.	
	Affected components	SM-L
	Contributing Partner	All
	Comment	//

Table 4-50: BSR.SYS.2 requirement

Req. Id	Requirement Description	
BSR.SYS.3	The platform **SHOULD** offer flexibility for integrators to introduce applications and policies without being bound by low-level infrastructure restrictions. This enables integrators to tailor solutions to specific client requirements.	
	Affected components	SM-L
	Contributing Partner	All
	Comment	//

Table 4-51: BSR.SYS.3 requirement

Req. Id	Requirement Description	
BSR.SYS.4	INCODE platform **SHOULD** adhere to regulatory and compliance standards from integrators perspective, especially if they serve clients in highly regulated industries like critical infrastructures or finance.	
	Affected components	RM-L, SM-L
	Contributing Partner	All
	Comment	//

Table 4-52: BSR.SYS.4 requirement

Req. Id	Requirement Description	
BSR.SYS.5	Integrators could benefit from the opportunity to collaborate with relevant stakeholders in the INCODE ecosystem. Creating synergies and partnerships can enhance the business opportunities and offerings of the consolidator's customers.	
	Affected components	PHY-L, RM-L, SM-L, ST_L
	Contributing Partner	All
	Comment	//

Table 4-53: BSR.SYS.5 requirement

4.2 Functional Requirements

This section lists all the Functional requirements identified by the project team at the time of publication of this document. The requirements are grouped according to the architecture components. Both mandatory (MUST) and optional (COULD) requirements are listed.

Co-funded by the European Union

4.2.1 Generic Functional Requirements

This section provides a list of generic requirements that need to be provided by the INCODE platform.

4.2.1.1 Generic Platform Requirements

The following tables provide generic functional requirements related to the entire INCODE platform.

Req. Id	Requirement Description	
FNR.GPR.1	The platform's basic building units **MUST** Include the RM-L and SM-L units	
	Affected components	ALL
	Contributing Partner	UBI
	Comment	Platform definition

Table 4-54: FNR.GPR1 requirement

Req. Id	Requirement Description	
FNR.GPR.2	The SM-L components **COULD** be deployed as cloud-manged services at a central cloud infrastructure	
	Affected components	SM-L
	Contributing Partner	UBI
	Comment	Deployment strategy

Table 4-55: FNR.GPR.2 requirement

Req. Id	Requirement Description	
FNR.GPR.3	The RM-L components **COULD** be deployed in a highly distributed fashion	
	Affected components	RM-L
	Contributing Partner	UBI
	Comment	Deployment strategy

Table 4-56: FNR.GPR.3 requirement

Req. Id	Requirement Description	
FNR.GPR.4	Core RM-L components (e.g., Resource Manager, Scheduler) **COULD** be deployed as cloud-managed services at a central cloud infrastructure	
	Affected components	RM-L
	Contributing Partner	UBI
	Comment	Deployment strategy

Table 4-57: FNR.GPR.4 requirement

Req. Id	Requirement Description	
FNR.GPR.5	RM-L Controllers and Drivers **COULD** be deployed at designated edge locations on demand	
	Affected components	RM-L
	Contributing Partner	UBI
	Comment	Deployment strategy

Table 4-58: FNR.GPR.5 requirement

Co-funded by the European Union

4.2.1.2 Generic RM-L Implementation Requirements

The following tables provide generic functional requirements related to INCODE's RM-L.

Req. Id	Requirement Description	
FNR.GRML.1	The Resource Management Layer (RM-L) **MUST** deploy multiple infrastructure management instances across the continuum	
	Affected components	RM-L
	Contributing Partner	RedHat
	Comment	//

Table 4-59: FNR.GRML.1 requirement

Req. Id	Requirement Description	
FNR.GRML.2	The Resource Management Layer (RM-L) **MUST** maintain a consistent view of the entire system (centralized control)	
	Affected components	RM-L
	Contributing Partner	RedHat
	Comment	//

Table 4-60: FNR.GRML.2 requirement

Req. Id	Requirement Description	
FNR.GRML.3	The Resource Management Layer (RM-L) **MUST** provide mechanisms to organize compute and storage resources across the continuum as a single virtual server	
	Affected components	RM-L
	Contributing Partner	RedHat
	Comment	//

Table 4-61: FNR.GRML.3 requirement

Req. Id	Requirement Description	
FNR.GRML.4	The Resource Management Layer (RM-L) **MUST** Two level scheduler: single point of application deployment, meta-scheduler selects target cluster, scheduler at target cluster selects node	
	Affected components	RM-L
	Contributing Partner	RedHat
	Comment	//

Table 4-62: FNR.GRML.4 requirement

Req. Id	Requirement Description	
FNR.GRML.5	The Resource Management Layer (RM-L) **MUST** be accessible at every (edge) site where INCODE may deploy applications	
	Affected components	RM-L
	Contributing Partner	UBI
	Comment	Northbound connectivity (towards SM-L)

Table 4-63: FNR.GRML.5 requirement

Req. Id	Requirement Description

Co-funded by the European Union

FNR.GRML.6	The Resource Management Layer (RM-L) **MUST** provide technology-agnostic northbound APIs to SM-L and external stakeholders	
	Affected components	RM-L
	Contributing Partner	UBI
	Comment	Northbound connectivity (towards SM-L)

Table 4-64: FNR.GRML.6 requirement

Req. Id	Requirement Description	
FNR.GRML.7	The Resource Management Layer (RM-L) **MUST** provide a northbound API for end-to-end slicing (i.e., resource allocation, adaptation, and release) as a service to the SM-L	
	Affected components	RM-L
	Contributing Partner	UBI
	Comment	Northbound connectivity (towards SM-L). Focus of T3.2

Table 4-65: FNR.GRML.7 requirement

Req. Id	Requirement Description	
FNR.GRML.8	The Resource Management Layer (RM-L) **MUST** provide a northbound API for application scheduling as a service to the SM-L	
	Affected components	RM-L
	Contributing Partner	RedHat
	Comment	Northbound connectivity (towards SM-L). Focus of T3.1

Table 4-66: FNR.GRML.8 requirement

Req. Id	Requirement Description	
FNR.GRML.9	The Resource Management Layer (RM-L) **MUST** be able to decompose end-to-end slicing requests into sub-slices targeting different infrastructure controllers	
	Affected components	RM-L
	Contributing Partner	UBI
	Comment	Internal logic. Focus of T3.2

Table 4-67: FNR.GRML.9 requirement

Req. Id	Requirement Description	
FNR.GRML.10	The Resource Management Layer (RM-L) **MUST** implement at least 5 infrastructure controller interfaces towards (i) cloud, (ii) network, (iii) mobile core, (iv) RAN, and (v) IoT controllers (see controller-specific requirements below)	
	Affected components	RM-L
	Contributing Partner	UBI
	Comment	Internal logic. Focus of T3.2

Table 4-68: FNR.GRML.10 requirement

Req. Id	Requirement Description
FNR.GRML.11	The Resource Management Layer (RM-L) **MUST** implement a cloud controller (Cloud-C) interface for compute resource allocation and management on top of modern cloud orchestration platforms, such as Kubernetes (container-based) and/or OpenStack (VM-based)

Co-funded by the European Union

	Affected components	RM-L
	Contributing Partner	UBI
	Comment	Internal logic. Focus of T3.2

Table 4-69: FNR.GRML.11 requirement

Req. Id	Requirement Description	
FNR.GRML.12	The Resource Management Layer (RM-L) **MUST** implement a network controller (Net-C) interface for intra-datacenter (within an edge/core site) connectivity, supporting next-generation SDN technologies, such as P4, OpenConfig/YANG	
	Affected components	RM-L
	Contributing Partner	UBI
	Comment	Internal logic. Focus of T3.2

Table 4-70: FNR.GRML.12 requirement

Req. Id	Requirement Description	
FNR.GRML.13	The Resource Management Layer (RM-L) **MUST** implement a network controller (Net-C) interface for inter-datacenter (across sites) connectivity	
	Affected components	RM-L
	Contributing Partner	UoP, UBI
	Comment	Internal logic. Focus of T3.2

Table 4-71: FNR.GRML.13 requirement

Req. Id	Requirement Description	
FNR.GRML.14	The Resource Management Layer (RM-L) **MUST** implement a mobile core controller (Mob-C) interface supporting 5G Core NFs' (e.g., AMF, SMF, UPF) management through open-source orchestration platforms, such as ETSI Open Source MANO (OSM)	
	Affected components	RM-L
	Contributing Partner	UBI
	Comment	Internal logic. Focus of T3.2

Table 4-72: FNR.GRML.14 requirement

Req. Id	Requirement Description	
FNR.GRML.15	The Resource Management Layer (RM-L) **MUST** implement a radio access network (RAN) controller (RAN-C) interface supporting RAN NFs' (i.e., O-DU, O-CU) management either through the integration with an O-RAN RAN Intelligent Controller, such as µONOS, or natively	
	Affected components	RM-L
	Contributing Partner	UBI
	Comment	Internal logic. Focus of T3.2

Table 4-73: FNR.GRML.15 requirement

Req. Id	Requirement Description
FNR.GRML.16	The Resource Management Layer (RM-L) **MUST** implement an IoT controller (IoT-C) interface supporting interoperability with existing IoT platforms, such as FIWARE, for managing IoT devices and data

Co-funded by the European Union

	Affected components	RM-L
	Contributing Partner	UBI
	Comment	Internal logic. Focus of T3.2

Table 4-74: FNR.GRML.16 requirement

Req. Id	Requirement Description	
FNR.GRML.17	The Resource Management Layer (RM-L) **MUST** be able to interface with multiple cloud controllers (one or more edge domains and a core domain) at the same time	
	Affected components	RM-L
	Contributing Partner	UBI
	Comment	Internal logic. Focus of T3.2

Table 4-75: FNR.GRML.17 requirement

Req. Id	Requirement Description	
FNR.GRML.18	The Resource Management Layer (RM-L) **MUST** provide cloud drivers (Cloud-D) to translate Cloud-C commands into specific cloud agent instructions, e.g., for OpenStack, nova agents and for Kubernetes, kubelet agents	
	Affected components	RM-L
	Contributing Partner	UBI
	Comment	Southbound connectivity (towards infrastructure). Focus of T3.2

Table 4-76: FNR.GRML.18 requirement

Req. Id	Requirement Description	
FNR.GRML.19	The Resource Management Layer (RM-L) **MUST** provide programmable intra-datacenter network drivers (Net-D) to translate high level Net-C flow objectives into protocol-specific instructions for next-generation SDN fabric of programmable white-box switches, supporting state-of-the-art	
	Affected components	RM-L
	Contributing Partner	UBI
	Comment	Southbound connectivity (towards infrastructure). Focus of T3.2

Table 4-77: FNR.GRML.19 requirement

Req. Id	Requirement Description	
FNR.GRML.20	The Resource Management Layer (RM-L) **MUST** provide mobile core drivers (Mob-D) which realise distributed 5G mobile core user plane functions (UPFs), thus providing local breakout at all remote INCODE (edge) cloud sites	
	Affected components	RM-L
	Contributing Partner	UBI
	Comment	Southbound connectivity (towards infrastructure). Focus of T3.2

Table 4-78: FNR.GRML.20 requirement

Req. Id	Requirement Description

Co-funded by the European Union

FNR.GRML.21	The Resource Management Layer (RM-L) **MUST** provide RAN drivers (RAN-D) which implement critical RAN functions, such as distributed units (DUs) as well as control and user plane central units (CUs)	
	Affected components	RM-L
	Contributing Partner	UBI
	Comment	*Southbound connectivity (towards infrastructure). Focus of T3.2*

Table 4-79: FNR.GRML.21 requirement

Req. Id	Requirement Description	
FNR.GRML.22	The Resource Management Layer (RM-L) **MUST** provide IoT drivers (IoT-D) for inventory and management capabilities across several IoT technologies	
	Affected components	RM-L
	Contributing Partner	UBI, FIWARE
	Comment	Southbound connectivity (towards infrastructure). Focus of T3.2

Table 4-80: FNR.GRML.22 requirement

Req. Id	Requirement Description	
FNR.GRML.23	The Resource Management Layer (RM-L) **MUST** use data models' representation of the information based on open schemas definition and using well-known ontologies to represent the information	
	Affected components	RM-L
	Contributing Partner	FIWARE
	Comment	Registration, authentication, and access control. Focus of T3.3. (Mentioned in the proposal; unclear if meaningful and which partner is responsible for this task)

Table 4-81: FNR.GRML.23 requirement

Req. Id	Requirement Description	
FNR.GRML.24	The Resource Management Layer (RM-L) **COULD** ensure that each infrastructure controller abstracts the technological specificities of the underlying hardware	
	Affected components	RM-L
	Contributing Partner	UBI
	Comment	Internal logic. Focus of T3.2

Table 4-82: FNR.GRML.24 requirement

Req. Id	Requirement Description	
FNR.GRML.25	The Resource Management Layer (RM-L) **COULD** develop a data provenance mechanism based on permissioned blockchain for logging/auditing models, datasets and s/w artefacts destined for use by the applications	
	Affected components	RM-L
	Contributing Partner	FIWARE
	Comment	Internal logic. Focus of T3.2

Table 4-83: FNR.GRML.25 requirement

Infrastructure controller in the RM-L

Co-funded by the European Union

Req. Id	Requirement Description	
FNR.GRML.26	Each infrastructure controller in the RM-L **MUST** have access to the underlying infrastructure it will be affecting (e.g., IoT, RAN, Mobile Core, Network fabric, Cloud)	
	Affected components	RM-L
	Contributing Partner	UBI
	Comment	Southbound connectivity (towards infrastructure). Focus of T3.2

Table 4-84: FNR.GRML.26 requirement

Req. Id	Requirement Description	
FNR.GRML.27	Each infrastructure controller in the RM-L **MUST** be able to make changes to the underlying infrastructure	
	Affected components	RM-L
	Contributing Partner	UBI
	Comment	Southbound connectivity (towards infrastructure). Focus of T3.2

Table 4-85: FNR.GRML.27 requirement

Req. Id	Requirement Description	
FNR.GRML.28	Each infrastructure controller in the RM-L **MUST** translate abstract control plane instructions into technology-specific instructions towards relevant infrastructure drivers	
	Affected components	RM-L
	Contributing Partner	UBI
	Comment	Southbound connectivity (towards infrastructure). Focus of T3.2

Table 4-86: FNR.GRML.28 requirement

Req. Id	Requirement Description	
FNR.GRML.29	Each infrastructure controller in the RM-L **COULD** get access to infrastructure telemetry	
	Affected components	RM-L
	Contributing Partner	UBI
	Comment	Southbound connectivity (towards infrastructure). Focus of T3.2

Table 4-87: FNR.GRML.29 requirement

4.2.1.3 Generic SM-L Implementation Requirements

The following tables provide generic functional requirements related to INCODE's SM-L.

Req. Id	Requirement Description	
FNR.GSML.1	The application and business programmability stratum (SM-L) **MUST** provide a northbound interface of the INCODE SM-L module for deployment by application provider	
	Affected components	ALL
	Contributing Partner	UBI

Co-funded by the European Union

	Comment	//

Table 4-88: FNR.GSML.1 requirement

Req. Id	Requirement Description	
FNR.GSML.2	The application and business programmability stratum (SM-L) **MUST** provide telemetry functions that allow infrastructure owners and operators to oversee and monitor their infrastructures and the overlay mesh of microservices	
	Affected components	SM-L
	Contributing Partner	AGE/UBI
	Comment	//

Table 4-89: FNR.GSML.2 requirement

Req. Id	Requirement Description	
FNR.GSML.3	The SM-L telemetry functions **MUST** provide real-time data about the performance and health of the infrastructure and the applications running on it	
	Affected components	SM-L
	Contributing Partner	AGE/UBI
	Comment	//

Table 4-90: FNR.GSML.3 requirement

Req. Id	Requirement Description	
FNR.GSML.4	The application and business programmability stratum (SM-L) **MUST** provide an open and extensible programming toolset to facilitate the development and deployment of large swarms of devices at the edge	
	Affected components	SM-L
	Contributing Partner	AGE/UBI
	Comment	//

Table 4-91: FNR.GSML.4 requirement

Req. Id	Requirement Description	
FNR.GSML.5	The application and business programmability stratum (SM-L) **MUST** implement a secure and trusted framework for registering and authenticating IoT device and edge nodes entering the system as well as the data sharing and application deployment	
	Affected components	SM-L
	Contributing Partner	AGE/UBI
	Comment	//

Table 4-92: FNR.GSML.5 requirement

Req. Id	Requirement Description	
FNR.GSML.6	The application and business programmability stratum (SM-L) **MUST** provide an Internal Developer Platform (IDP) for Edge IoT application configuration, deployment, and self-serving support to reduce programming complexity and maintenance	
	Affected components	SM-L, IDP
	Contributing Partner	AGE/UBI
	Comment	//

Co-funded by the European Union

Table 4-93: FNR.GSML.6 requirement

Req. Id	Requirement Description	
FNR.GSML.7	The application and business programmability stratum (SM-L) **MUST** provide real-time synchronization of the information between SM-L and IMLs in terms of configuration of the platform and real-time communication through federation capabilities of the SM-L and IMLs nodes	
	Affected components	SM-L, RM-L
	Contributing Partner	FIWARE
	Comment	//

Table 4-94: FNR.GSML.7 requirement

4.2.1.4 Generic User Interfacing Requirements and Information Sharing

The following tables provide user facing requirements and requirements related to information sharing.

Req. Id	Requirement Description	
FNR.GUIR.1	The platform frontend interface (GUI) **MUST** provide the necessary resource deployment information to the operator, via a monitoring dashboard	
	Affected components	SM-L
	Contributing Partner	AGE/UBI
	Comment	//

Table 4-95: FNR.GUIR.1 requirement

Req. Id	Requirement Description	
FNR.GUIR.2	The platform frontend interface (GUI) **MUST** provide role-based access control assisted to accommodate the real needs from multiple types of stakeholders that manage the heterogeneous platforms	
	Affected components	SM-L
	Contributing Partner	AGE/UBI
	Comment	//

Table 4-96: FNR.GUIR.2 requirement

Req. Id	Requirement Description	
FNR.GUIR.3	The platform frontend interface (GUI) **COULD** provide intuitive, user-friendly, and easy access to the features and tools of the SM-L	
	Affected components	SM-L
	Contributing Partner	AGE/UBI
	Comment	//

Table 4-97: FNR.GUIR.3 requirement

Req. Id	Requirement Description	
FNR.GUIR.4	Information sharing **MUST** include mechanisms for ensuring the confidentiality, integrity, and availability of data and applications running on the platform	
	Affected components	SM-L
	Contributing Partner	AGE/UBI

Co-funded by the European Union

	Comment	//

Table 4-98: FNR.GUIR.4 requirement

Req. Id	Requirement Description	
FNR.GUIR.5	Information sharing **MUST** provide a well-defined driver API framework that facilitates the integration of state-of-the-art IoT, edge/cloud computing, and networking platforms in its southbound interface, which allows for full programmability and reconfigurability of resources across the continuum	
	Affected components	SM-L, RM-L
	Contributing Partner	AGE/UBI
	Comment	//

Table 4-99: FNR.GUIR.5 requirement

Req. Id	Requirement Description	
FNR.GUIR.6	Information sharing **MUST** facilitate data exchange and interoperability through the provision of shared and standard data models.	
	Affected components	SM-L, RM-L
	Contributing Partner	FIWARE
	Comment	//

Table 4-100: FNR.GUIR.6 requirement

4.2.1.5 Generic Programming Requirements

The following tables provide functional requirements related to programming aspects of INCODE's IDP.

Req. Id	Requirement Description	
FNR.GPROG.1	The Internal Developer Platform (IDP) **MUST** allow Application Configuration Management	
	Affected components	IDP
	Contributing Partner	UBI/UWS
	Comment	Other interested partners in T4.4 to specify their contributions

Table 4-101: FNR.GPROG.1 requirement

Req. Id	Requirement Description	
FNR.GPROG.2	The Internal Developer Platform (IDP) **MUST** allow Infrastructure as Code	
	Affected components	IDP
	Contributing Partner	UWS/UBI
	Comment	Other interested partners in T4.4 to specify their contributions

Table 4-102: FNR.GPROG.2 requirement

Req. Id	Requirement Description
FNR.GPROG.3	The Internal Developer Platform (IDP) **MUST** allow Environment Management

Co-funded by the European Union

	Affected components	IDP
	Contributing Partner	UWS/UBI
	Comment	Other interested partners in T4.4 to specify their contributions

Table 4-103: FNR.GPROG.3 requirement

Req. Id	Requirement Description	
FNR.GPROG.4	The Internal Developer Platform (IDP) **MUST** allow Deployment Management	
	Affected components	IDP
	Contributing Partner	UWS/UBI
	Comment	Other interested partners in T4.4 to specify their contributions

Table 4-104: FNR.GPROG.4 requirement

Req. Id	Requirement Description	
FNR.GPROG.5	The Internal Developer Platform (IDP) **MUST** allow Role-Based Access Control	
	Affected components	IDP
	Contributing Partner	UWS/UBI
	Comment	Other interested partners in T4.4 to specify their contributions

Table 4-105: FNR.GPROG.5 requirement

Req. Id	Requirement Description	
FNR.GPROG.6	The Internal Developer Platform (IDP) **MUST** allow supporting developers by providing, logging and debugging mechanism	
	Affected components	IDP
	Contributing Partner	UWS/UBI
	Comment	Other interested partners in T4.4 to specify their contributions

Table 4-106: FNR.GPROG.6 requirement

Req. Id	Requirement Description	
FNR.GPROG.7	The Internal Developer Platform (IDP) **MUST** allow a Continuous Integration (CI) and Continuous Development (CD) methodology	
	Affected components	IDP
	Contributing Partner	UWS/UBI
	Comment	Other interested partners in T4.4 to specify their contributions

Table 4-107: FNR.GPROG.7 requirement

Req. Id	Requirement Description
FNR.GPROG.8	The Internal Developer Platform (IDP) **MUST** allow release management (e.g., A/B or canary deployments) through custom Kubernetes deployment strategies

Co-funded by the European Union

	Affected components	IDP
	Contributing Partner	UWS/UBI
	Comment	Other interested partners in T4.4 to specify their contributions

Table 4-108: FNR.GPROG.8 requirement

Req. Id	Requirement Description	
FNR.GPROG.9	The Internal Developer Platform (IDP) **MUST** support the integration of the state-of-the-art IDE plugin for assisting the developers for any actions during development of their INCODE enabled application	
	Affected components	IDP, IDE
	Contributing Partner	UWS/UBI
	Comment	Other interested partners in T4.4 to specify their contributions

Table 4-109: FNR.GPROG.9 requirement

Req. Id	Requirement Description	
FNR.GPROG.10	The Internal Developer Platform (IDP) **MUST** be integrated within the Runtime Orchestrator (RO) of SM-L to optimally adapt the code of multiple application components to the allocated heterogeneous hardware resources	
	Affected components	RO, Code accelerator (TornadoVM)
	Contributing Partner	UBI/MAN
	Comment	Other interested partners in T4.4 to specify their contributions

Table 4-110: FNR.GPROG.10 requirement

Req. Id	Requirement Description	
FNR.GPROG.11	The Internal Developer Platform (IDP) **COULD** be tailored to fit each use case's development and deployment needs	
	Affected components	IDP, IDE
	Contributing Partner	UWS/UBI
	Comment	Other interested partners in T4.4 to specify their contributions

Table 4-111: FNR.GPROG.11 requirement

4.2.1.6 Generic Security Requirements

The following tables provide generic functional requirements related to security.

Req. Id	Requirement Description	
FNR.GSEC.1	The Platform **MUST** be tolerable enough to survive failures and various attacks (resiliency).	
	Affected components	ALL
	Contributing Partner	ALL
	Comment	//

Table 4-112: FNR.GSEC.1 requirement

Co-funded by the European Union

Req. Id	Requirement Description	
FNR.GSEC.2	The Platform **COULD** be able to use tools to automate configuration and deployment to minimise human errors.	
	Affected components	ALL
	Contributing Partner	ALL
	Comment	//

Table 4-113: FNR.GSEC.2 requirement

Req. Id	Requirement Description	
FNR.GSEC.3	Connections among systems and equipment **COULD** be secured by authentication and encryption	
	Affected components	ALL
	Contributing Partner	ALL
	Comment	(Not within the prime scope of the project). Proposed to have an analysis of the system architecture in terms of supporting resiliency and added security measures at platform level

Table 4-114: FNR.GSEC.3 requirement

Req. Id	Requirement Description	
FNR.GSEC.4	The Resource Management Layer (RM-L) **MUST** implement suitable access-control mechanisms to mitigate the risk of attackers entering INCODE	
	Affected components	RM-L
	Contributing Partner	NEC
	Comment	//

Table 4-115: FNR.GSEC.4 requirement

Req. Id	Requirement Description	
FNR.GSEC.5	The Resource Management Layer (RM-L) **MUST** provide detection- and attack-mitigation mechanisms to identify adversarial attacks	
	Affected components	RM-L
	Contributing Partner	K3Y
	Comment	//

Table 4-116: FNR.GSEC.5 requirement

Req. Id	Requirement Description	
FNR.GSEC.6	The Resource Management Layer (RM-L) **MUST** ensure that IoT data is collected only from devices that have been registered to the INCODE platform	
	Affected components	RM-L
	Contributing Partner	NEC
	Comment	//

Table 4-117: FNR.GSEC.6 requirement

Co-funded by the European Union

4.2.1.7 Generic Data Processing Requirements and Regulations

The following tables provide generic functional requirements related to data processing.

Req. Id	Requirement Description	
FNR.GDAT.1	INCODE components and processes **MUST** follow the NIS directive	
	Affected components	ALL
	Contributing Partner	ALL
	Comment	//

Table 4-118: FNR.GDAT.1 requirement

Req. Id	Requirement Description	
FNR.GDAT.2	INCODE components and processes **MUST** follow schemes established under the Cybersecurity Act	
	Affected components	ALL
	Contributing Partner	ALL
	Comment	//

Table 4-119: FNR.GDAT.2 requirement

Req. Id	Requirement Description	
FNR.GDAT.3	The processing of personal data **MUST** be avoided in all cases	
	Affected components	ALL
	Contributing Partner	ALL
	Comment	//

Table 4-120: FNR.GDAT.3 requirement

Req. Id	Requirement Description	
FNR.GDAT.4	Processing of data **COULD** be aligned with the General Data Protection Regulation (EU) 2016/679 (GDPR)	
	Affected components	ALL
	Contributing Partner	ALL
	Comment	//

Table 4-121: FNR.GDAT.4 requirement

Req. Id	Requirement Description	
FNR.GDAT.5	The processed personal data, in case these are required for running a use case **COULD** be restricted only to minimum necessary for fulfilling identified purposes.	
	Affected components	ALL
	Contributing Partner	ALL
	Comment	//

Table 4-122: FNR.GDAT.5 requirement

Req. Id	Requirement Description

Co-funded by the European Union

FNR.GDAT.6	The storage of personal data by the platform, in case these are required for running a use case **COULD** be restricted to the minimum period necessary to fulfil the purpose of collection, as determined by each use case.	
	Affected components	ALL
	Contributing Partner	ALL
	Comment	//

Table 4-123: FNR.GDAT.6 requirement

Req. Id	Requirement Description	
FNR.GDAT.7	The platform modules **COULD** be unable to store personal data outside the jurisdiction of the EU/EEA, (except where adequate data protection framework exists in a non-EU/EEA jurisdiction according to the GDPR).	
	Affected components	ALL
	Contributing Partner	ALL
	Comment	//

Table 4-124: FNR.GDAT.7 requirement

4.2.2 Functional SM-L requirements

This section provides the complete list of functional requirements associated to the SM-L that need to be provided by the INCODE platform.

4.2.2.1 User Interfaces Functional Requirements

Req. Id	Requirement Description	
FNR.SML.UI.1	All UIs **MUST** support user authentication and access control to ensure that only authorized users can access the portal and perform specific actions	
	Affected components	Application provider UI, Application developer UI, Operations portal, Edge portal, Telemetry centre, Edge telemetry
	Contributing Partner	ALL
	Comment	//

Table 4-125: FNR.SML.UI.1 requirement

Req. Id	Requirement Description	
FNR.SML.UI.2	The application provider UI **MUST** provide a user-friendly interface for creating deployment workflows for applications that are to be deployed on the INCODE infrastructure	
	Affected components	Application provider UI
	Contributing Partner	ALL
	Comment	//

Table 4-126: FNR.SML.UI.2 requirement

Co-funded by the European Union

Req. Id	Requirement Description	
FNR.SML.UI.3	The application provider UI **MUST** allow the application providers to define security policies that will be used to ensure the security of the deployed applications	
	Affected components	Application provider UI
	Contributing Partner	ALL
	Comment	//

Table 4-127: FNR.SML.UI.3 requirement

Req. Id	Requirement Description	
FNR.SML.UI.4	The application provider UI **MUST** be integrated with the telemetry and runtime orchestration components to ensure that the deployed applications can be monitored and managed efficiently	
	Affected components	Application provider UI
	Contributing Partner	ALL
	Comment	//

Table 4-128: FNR.SML.UI.4 requirement

Req. Id	Requirement Description	
FNR.SML.UI.5	The application provider UI **MUST** provide capabilities for managing the application lifecycle, including application updates, versioning, and rollback	
	Affected components	Application provider UI
	Contributing Partner	ALL
	Comment	//

Table 4-129: FNR.SML.UI.5 requirement

Req. Id	Requirement Description	
FNR.SML.UI.6	The application provider UI **MUST** be integrated with continuous integration and continuous deployment (CI/CD) tools to enable the automatic building, testing, and deployment of applications	
	Affected components	Application provider UI
	Contributing Partner	ALL
	Comment	//

Table 4-130: FNR.SML.UI.6 requirement

Req. Id	Requirement Description	
FNR.SML.UI.7	The application provider UI **COULD** allow the application providers to define performance policies that will be used to ensure that the deployed applications meet the required performance criteria	
	Affected components	Application provider UI
	Contributing Partner	ALL
	Comment	//

Table 4-131: FNR.SML.UI.7 requirement

Co-funded by the European Union

Req. Id	Requirement Description	
FNR.SML.UI.8	The application provider UI **COULD** allow the application providers to define location-specific policies that will be used to determine the deployment location of the applications	
	Affected components	Application provider UI
	Contributing Partner	ALL
	Comment	//

Table 4-132: FNR.SML.UI.8 requirement

Req. Id	Requirement Description	
FNR.SML.UI.9	The application provider UI **COULD** provide a user-friendly interface that is easy to navigate and use, even for non-technical users	
	Affected components	Application provider UI
	Contributing Partner	ALL
	Comment	//

Table 4-133: FNR.SML.UI.9 requirement

Req. Id	Requirement Description	
FNR.SML.UI.10	The application provider UI **COULD** provide reporting and analytics capabilities to enable the application providers to track the performance of their deployed applications and make data-driven decisions	
	Affected components	Application provider UI
	Contributing Partner	ALL
	Comment	//

Table 4-134: FNR.SML.UI.10 requirement

Req. Id	Requirement Description	
FNR.SML.UI.11	The operations portal **MUST** provide real-time information about the various deployments	
	Affected components	Operations Portal
	Contributing Partner	ALL
	Comment	//

Table 4-135: FNR.SML.UI.11 requirement

Req. Id	Requirement Description	
FNR.SML.UI.12	The operations portal **MUST** provide alerting capabilities to notify users of any issues or outages	
	Affected components	Operations Portal
	Contributing Partner	ALL
	Comment	//

Table 4-136: FNR.SML.UI.12 requirement

Req. Id	Requirement Description
FNR.SML.UI.13	The operations portal **MUST** be able to handle many deployments and users, and be able to scale as the system grows

Co-funded by the European Union

	Affected components	Operations Portal
	Contributing Partner	ALL
	Comment	//

Table 4-137: FNR.SML.UI.13 requirement

Req. Id	Requirement Description	
FNR.SML.UI.14	The operations portal **COULD** allow users to customize the dashboard to fit their needs	
	Affected components	Operations Portal
	Contributing Partner	ALL
	Comment	//

Table 4-138: FNR.SML.UI.14 requirement

Req. Id	Requirement Description	
FNR.SML.UI.15	The operations portal **COULD** allow users to generate reports on system performance, uptime, and other metrics	
	Affected components	Operations Portal
	Contributing Partner	ALL
	Comment	//

Table 4-139: FNR.SML.UI.15 requirement

Req. Id	Requirement Description	
FNR.SML.UI.16	The operations portal **COULD** allow administrators to manage user access and permissions	
	Affected components	Operations Portal
	Contributing Partner	ALL
	Comment	//

Table 4-140: FNR.SML.UI.16 requirement

Req. Id	Requirement Description	
FNR.SML.UI.17	The operations portal **COULD** provide a search functionality to allow users to quickly find specific deployments	
	Affected components	Operations Portal
	Contributing Partner	ALL
	Comment	//

Table 4-141: FNR.SML.UI.17 requirement

Req. Id	Requirement Description	
FNR.SML.UI.18	The operations portal **COULD** allow users to filter deployments based on various criteria, such as application name or location	
	Affected components	Operations Portal
	Contributing Partner	ALL
	Comment	//

Table 4-142: FNR.SML.UI.18 requirement

Co-funded by the European Union

Req. Id	Requirement Description	
FNR.SML.UI.19	The operations portal **COULD** store historical data for deployments, so users can view past performance and compare it to current performance	
	Affected components	Operations Portal
	Contributing Partner	ALL
	Comment	//

Table 4-143: FNR.SML.UI.19 requirement

Req. Id	Requirement Description	
FNR.SML.UI.20	The operations portal **COULD** be able to integrate with other tools and systems, such as monitoring and logging tools	
	Affected components	Operations Portal
	Contributing Partner	ALL
	Comment	//

Table 4-144: FNR.SML.UI.20 requirement

Req. Id	Requirement Description	
FNR.SML.UI.21	The telemetry centre **MUST** display monitoring data in real-time, so that any issues or anomalies can be identified and addressed immediately	
	Affected components	Telemetry Centre
	Contributing Partner	ALL
	Comment	//

Table 4-145: FNR.SML.UI.21 requirement

Req. Id	Requirement Description	
FNR.SML.UI.22	The telemetry centre **MUST** be scalable to handle large volumes of monitoring data from multiple sources	
	Affected components	Telemetry Centre
	Contributing Partner	ALL
	Comment	//

Table 4-146: FNR.SML.UI.22 requirement

Req. Id	Requirement Description	
FNR.SML.UI.23	The telemetry centre **MUST** provide alerts and notifications when certain metrics or thresholds are exceeded, to enable proactive issue resolution.	
	Affected components	Telemetry Centre
	Contributing Partner	ALL
	Comment	//

Table 4-147: FNR.SML.UI.23 requirement

Req. Id	Requirement Description
FNR.SML.UI.24	The telemetry centre **MUST** allow users to view and analyse historical data to identify trends and patterns, and to make informed decisions based on this analysis

Co-funded by the European Union

	Affected components	Telemetry Centre
	Contributing Partner	ALL
	Comment	//

Table 4-148: FNR.SML.UI.24 requirement

Req. Id	Requirement Description	
FNR.SML.UI.25	The telemetry centre **MUST** ensure the security and privacy of monitoring data, to prevent unauthorized access or data breaches	
	Affected components	Telemetry Centre
	Contributing Partner	ALL
	Comment	//

Table 4-149: FNR.SML.UI.25 requirement

Req. Id	Requirement Description	
FNR.SML.UI.26	The telemetry centre **COULD** allow users to customize the monitoring dashboards and views to suit their specific needs and requirements	
	Affected components	Telemetry Centre
	Contributing Partner	ALL
	Comment	//

Table 4-150: FNR.SML.UI.26 requirement

Req. Id	Requirement Description	
FNR.SML.UI.27	The telemetry centre **COULD** integrate with other tools and systems used by the organization, such as ticketing systems, to enable seamless collaboration and issue resolution	
	Affected components	Telemetry Centre
	Contributing Partner	ALL
	Comment	//

Table 4-151: FNR.SML.UI.27 requirement

4.2.2.2 RBAC API Functional Requirements

Req. Id	Requirement Description	
FNR.SML.RBAC.1	All provided API calls **MUST** be accessible only by authenticated users	
	Affected components	SM-L RBAC API
	Contributing Partner	ALL
	Comment	//

Table 4-152: FNR.SML.RBAC.1 requirement

Req. Id	Requirement Description	
FNR.SML.RBAC.2	All provided API calls **MUST** be accessible only by authorized users	
	Affected components	SM-L RBAC API
	Contributing Partner	ALL

Co-funded by the European Union

	Comment	//

Table 4-153: FNR.SML.RBAC.2 requirement

Req. Id	Requirement Description	
FNR.SML.RBAC.3	SM-L APIs **MUST** support roles management	
	Affected components	SM-L RBAC API
	Contributing Partner	ALL
	Comment	//

Table 4-154: FNR.SML.RBAC.3 requirement

Req. Id	Requirement Description	
FNR.SML.RBAC.4	SM-L APIs **MUST** support user management	
	Affected components	SM-L RBAC API
	Contributing Partner	ALL
	Comment	//

Table 4-155: FNR.SML.RBAC.4 requirement

Req. Id	Requirement Description	
FNR.SML.RBAC.5	SM-L APIs **MUST** support ACLs	
	Affected components	SM-L RBAC API
	Contributing Partner	ALL
	Comment	//

Table 4-156: FNR.SML.RBAC.5 requirement

Req. Id	Requirement Description	
FNR.SML.RBAC.6	SM-L APIs **MUST** support auditing and logging	
	Affected components	SM-L RBAC API
	Contributing Partner	ALL
	Comment	//

Table 4-157: FNR.SML.RBAC.6 requirement

Req. Id	Requirement Description	
FNR.SML.RBAC.7	SM-L APIs **MUST** integrate seamlessly with the IDP	
	Affected components	SM-L RBAC API
	Contributing Partner	ALL
	Comment	//

Table 4-158: FNR.SML.RBAC.7 requirement

Req. Id	Requirement Description	
FNR.SML.RBAC.8	SM-L APIs **MUST** be extensible and customizable	
	Affected components	SM-L RBAC API
	Contributing Partner	ALL
	Comment	//

Co-funded by the European Union

Table 4-159: FNR.SML.RBAC.8 requirement

Req. Id	Requirement Description	
FNR.SML.RBAC.9	SM-L APIs **MUST** be well-documented (e.g., OpenAPI)	
	Affected components	SM-L RBAC API
	Contributing Partner	ALL
	Comment	//

Table 4-160: FNR.SML.RBAC.9 requirement

4.2.2.3 Service Orchestrator Functional Requirements

Req. Id	Requirement Description	
FNR.SML.SO.1	The orchestrator **MUST** be able to request resources from RM-L	
	Affected components	SM-L Runtime Orchestration
	Contributing Partner	ALL
	Comment	//

Table 4-161: FNR.SML.SO.1 requirement

Req. Id	Requirement Description	
FNR.SML.SO.2	The orchestrator **MUST** be able to create and store the application graph	
	Affected components	SM-L Runtime Orchestration
	Contributing Partner	ALL
	Comment	//

Table 4-162: FNR.SML.SO.2 requirement

Req. Id	Requirement Description	
FNR.SML.SO.3	The orchestrator **MUST** be able to deploy the application graph	
	Affected components	SM-L Runtime Orchestration
	Contributing Partner	ALL
	Comment	//

Table 4-163: FNR.SML.SO.3 requirement

Req. Id	Requirement Description	
FNR.SML.SO.4	The orchestrator **MUST** be able to undeploy/delete the application graph	
	Affected components	SM-L Runtime Orchestration
	Contributing Partner	ALL
	Comment	//

Table 4-164: FNR.SML.SO.4 requirement

Req. Id	Requirement Description	
FNR.SML.SO.8	The resource management algorithm **MUST** be able to read relevant data/logs from the INCODE Telemetry	
	Affected components	SM-L Runtime Orchestration
	Contributing Partner	ALL

Co-funded by the European Union

	Comment	//

Table 4-165: FNR.SML.SO.8 requirement

Req. Id	Requirement Description	
FNR.SML.SO.9	The resource management algorithm **MUST** be able to trigger the NBI of the RO to enforce a decision to the infrastructure	
	Affected components	SM-L Runtime Orchestration
	Contributing Partner	ALL
	Comment	//

Table 4-166: FNR.SML.SO.9 requirement

Req. Id	Requirement Description	
FNR.SML.SO.10	The policy engine **MUST** be able to make decisions using the locally available storage	
	Affected components	SM-L Policy Engine, SM-L Telemetry
	Contributing Partner	ALL
	Comment	//

Table 4-167: FNR.SML.SO.10 requirement

Req. Id	Requirement Description	
FNR.SML.SO.12	End-users or service providers or vertical industry operators **MUST** use the policy engine API to create/update/delete relevant policies for their services	
	Affected components	SM-L Policy Engine
	Contributing Partner	ALL
	Comment	//

Table 4-168: FNR.SML.SO.12 requirement

Req. Id	Requirement Description	
FNR.SML.SO.5	The orchestrator **COULD** receive feedback from a specialized profiler to deploy each application among the heterogeneous devices in an optimal manner.	
	Affected components	SM-L Runtime Orchestration
	Contributing Partner	ALL
	Comment	//

Table 4-169: FNR.SML.SO.5 requirement

Req. Id	Requirement Description	
FNR.SML.SO.6	The orchestrator **COULD** view the providers infrastructure	
	Affected components	SM-L Runtime Orchestration
	Contributing Partner	ALL
	Comment	//

Table 4-170: FNR.SML.SO.6 requirement

Req. Id	Requirement Description

Co-funded by the European Union

FNR.SML.SO.7	The orchestrator **COULD** provide affinity/anti affinity rules for the application's components	
	Affected components	SM-L Runtime Orchestration
	Contributing Partner	ALL
	Comment	//

Table 4-171: FNR.SML.SO.7 requirement

Req. Id	Requirement Description	
FNR.SML.SO.11	The policy engine **COULD** use a centralized pool of rules	
	Affected components	SM-L Policy Engine
	Contributing Partner	ALL
	Comment	//

Table 4-172: FNR.SML.SO.11 requirement

4.2.2.4 Telemetry Functional Requirements

Req. Id	Requirement Description	
FNR.SML.TM.1	The status of the distributed applications' component **MUST** be monitored	
	Affected components	SM-L Telemetry
	Contributing Partner	ALL
	Comment	//

Table 4-173: FNR.SML.TM.1 requirement

Req. Id	Requirement Description	
FNR.SML.TM.2	Metrics per component (CPU, RAM) **MUST** allow access only to the component owner	
	Affected components	SM-L Telemetry
	Contributing Partner	ALL
	Comment	//

Table 4-174: FNR.SML.TM.2 requirement

Req. Id	Requirement Description	
FNR.SML.TM.3	The provider specific metrics **MUST** be accessible only by those who can see and deploy to this provider	
	Affected components	SM-L Telemetry
	Contributing Partner	ALL
	Comment	//

Table 4-175: FNR.SML.TM.3 requirement

Req. Id	Requirement Description	
FNR.SML.TM.4	Logs per component **COULD** allow access only to the component owner	
	Affected components	SM-L Telemetry
	Contributing Partner	ALL
	Comment	//

Co-funded by the European Union

Table 4-176: FNR.SML.TM.3 requirement

Req. Id	Requirement Description	
FNR.SML.TM.5	Custom metrics per component **COULD** be accessed only by the component owner	
	Affected components	SM-L Telemetry
	Contributing Partner	ALL
	Comment	//

Table 4-177: FNR.SML.TM.5 requirement

Req. Id	Requirement Description	
FNR.SML.TM.6	Telemetry component metrics **COULD** be accessed from another controller that manage the component	
	Affected components	SM-L Telemetry
	Contributing Partner	ALL
	Comment	//

Table 4-178: FNR.SML.TM.6 requirement

Req. Id	Requirement Description	
FNR.SML.TM.7	Telemetry provider metrics **COULD** be accessed from other controllers that manage the provider	
	Affected components	SM-L Telemetry
	Contributing Partner	ALL
	Comment	//

Table 4-179: FNR.SML.TM.7 requirement

4.2.2.5 Software Lifecycle Functional Requirements

Req. Id	Requirement Description	
FNR.SML.SL.1	DevOps **MUST** be able to manage the deployed application	
	Affected components	SM-L Software Lifecycle
	Contributing Partner	ALL
	Comment	//

Table 4-180: FNR.SML.SL.1 requirement

Req. Id	Requirement Description	
FNR.SML.SL.2	SM-L **MUST** use git repositories to store application and infrastructure configuration	
	Affected components	SM-L Software Lifecycle
	Contributing Partner	ALL
	Comment	//

Table 4-181: FNR.SML.SL.2 requirement

Req. Id	Requirement Description
FNR.SML.SL.3	SM-L **MUST** offer a CI/CD based lifecycle

Co-funded by the European Union

	Affected components	SM-L Software Lifecycle
	Contributing Partner	ALL
	Comment	//

Table 4-182: FNR.SML.SL.3 requirement

Req. Id	Requirement Description	
FNR.SML.SL.4	Component management **COULD** update components with new versions	
	Affected components	SM-L Software Lifecycle
	Contributing Partner	ALL
	Comment	//

Table 4-183: FNR.SML.SL.4 requirement

Req. Id	Requirement Description	
FNR.SML.SL.5	Security **COULD** provide vulnerabilities scanning for components	
	Affected components	SM-L Software Lifecycle
	Contributing Partner	ALL
	Comment	//

Table 4-184: FNR.SML.SL.5 requirement

Req. Id	Requirement Description	
FNR.SML.SL.6	Software Quality (e.g., sonar) **COULD** provide code base analysis	
	Affected components	SM-L Software Lifecycle
	Contributing Partner	ALL
	Comment	//

Table 4-185: FNR.SML.SL.6 requirement

4.2.2.6 Internal Developer Platform Functional Requirements

Req. Id	Requirement Description	
FNR.RML.IDP.1	IDE plugin COULD be offered to allow easier usage of the platform by developers	
	Affected components	IDP and Software Lifecycle Management
	Contributing Partner	UBITECH
	Comment	//

Table 4-186: FNR.RML.IDP.1 requirement

Req. Id	Requirement Description	
FNR.RML.IDP.2	The use cases that will be offloaded to the IDE plugin as candidates for acceleration MUST be written using the Java programming language	
	Affected components	IDP and Software Lifecycle Management
	Contributing Partner	UMAN and use case leaders and contributors
	Comment	//

Table 4-187: FNR.RML.IDP.2 requirement

Co-funded by the European Union

Req. Id	Requirement Description	
FNR.RML.IDP.3	The use cases that will be offloaded to the IDE plugin as candidates for acceleration MUST have a high volume of input data	
	Affected components	IDP and Software Lifecycle Management
	Contributing Partner	UMAN and use case leaders and contributors
	Comment	//

Table 4-188: FNR.RML.IDP.3 requirement

Req. Id	Requirement Description	
FNR.RML.IDP.4	The use cases that will be offloaded to the IDE plugin as candidates for acceleration MUST avoid having data dependencies in their executable code.	
	Affected components	IDP and Software Lifecycle Management
	Contributing Partner	UMAN and use case leaders and contributors
	Comment	//

Table 4-189: FNR.RML.IDP.4 requirement

Req. Id	Requirement Description	
FNR.RML.IDP.5	The IDP MUST provide a comprehensive set of developer tools (code editor, version control, build tools)	
	Affected components	IDP and Software Lifecycle Management
	Contributing Partner	UWS, UBITECH, SIEMENS, UMAN
	Comment	//

Table 4-190: FNR.RML.IDP.5 requirement

Req. Id	Requirement Description	
FNR.RML.IDP.6	The software lifecycle manager MUST provide SDK and APIs	
	Affected components	IDP and Software Lifecycle Management
	Contributing Partner	SIEMENS
	Comment	//

Table 4-191: FNR.RML.IDP.6 requirement

Req. Id	Requirement Description	
FNR.RML.IDP.7	The IDP MUST support DevOps tools and processes (CI/CD)	
	Affected components	IDP and Software Lifecycle Management
	Contributing Partner	SIEMENS, UWS
	Comment	//

Table 4-192: FNR.RML.IDP.7 requirement

Req. Id	Requirement Description	
FNR.RML.IDP.8	The IDE COULD support collaborative coding environments, chat or messaging, documentation sharing	
	Affected components	IDP and Software Lifecycle Management
	Contributing Partner	UWS
	Comment	//

Table 4-193: FNR.RML.IDP.8 requirement

Co-funded by the European Union

Req. Id	Requirement Description	
FNR.RML.IDP.9	The software lifecycle manager MUST provide sandbox and staging environments	
	Affected components	IDP and Software Lifecycle Management
	Contributing Partner	SIEMENS
	Comment	//

Table 4-194: FNR.RML.IDP.9 requirement

Req. Id	Requirement Description	
FNR.RML.IDP.10	The software lifecycle manager MUST support monitoring and analytics capabilities	
	Affected components	IDP and Software Lifecycle Management
	Contributing Partner	SIEMENS
	Comment	//

Table 4-195: FNR.RML.IDP.10 requirement

Req. Id	Requirement Description	
FNR.RML.IDP.11	The software lifecycle manager MUST be scalable and capable of handling a growing number of developers, applications, workloads	
	Affected components	IDP and Software Lifecycle Management
	Contributing Partner	SIEMENS
	Comment	//

Table 4-196: FNR.RML.IDP.11 requirement

Req. Id	Requirement Description	
FNR.RML.IDP.12	The software lifecycle manager MUST support integration with external services (bug tracking, project management)	
	Affected components	IDP and Software Lifecycle Management
	Contributing Partner	SIEMENS
	Comment	//

Table 4-197: FNR.RML.IDP.12 requirement

Req. Id	Requirement Description	
FNR.RML.IDP.13	The software lifecycle manager MUST implement robust security measures (authentication & authorization)	
	Affected components	IDP and Software Lifecycle Management
	Contributing Partner	AGENT SCAPE
	Comment	//

Table 4-198: FNR.RML.IDP.13 requirement

4.2.3 Functional RM-L requirements

This section provides the complete list of functional requirements associated to the RM-L that need to be provided by the INCODE platform.

Co-funded by the European Union

4.2.3.1 Scheduler Functional Requirements

Req. Id		Requirement Description	
FNR.RML.SCH.1		SC **MUST** be able to schedule tasks across clusters.	
		Affected components	Scheduler
		Contributing Partner	RedHat
		Comment	//

Table 4-199: FNR.RML.SCH.1 requirement

Req. Id	Requirement Description	
FNR.RML.SCH.2	SC **MUST** support scheduling algorithms that consider special hardware (e.g., IoT devices, GPUs)	
	Affected components	Scheduler
	Contributing Partner	RedHat
	Comment	//

Table 4-200: FNR.RML.SCH.2 requirement

Req. Id	Requirement Description	
FNR.RML.SCH.3	SC **MUST** be able to select specific cluster and node	
	Affected components	Scheduler
	Contributing Partner	RedHat
	Comment	//

Table 4-201: FNR.RML.SCH.3 requirement

Req. Id	Requirement Description	
FNR.RML.SCH.4	SC **MUST** be able to work with different compute orchestration platform variants (e.g., Openshift, microshift, K8s, microk8s)	
	Affected components	Scheduler
	Contributing Partner	RedHat
	Comment	//

Table 4-202: FNR.RML.SCH.4 requirement

Req. Id	Requirement Description	
FNR.RML.SCH.5	SC **MUST** be able to synchronize information between clusters and node	
	Affected components	Scheduler
	Contributing Partner	FIWARE
	Comment	//

Table 4-203: FNR.RML.SCH.5 requirement

4.2.3.2 Resource Manager Functional Requirements

Req. Id	Requirement Description
FNR.RML.RM.1	Access to Infrastructure (SBI) **MUST** have access to the underlying infrastructure it will be affecting (RAN, Transport, Cloud, etc.)

Co-funded by the European Union

	Affected components	Resource Manager
	Contributing Partner	UoP
	Comment	//

Table 4-204: FNR.RML.RM.1 requirement

Req. Id	Requirement Description	
FNR.RML.RM.2	Prometheus Telemetry **MUST** consume telemetry in Prometheus format	
	Affected components	Resource Manager
	Contributing Partner	UoP
	Comment	//

Table 4-205: FNR.RML.RM.2 requirement

Req. Id	Requirement Description	
FNR.RML.RM.3	Expose NBI **MUST** provide Northbound APIs to overlay components	
	Affected components	Resource Manager
	Contributing Partner	UoP
	Comment	//

Table 4-206: FNR.RML.RM.3 requirement

Req. Id	Requirement Description	
FNR.RML.RM.4	Standardized management interfaces **MUST** communicate through open, standardized APIs	
	Affected components	Resource Manager
	Contributing Partner	UoP, FIWARE
	Comment	//

Table 4-207: FNR.RML.RM.4 requirement

Req. Id	Requirement Description	
FNR.RML.RM.5	Infrastructure interfaces **MUST** implement domain-specific interfaces for compute and network resource management	
	Affected components	Resource Manager
	Contributing Partner	UoP
	Comment	//

Table 4-208: FNR.RML.RM.5 requirement

Req. Id	Requirement Description	
FNR.RML.RM.6	Northbound connectivity **MUST** be accessible	
	Affected components	Resource Manager
	Contributing Partner	UoP
	Comment	//

Table 4-209: FNR.RML.RM.6 requirement

Co-funded by the European Union

Req. Id	Requirement Description	
FNR.RML.RM.7	The specialized edge processors **MUST** have the necessary components installed (e.g., OpenCL drivers) for hardware acceleration	
	Affected components	Resource Manager
	Contributing Partner	UMAN
	Comment	//

Table 4-210: FNR.RML.RM.7 requirement

Req. Id	Requirement Description	
FNR.RML.RM.8	The resource management algorithm **MUST** be able to store the outcomes of the smart resource allocation back to the INCODE storage subsystem	
	Affected components	Resource Manager
	Contributing Partner	Axon
	Comment	//

Table 4-211: FNR.RML.RM.8 requirement

Req. Id	Requirement Description	
FNR.RML.RM.9	The resource manager **MUST** provide federation capabilities to facilitate the automatic configuration of the devices	
	Affected components	Resource Manager
	Contributing Partner	FIWARE
	Comment	//

Table 4-212: FNR.RML.RM.9 requirement

Req. Id	Requirement Description	
FNR.RML.RM.10	Access to make changes to the infrastructure **COULD** be able to make changes to the domain's infrastructure	
	Affected components	Resource Manager
	Contributing Partner	UoP
	Comment	//

Table 4-213: FNR.RML.RM.10 requirement

Req. Id	Requirement Description	
FNR.RML.RM.11	RM **MUST** provide distributed storage primitives through unified models for data exchange	
	Affected components	Distributed Storage
	Contributing Partner	ARHS, FIWARE
	Comment	Archive data

Table 4-214: FNR.RML.RM.11 requirement

Req. Id	Requirement Description	
FNR.RML.RM.12	RM **MUST** support distributed storage through JSON-based data models for Ultra- Wide-Band devices	
	Affected components	Distributed Storage
	Contributing Partner	iLink, FIWARE

Co-funded by the European Union

	Comment	Input data consumed by sensors

Table 4-215: FNR.RML.RM.12 requirement

4.2.3.3 IoT Controller Functional Requirements

Req. Id	Requirement Description	
FNR.RML.CTL.IOT.1	IoT Controller **MUST** provide a transport protocol to access the data using HTTP or MQTT	
	Affected components	IoT Controller
	Contributing Partner	FIWARE
	Comment	//

Table 4-216: FNR.RML.CTL.IOT.1 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.IOT.2	IoT Controller **MUST** provide a standard open API protocol to access the data (e.g., ETSI NGSI-LD)	
	Affected components	IoT Controller
	Contributing Partner	FIWARE
	Comment	//

Table 4-217: FNR.RML.CTL.IOT.2 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.IOT.3	IoT Controller **MUST** use harmonized representation of the data defined through data models and using well-known ontologies	
	Affected components	IoT Controller
	Contributing Partner	FIWARE
	Comment	//

Table 4-218: FNR.RML.CTL.IOT.3 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.IOT.4	IoT Controller **MUST** provide data represented in JSON-LD payload format	
	Affected components	IoT Controller
	Contributing Partner	FIWARE
	Comment	//

Table 4-219: FNR.RML.CTL.IOT.4 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.IOT.5	IoT Controller **MUST** provide mechanism to authenticate and authorise based on token or JWT defined in the header of the request	
	Affected components	IoT Controller
	Contributing Partner	FIWARE
	Comment	//

Table 4-220: FNR.RML.CTL.IOT.5 requirement

Co-funded by the European Union

Req. Id	Requirement Description	
FNR.RML.CTL.IOT.6	IoT Controller **MUST** Implement public-subscribe mechanism to allow sending the data to the services that requires it	
	Affected components	IoT Controller
	Contributing Partner	FIWARE
	Comment	//

Table 4-221: FNR.RML.CTL.IOT.6 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.IOT.7	IoT Controller **MUST** persist the last received data into a DB to recover it in case that the IoT-C is going down	
	Affected components	IoT Controller
	Contributing Partner	FIWARE
	Comment	//

Table 4-222: FNR.RML.CTL.IOT.7 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.IOT.8	IoT Controller **MUST** provide federation functionalities to relate to the Resource Manager and facilitate the configuration of the IoT-C and IoT-D as well as facilitate the exchange and search of information between the Resource Manager and the IoT Devices.	
	Affected components	IoT Controller
	Contributing Partner	FIWARE
	Comment	//

Table 4-223: FNR.RML.CTL.IOT.8 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.IOT.9	IoT Controller **COULD** provide federation capabilities to facilitate the synchronization of data between the Cloud Node and the Edge Nodes in collaboration with the Distributed Storage	
	Affected components	IoT Controller
	Contributing Partner	FIWARE
	Comment	//

Table 4-224: FNR.RML.CTL.IOT.9 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.IOT.10	IoT Controller **COULD** provide a temporal operations API to persist all the received data into TimeSeriesDB	
	Affected components	IoT Controller
	Contributing Partner	FIWARE
	Comment	//

Table 4-225: FNR.RML.CTL.IOT.10 requirement

Co-funded by the European Union

Req. Id	Requirement Description	
FNR.RML.CTL.IOT.11	IoT Controller **COULD** provide an architecture of High Availability to prevent the losing of data in case of one broker is down	
	Affected components	IoT Controller
	Contributing Partner	FIWARE
	Comment	//

Table 4-226: FNR.RML.CTL.IOT.11 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.IOT.12	IoT Controller **COULD** support mesh and ad-hoc networking capabilities, enabling dynamic groups of IoT devices to form swarms and collaborate in a decentralized manner.	
	Affected components	IoT Controller
	Contributing Partner	K3Y
	Comment	//

Table 4-227: FNR.RML.CTL.IOT.12 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.IOT.13	IoT Controller **COULD** have event handling and alerting mechanisms to detect and respond to critical events or abnormal behaviour from IoT devices. It should enable administrators to define thresholds, set up notifications, and trigger automated actions when specific events occur.	
	Affected components	IoT Controller
	Contributing Partner	K3Y
	Comment	//

Table 4-228: FNR.RML.CTL.IOT.13 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.IOT.14	IoT Controller **COULD** have data analytics capabilities to process and analyse the collected IoT data. It should provide insights and visualizations to enable administrators and users to derive meaningful information and make informed decisions based on the IoT data.	
	Affected components	IoT Controller
	Contributing Partner	K3Y
	Comment	//

Table 4-229: FNR.RML.CTL.IOT.14 requirement

4.2.3.4 IoT Driver Functional Requirements

Req. Id	Requirement Description	
FNR.RML.DRV.IOT.1	Transport Protocol support (Southbound) **MUST** support a wide range of communication protocols such as OPC-UA, MQTT, HTTP, SigFox	
	Affected components	IoT Driver
	Contributing Partner	POLIMI, FIWARE

Co-funded by the European Union

	Comment	//

Table 4-230: FNR.RML.DRV.IOT.1 requirement

Req. Id	Requirement Description	
FNR.RML.DRV.IOT.2	Payload format (Southbound) **MUST** support common payload formats such as JSON, UL, or CSV	
	Affected components	IoT Driver
	Contributing Partner	FIWARE
	Comment	//

Table 4-231: FNR.RML.DRV.IOT.2 requirement

Req. Id	Requirement Description	
FNR.RML.DRV.IOT.3	Northbound connectivity **MUST** provide connectivity to the IoT -C through ETSI NGSI-LD API	
	Affected components	IoT Driver
	Contributing Partner	FIWARE
	Comment	//

Table 4-232: FNR.RML.DRV.IOT.3 requirement

Req. Id	Requirement Description	
FNR.RML.DRV.IOT.4	IoT Driver **MUST** provide an API in order to preconfigure the connectivity with the IoT-C	
	Affected components	IoT Driver
	Contributing Partner	FIWARE
	Comment	//

Table 4-233: FNR.RML.DRV.IOT.4 requirement

Req. Id	Requirement Description	
FNR.RML.DRV.IOT.5	IoT Driver **MUST** provide an API to allow the management of actuations received from the IoT-C	
	Affected components	IoT Driver
	Contributing Partner	FIWARE
	Comment	//

Table 4-234: FNR.RML.DRV.IOT.5 requirement

Req. Id	Requirement Description	
FNR.RML.DRV.IOT.6	IoT Driver payload information **MUST** use harmonized representation of data defined through data models and using well-known ontologies.	
	Affected components	IoT Driver
	Contributing Partner	FIWARE
	Comment	//

Table 4-235: FNR.RML.DRV.IOT.6 requirement

Co-funded by the European Union

Req. Id	Requirement Description	
FNR.RML.DRV.IOT.7	IoT Driver **MUST** be able to collect data from IoT devices in real-time or at scheduled intervals.	
	Affected components	IoT Driver
	Contributing Partner	K3Y
	Comment	//

Table 4-236: FNR.RML.DRV.IOT.7 requirement

Req. Id	Requirement Description	
FNR.RML.DRV.IOT.8	IoT Driver **MUST** be capable of discovering and identifying IoT devices within the network automatically.	
	Affected components	IoT Driver
	Contributing Partner	K3Y
	Comment	//

Table 4-237: FNR.RML.DRV.IOT.8 requirement

Req. Id	Requirement Description	
FNR.RML.DRV.IOT.9	IoT Driver **MUST** provide functionalities to manage IoT devices efficiently in terms software updates, upgrades and remote reboot	
	Affected components	IoT Driver
	Contributing Partner	K3Y
	Comment	//

Table 4-238: FNR.RML.DRV.IOT.9 requirement

Req. Id	Requirement Description	
FNR.RML.DRV.IOT.10	IoT Driver **MUST** support GPS sensor and GPS telematic devices	
	Affected components	IoT Driver
	Contributing Partner	iLink
	Comment	//

Table 4-239: FNR.RML.DRV.IOT.10 requirement

Req. Id	Requirement Description	
FNR.RML.DRV.IOT.11	IoT Driver **COULD** provide a mechanism to resend data in case that the communication with IoT-C is not possible to contact	
	Affected components	IoT Driver
	Contributing Partner	FIWARE
	Comment	//

Table 4-240: FNR.RML.DRV.IOT.11 requirement

4.2.3.5 RAN Controller Functional Requirements

Req. Id	Requirement Description
FNR.RML.CTL.RAN.1	RAN controllers **MUST** support multiple RAN splits, depending on the use case

Co-funded by the European Union

	Affected components	RAN Controller
	Contributing Partner	UBI
	Comment	//

Table 4-241: FNR.RML.CTL.RAN.1 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.RAN.2	O-RAN-C **MUST** support O-RAN split 7.2 among other splits	
	Affected components	RAN Controller
	Contributing Partner	UBI
	Comment	//

Table 4-242: FNR.RML.CTL.RAN.2 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.RAN.3	O-RAN-C **MUST** implement an E2 interface towards O-DUs and O-CUs	
	Affected components	RAN Controller
	Contributing Partner	UBI
	Comment	//

Table 4-243: FNR.RML.CTL.RAN.3 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.RAN.4	O-RAN-C **MUST** implement an O1 interface towards O-RUs, O-Dus, and O-CUs	
	Affected components	RAN Controller
	Contributing Partner	UBI
	Comment	//

Table 4-244: FNR.RML.CTL.RAN.4 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.RAN.5	O-RAN-C **MUST** implement an O2 interface towards a cloud system for managing the deployments of the software-based RAN NFs	
	Affected components	RAN Controller
	Contributing Partner	UBI
	Comment	//

Table 4-245: FNR.RML.CTL.RAN.5 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.RAN.6	O-RAN-C **MUST** provide APIs for radio resource management (RRM)	
	Affected components	RAN Controller
	Contributing Partner	UBI
	Comment	//

Table 4-246: FNR.RML.CTL.RAN.6 requirement

Req. Id	Requirement Description
FNR.RML.CTL.RAN.7	O-RAN-C **COULD** implement RRM as an xApp/rApp

Co-funded by the European Union

	Affected components	RAN Controller
	Contributing Partner	UBI
	Comment	//

Table 4-247: FNR.RML.CTL.RAN.7 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.RAN.8	O-RAN-C **COULD** provide RAN telemetry as a service to overlay components	
	Affected components	RAN Controller
	Contributing Partner	UBI
	Comment	//

Table 4-248: FNR.RML.CTL.RAN.8 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.RAN.9	O-RAN-C **COULD** Implement R1 interface towards r-APPs	
	Affected components	RAN Controller
	Contributing Partner	UoP
	Comment	//

Table 4-249: FNR.RML.CTL.RAN.9 requirement

4.2.3.6 RAN Driver Functional Requirements

Req. Id	Requirement Description	
FNR.RML.DRV.RAN.1	RAN drivers **MUST** support both legacy radio units (RUs, e.g., Amarisoft) and open RUs (O-RUs)	
	Affected components	RAN Driver
	Contributing Partner	UBI
	Comment	//

Table 4-250: FNR.RML.DRV.RAN.1 requirement

Req. Id	Requirement Description	
FNR.RML.DRV.RAN.2	O-RAN-D **MUST** be able to deploy and manage open RUs	
	Affected components	RAN Driver
	Contributing Partner	UBI
	Comment	//

Table 4-251: FNR.RML.DRV.RAN.2 requirement

Req. Id	Requirement Description	
FNR.RML.DRV.RAN.3	O-RAN-D **MUST** be able to deploy and manage software-based distributed units (O-DUs) on commodity hardware	
	Affected components	RAN Driver
	Contributing Partner	UBI
	Comment	//

Table 4-252: FNR.RML.DRV.RAN.3 requirement

Co-funded by the European Union

Req. Id	Requirement Description	
FNR.RML.DRV.RAN.4	O-RAN-D **MUST** be able to deploy and manage software-based user plane central units (O-CU-U) on commodity hardware	
	Affected components	RAN Driver
	Contributing Partner	UBI
	Comment	//

Table 4-253: FNR.RML.DRV.RAN.4 requirement

Req. Id	Requirement Description	
FNR.RML.DRV.RAN.5	O-RAN-D **MUST** be able to deploy and manage software-based control plane central units (O-CU-C) on commodity hardware	
	Affected components	RAN Driver
	Contributing Partner	UBI
	Comment	//

Table 4-254: FNR.RML.DRV.RAN.5 requirement

Req. Id	Requirement Description	
FNR.RML.DRV.RAN.6	O-RAN-D **MUST** implement an F1-C interface between O-DU and O-CU-C	
	Affected components	RAN Driver
	Contributing Partner	UBI
	Comment	//

Table 4-255: FNR.RML.DRV.RAN.6 requirement

Req. Id	Requirement Description	
FNR.RML.DRV.RAN.7	O-RAN-D **MUST** implement an F1-U interface between O-DU and O-CU-U	
	Affected components	RAN Driver
	Contributing Partner	UBI
	Comment	//

Table 4-256: FNR.RML.DRV.RAN.7 requirement

Req. Id	Requirement Description	
FNR.RML.DRV.RAN.8	O-RAN-D **MUST** implement open fronthaul Control User Synchronization (CUS) and Management interfaces between O-DU and O-RU	
	Affected components	RAN Driver
	Contributing Partner	UBI
	Comment	//

Table 4-257: FNR.RML.DRV.RAN.8 requirement

Req. Id	Requirement Description	
FNR.RML.DRV.RAN.9	O-RAN-D **MUST** implement an E2 interface between O-DU/O-CU and the RAN-C	
	Affected components	RAN Driver
	Contributing Partner	UBI
	Comment	//

Co-funded by the European Union

Table 4-258: FNR.RML.DRV.RAN.9 requirement

Req. Id	Requirement Description	
FNR.RML.DRV.RAN.10	O-RAN-D **MUST** implement an O1 interface between O-RU/O-DU/O-CU and the RAN-C	
	Affected components	RAN Driver
	Contributing Partner	UBI
	Comment	//

Table 4-259: FNR.RML.DRV.RAN.10 requirement

Req. Id	Requirement Description	
FNR.RML.DRV.RAN.11	O-RAN-D **MUST** implement an N2 interface between O-CU-C and the Mob-C	
	Affected components	RAN Driver
	Contributing Partner	UBI
	Comment	//

Table 4-260: FNR.RML.DRV.RAN.11 requirement

Req. Id	Requirement Description	
FNR.RML.DRV.RAN.12	O-RAN-D **MUST** implement an N3 interface between O-CU-U and the Mob-C	
	Affected components	RAN Driver
	Contributing Partner	UBI
	Comment	//

Table 4-261: FNR.RML.DRV.RAN.12 requirement

Req. Id	Requirement Description	
FNR.RML.DRV.RAN.13	O-RAN-D **MUST** implement an E1 interface between O-CU-U and O-CU-C	
	Affected components	RAN Driver
	Contributing Partner	UBI
	Comment	//

Table 4-262: FNR.RML.DRV.RAN.13 requirement

4.2.3.7 Network Fabric Controller Functional Requirements

Req. Id	Requirement Description	
FNR.RML.CTL.NF.1	Intra-DC Net-C **MUST** provide a technology-agnostic API to create L2 connectivity services between any two endpoints within a datacenter	
	Affected components	Network Fabric Controller
	Contributing Partner	UBI
	Comment	//

Table 4-263: FNR.RML.CTL.NF.1 requirement

Co-funded by the European Union

Req. Id	Requirement Description	
FNR.RML.CTL.NF.2	Intra-DC Net-C **MUST** translate connectivity service requests into target-specific configuration based on the available network drivers	
	Affected components	Network Fabric Controller
	Contributing Partner	UBI
	Comment	//

Table 4-264: FNR.RML.CTL.NF.2 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.NF.3	Intra-DC Net-C **MUST** implement one or more SDN protocols for managing programmable targets (i.e., hardware/software switches, SmartNICs)	
	Affected components	Network Fabric Controller
	Contributing Partner	UBI
	Comment	//

Table 4-265: FNR.RML.CTL.NF.3 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.NF.4	Intra-DC Net-C **MUST** implement the P4Runtime SDN protocol for managing P4 targets (i.e., hardware/software switches, SmartNICs)	
	Affected components	Network Fabric Controller
	Contributing Partner	UBI
	Comment	//

Table 4-266: FNR.RML.CTL.NF.4 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.NF.5	Intra-DC Net-C **MUST** be able to update the forwarding tables (i.e., insert/update/delete entries) of the underlying devices on the fly	
	Affected components	Network Fabric Controller
	Contributing Partner	UBI
	Comment	//

Table 4-267: FNR.RML.CTL.NF.5 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.NF.6	Intra-DC Net-C **COULD** be able to update the forwarding pipeline (e.g., the P4 program) of the underlying devices on the fly	
	Affected components	Network Fabric Controller
	Contributing Partner	UBI
	Comment	//

Table 4-268: FNR.RML.CTL.NF.6 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.NF.7	Intra-DC Net-C **COULD** provide a network telemetry API as a service	
	Affected components	Network Fabric Controller
	Contributing Partner	UBI

Co-funded by the European Union

	Comment	//

Table 4-269: FNR.RML.CTL.NF.7 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.NF.8	Intra-DC Net-C **COULD** discover and manage the network topology, including edge deployments, core cloud, autonomous systems, and connectivity between them.	
	Affected components	Network Fabric Controller
	Contributing Partner	K3Y
	Comment	//

Table 4-270: FNR.RML.CTL.NF.8 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.NF.9	Net-C **COULD** provide comprehensive telemetry facilities, allowing administrators to collect, analyse, and visualize network performance and operational data.	
	Affected components	Network Fabric Controller
	Contributing Partner	K3Y
	Comment	//

Table 4-271: FNR.RML.CTL.NF.9 requirement

4.2.3.8 Network Fabric Driver Functional Requirements

Req. Id	Requirement Description	
FNR.RML.DRV.NF.1	Intra-DC Net-D for P4 devices **MUST** implement the P4Runtime SDN protocol for managing P4 targets (i.e., hardware/software switches, SmartNICs)	
	Affected components	Network Fabric Driver
	Contributing Partner	UBI
	Comment	//

Table 4-272: FNR.RML.DRV.NF.1 requirement

Req. Id	Requirement Description	
FNR.RML.DRV.NF.2	Intra-DC Net-D for P4 devices **MUST** provide an API to connect/disconnect to/from a P4 target	
	Affected components	Network Fabric Driver
	Contributing Partner	UBI
	Comment	//

Table 4-273: FNR.RML.DRV.NF.2 requirement

Req. Id	Requirement Description	
FNR.RML.DRV.NF.3	Intra-DC Net-D for P4 devices **MUST** provide an API to load a P4 program onto a specific P4 target	
	Affected components	Network Fabric Driver
	Contributing Partner	UBI

Co-funded by the European Union

	Comment	//

Table 4-274: FNR.RML.DRV.NF.3 requirement

Req. Id	Requirement Description	
FNR.RML.DRV.NF.4	Intra-DC Net-D for P4 devices **MUST** provide an API to insert/modify/delete forwarding entries into/from P4 targets	
	Affected components	Network Fabric Driver
	Contributing Partner	UBI
	Comment	//

Table 4-275: FNR.RML.DRV.NF.4 requirement

Req. Id	Requirement Description	
FNR.RML.DRV.NF.5	Intra-DC Net-D for P4 devices **COULD** provide an API to insert/modify/delete network telemetry instructions into/from P4 targets	
	Affected components	Network Fabric Driver
	Contributing Partner	UBI
	Comment	//

Table 4-276: FNR.RML.DRV.NF.5 requirement

Req. Id	Requirement Description	
FNR.RML.DRV.NF.6	Net-D for P4 devices **COULD** support network traffic isolation mechanisms, ensuring that different traffic streams or user groups are logically separated and protected from each other.	
	Affected components	Network Fabric Driver
	Contributing Partner	K3Y
	Comment	//

Table 4-277: FNR.RML.DRV.NF.6 requirement

4.2.3.9 Mobile Core Controller Functional Requirements

Req. Id	Requirement Description	
FNR.RML.CTL.MC.1	Access to the SBA architecture **MUST** have access to the 5G core services	
	Affected components	Mobile Core Controller
	Contributing Partner	UoP
	Comment	//

Table 4-278: FNR.RML.CTL.MC.1 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.MC.2	Deploy 5G services **MUST** deploy 5G services as microservices over the underlying 5G infrastructure	
	Affected components	Mobile Core Controller
	Contributing Partner	UoP
	Comment	//

Table 4-279: FNR.RML.CTL.MC.2 requirement

Co-funded by the European Union

Req. Id	Requirement Description	
FNR.RML.CTL.MC.3	Prometheus Telemetry **MUST** consume telemetry of the 5G microservices and 5G infrastructure in Prometheus format	
	Affected components	Mobile Core Controller
	Contributing Partner	UoP
	Comment	//

Table 4-280: FNR.RML.CTL.MC.3 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.MC.4	Expose NBI **MUST** provide Northbound APIs to overlay components	
	Affected components	Mobile Core Controller
	Contributing Partner	UoP
	Comment	//

Table 4-281: FNR.RML.CTL.MC.4 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.MC.5	Mob-C **MUST** support the deployment of network functions (NFs) within the mobile core. It must be able to initiate the installation and configuration of NFs, ensuring their proper integration into the network.	
	Affected components	Mobile Core Controller
	Contributing Partner	K3Y
	Comment	//

Table 4-282: FNR.RML.CTL.MC.5 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.MC.6	Mob-C **COULD** integrate with higher-level orchestration systems such as network orchestrators or service orchestrators. This integration allows for end-to-end management and coordination of network resources and services.	
	Affected components	Mobile Core Controller
	Contributing Partner	K3Y
	Comment	//

Table 4-283: FNR.RML.CTL.MC.6 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.MC.7	Mob-C **COULD** support policy management for the core functions	
	Affected components	Mobile Core Controller
	Contributing Partner	K3Y
	Comment	//

Table 4-284: FNR.RML.CTL.MC.7 requirement

Co-funded by the European Union

4.2.3.10 Mobile Core Driver Functional Requirements

Req. Id	Requirement Description	
FNR.RML.DRV.MC.1	Access to Infrastructure (SBI) **MUST** have access to the underlying drivers and infrastructure it will be affecting	
	Affected components	Mobile Core Driver
	Contributing Partner	UoP
	Comment	//

Table 4-285: FNR.RML.DRV.MC.1 requirement

Req. Id	Requirement Description	
FNR.RML.DRV.MC.2	5G core aaS **COULD** have the ability to deploy a 5G core on demand with a single API call	
	Affected components	Mobile Core Driver
	Contributing Partner	UoP
	Comment	//

Table 4-286: FNR.RML.DRV.MC.2 requirement

Req. Id	Requirement Description	
FNR.RML.DRV.MC.3	Provide vertical oriented network profiles **COULD** have the ability to deploy use case specific UPF instances on edge nodes according to network profiles of vertical applications	
	Affected components	Mobile Core Driver
	Contributing Partner	UoP
	Comment	//

Table 4-287: FNR.RML.DRV.MC.3 requirement

4.2.3.11 Cloud Controller Functional Requirements

Req. Id	Requirement Description	
FNR.RML.CTL.CL.1	Cloud-C **MUST** expose cluster information, such as cluster size (number of nodes), cluster formation (i.e., master node, worker nodes)	
	Affected components	Cloud Controller
	Contributing Partner	UBI
	Comment	//

Table 4-288: FNR.RML.CTL.CL.1 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.CL.2	Cloud-C **MUST** expose cluster resources in terms of CPU, RAM, Storage	
	Affected components	Cloud Controller
	Contributing Partner	UBI
	Comment	//

Table 4-289: FNR.RML.CTL.CL.2 requirement

Co-funded by the European Union

Req. Id	Requirement Description	
FNR.RML.CTL.CL.3	Cloud-C **MUST** expose special device resources (e.g., GPU, FPGA)	
	Affected components	Cloud Controller
	Contributing Partner	UNIS
	Comment	//

Table 4-290: FNR.RML.CTL.CL.3 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.CL.4	Cloud-C **MUST** expose APIs for managing deployment units	
	Affected components	Cloud Controller
	Contributing Partner	UBI
	Comment	//

Table 4-291: FNR.RML.CTL.CL.4 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.CL.5	Cloud-C **MUST** expose APIs for managing deployments	
	Affected components	Cloud Controller
	Contributing Partner	UBI
	Comment	//

Table 4-292: FNR.RML.CTL.CL.5 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.CL.6	Cloud-C **MUST** expose APIs for managing services	
	Affected components	Cloud Controller
	Contributing Partner	UBI
	Comment	//

Table 4-293: FNR.RML.CTL.CL.6 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.CL.7	Cloud-C **MUST** expose APIs for managing persistent storage towards the Distributed Storage component	
	Affected components	Cloud Controller
	Contributing Partner	UBI
	Comment	//

Table 4-294: FNR.RML.CTL.CL.7 requirement

Req. Id	Requirement Description	
FNR.RML.CTL.CL.8	Cloud-C **MUST** expose APIs for scaling resources in and out	
	Affected components	Cloud Controller
	Contributing Partner	UBI
	Comment	//

Table 4-295: FNR.RML.CTL.CL.8 requirement

Co-funded by the European Union

4.2.3.12 Cloud Driver Functional Requirements

Req. Id	Requirement Description	
FNR.RML.DRV.CL.1	Cloud-D for general purpose CPU **MUST** be able to allocate CPU, RAM, and Storage resources on a certain node	
	Affected components	Cloud Driver
	Contributing Partner	UBI
	Comment	//

Table 4-296: FNR.RML.DRV.CL.1 requirement

Req. Id	Requirement Description	
FNR.RML.DRV.CL.2	Cloud-D for general purpose CPU **MUST** be able to schedule/decommission tasks on a certain node	
	Affected components	Cloud Driver
	Contributing Partner	UBI
	Comment	//

Table 4-297: FNR.RML.DRV.CL.2 requirement

Req. Id	Requirement Description	
FNR.RML.DRV.CL.3	Cloud-D for general purpose CPU **MUST** support multiple hardware architectures, including high-end servers based on amd64 and low-end devices based on ARM64	
	Affected components	Cloud Driver
	Contributing Partner	UNIS
	Comment	//

Table 4-298: FNR.RML.DRV.CL.3 requirement

Req. Id	Requirement Description	
FNR.RML.DRV.CL.4	Cloud-D for GPU **MUST** be able to allocate GPU resources on a certain node	
	Affected components	Cloud Driver
	Contributing Partner	UNIS
	Comment	//

Table 4-299: FNR.RML.DRV.CL.4 requirement

Req. Id	Requirement Description	
FNR.RML.DRV.CL.5	Cloud-D for GPU **MUST** be able to schedule/decommission tasks on the GPU of a certain node	
	Affected components	Cloud Driver
	Contributing Partner	UNIS
	Comment	//

Table 4-300: FNR.RML.DRV.CL.5 requirement

4.2.4 Functional ST-L requirements

This section provides the complete list of functional requirements associated to the ST-L that need to be provided by the INCODE platform.

Co-funded by the European Union

4.2.4.1 Identity Management Functional Requirements

Req. Id	Requirement Description	
FNR.STL.IM.1	Service Authentication **MUST** use harmonized representation of the data defined through a data model and using well-known ontologies	
	Affected components	Identity Management
	Contributing Partner	FIWARE, NEC
	Comment	//

Table 4-301: FNR.STL.IM.1 requirement

Req. Id	Requirement Description	
FNR.STL.IM.2	Service Authentication **MUST** use JWT format to distribute the authentication information	
	Affected components	Identity Management
	Contributing Partner	FIWARE
	Comment	//

Table 4-302: FNR.STL.IM.2 requirement

Req. Id	Requirement Description	
FNR.STL.IM.3	Service Authentication **MUST** use DID to represent the identity of the device	
	Affected components	Identity Management
	Contributing Partner	FIWARE
	Comment	//

Table 4-303: FNR.STL.IM.3 requirement

Req. Id	Requirement Description	
FNR.STL.IM.4	Decentralised Authentication **MUST** use Verifiable credentials (VC) issued by Self-Sovereign Identity (SSI) Management of the organisation represented by the actor	
	Affected components	Identity Management
	Contributing Partner	FIWARE
	Comment	//

Table 4-304: FNR.STL.IM.4 requirement

Req. Id	Requirement Description	
FNR.STL.IM.5	Decentralised Authentication **MUST** use Decentralised Trust Anchor Framework to validate Verified Credentials, SSI which issues the VCs and Authorization	
	Affected components	Identity Management
	Contributing Partner	FIWARE
	Comment	//

Table 4-305: FNR.STL.IM.5 requirement

Co-funded by the European Union

Req. Id	Requirement Description	
FNR.STL.IM.6	Decentralised Authentication **MUST** use Attribute & Role Based Access Control based supporting Verifiable Credentials and SSI	
	Affected components	Identity Management
	Contributing Partner	FIWARE
	Comment	//

Table 4-306: FNR.STL.IM.6 requirement

Req. Id	Requirement Description	
FNR.STL.IM.7	A blockchain platform **MUST** provide a registration mechanism, that assigns unique IDs to entities registered to INCODE and records them in the ledger in the form of transactions	
	Affected components	Identity Management
	Contributing Partner	FIWARE, NEC
	Comment	//

Table 4-307: FNR.STL.IM.7 requirement

Req. Id	Requirement Description	
FNR.STL.IM.8	A blockchain platform **MUST** enable the verification of registered IDs	
	Affected components	Identity Management
	Contributing Partner	NEC
	Comment	//

Table 4-308: FNR.STL.IM.8 requirement

Req. Id	Requirement Description	
FNR.STL.IM.9	Registration and access control process **MUST** provide access to INCODE for IoT nodes, gateways and drivers through registration to the blockchain	
	Affected components	Identity Management
	Contributing Partner	NEC
	Comment	//

Table 4-309: FNR.STL.IM.9 requirement

Req. Id	Requirement Description	
FNR.STL.IM.10	Blockchain-bases consensus protocol **MUST** ensure consistency of recorded data and liveness/availability of the transaction ledger	
	Affected components	Identity Management
	Contributing Partner	NEC
	Comment	//

Table 4-310: FNR.STL.IM.10 requirement

Req. Id	Requirement Description	
FNR.STL.IM.11	A replication mechanism **MUST** replicate content across multiple nodes to ensure reliable and fault-tolerant storage	
	Affected components	Identity Management
	Contributing Partner	NEC

Co-funded by the European Union

	Comment	//

Table 4-311: FNR.STL.IM.11 requirement

Req. Id	Requirement Description	
FNR.STL.IM.12	Device authentication **MUST** use harmonized representation of the data defined through a data model and using well-known ontologies	
	Affected components	Identity Management
	Contributing Partner	FIWARE
	Comment	//

Table 4-312: FNR.STL.IM.12 requirement

4.2.4.2 Attestation Functional Requirements

Req. Id	Requirement Description	
FNR.STL.AT.1	Attestation mechanism **MUST** ensure only verified devices can be approved to enter the platform	
	Affected components	Attestation
	Contributing Partner	NEC
	Comment	//

Table 44-313: FNR.STL.AT.1 requirement

Req. Id	Requirement Description	
FNR.STL.AT.2	Attestation mechanism **MUST** use harmonized representation of the data defined through a data model and using well-known ontologies	
	Affected components	Attestation
	Contributing Partner	FIWARE
	Comment	//

Table 4-314: FNR.STL.AT.2 requirement

Req. Id	Requirement Description	
FNR.STL.AT.3	Attestation mechanism **MUST** define a set of attestable attributes to participate in the network	
	Affected components	Attestation
	Contributing Partner	NEC
	Comment	//

Table 4-315: FNR.STL.AT.3 requirement

Req. Id	Requirement Description	
FNR.STL.AT.4	Attestation mechanism **MUST** implement a Trusted Framework with a Trusted Participation List, where blockchain is used to implement the Trusted Participation List	
	Affected components	Attestation
	Contributing Partner	NEC
	Comment	//

Table 4-316: FNR.STL.AT.4 requirement

Co-funded by the European Union

Req. Id	Requirement Description	
FNR.STL.AT.5	Attestation mechanism **MUST** request the creation of one or more IoT Certification entities which are trusted by all participants	
	Affected components	Attestation
	Contributing Partner	NEC
	Comment	//

Table 4-317: FNR.STL.AT.5 requirement

Req. Id	Requirement Description	
FNR.STL.AT.6	Attestation **MUST** be able to include software binary analysis in order to ensure if the relevant software (in a binary format) has been compromised by a malware	
	Affected components	Attestation
	Contributing Partner	K3Y
	Comment	//

Table 4-318: FNR.STL.AT.6 requirement

Req. Id	Requirement Description	
FNR.STL.AT.7	Attestation mechanism **MUST** enable verification of the device's ID	
	Affected components	Attestation
	Contributing Partner	NEC
	Comment	//

Table 4-319: FNR.STL.AT.7 requirement

Req. Id	Requirement Description	
FNR.STL.AT.8	Attestation mechanism **MUST** enable verification of the integrity of the device's firmware and configuration	
	Affected components	Attestation
	Contributing Partner	NEC
	Comment	//

Table 4-320: FNR.STL.AT.8 requirement

Req. Id	Requirement Description	
FNR.STL.AT.9	Attestation mechanism **COULD** use HW-based remote attestation to certify the integrity of INCODE devices equipped with trusted hardware	
	Affected components	Attestation
	Contributing Partner	NEC
	Comment	//

Table 4-321: FNR.STL.AT.9 requirement

Co-funded by the European Union

4.2.4.3 Data Provenance Functional Requirements

Req. Id	Requirement Description	
FNR.STL.DP.1	Data Provenance **MUST** use harmonized representation of the data defined through a data model and using well-known ontologies	
	Affected components	Data Provenance
	Contributing Partner	FIWARE
	Comment	//

Table 4-322: FNR.STL.DP.1 requirement

Req. Id	Requirement Description	
FNR.STL.DP.2	Data Provenance **MUST** be deployed onto permissioned public blockchain network	
	Affected components	Data Provenance
	Contributing Partner	FIWARE, NEC
	Comment	//

Table 4-323: FNR.STL.DP.2 requirement

Req. Id	Requirement Description	
FNR.STL.DP.3	Data Provenance **MUST** interoperate with the Attack Mitigation component for detecting and mitigating adversarial attacks	
	Affected components	Data Provenance
	Contributing Partner	FIWARE, K3Y
	Comment	//

Table 4-324: FNR.STL.DP.3 requirement

Req. Id	Requirement Description	
FNR.STL.DP.4	Data Provenance **MUST** capture information about the original source of the data, including its author, creation timestamp, and relevant metadata. This information helps establish the trustworthiness and credibility of the data.	
	Affected components	Data Provenance
	Contributing Partner	FIWARE
	Comment	//

Table 4-325: FNR.STL.DP.4 requirement

Req. Id	Requirement Description	
FNR.STL.DP.5	Data Provenance **MUST** implement data- and application-provenance mechanisms for logging/auditing models, datasets and software artefacts in order to guarantee traceability	
	Affected components	Data Provenance
	Contributing Partner	FIWARE
	Comment	//

Table 4-326: FNR.STL.DP.5 requirement

Co-funded by the European Union

Req. Id	Requirement Description	
FNR.STL.DP.6	Data Provenance **MUST** use smart contracts to audit the consistency of data (reported by distributed data sources) and ensure trusted and reliable software and data inside the platform	
	Affected components	Data Provenance
	Contributing Partner	FIWARE
	Comment	//

Table 4-327: FNR.STL.DP.6 requirement

Req. Id	Requirement Description	
FNR.STL.DP.7	Data Provenance **MUST** capture information about the workflow of generating the data when this data is generated as part of a larger process. This includes details about the sequence of steps, parameters used, software tools employed, and any contextual information necessary to reproduce the workflow.	
	Affected components	Data Provenance
	Contributing Partner	FIWARE
	Comment	//

Table 4-328: FNR.STL.DP.7 requirement

Req. Id	Requirement Description	
FNR.STL.DP.8	Data Provenance **MUST** track who accessed the data, when, and for what purpose. It helps in ensuring data privacy, security, and compliance with data governance policies. It has to be related to the Identity Management tool.	
	Affected components	Data Provenance
	Contributing Partner	FIWARE
	Comment	//

Table 4-329: FNR.STL.DP.8 requirement

Req. Id	Requirement Description	
FNR.STL.DP.9	Data Provenance **MUST** be protected from unauthorized modifications or tampering. It should have mechanisms for ensuring data integrity, authenticity, and confidentiality.	
	Affected components	Data Provenance
	Contributing Partner	FIWARE
	Comment	//

Table 4-330: FNR.STL.DP.9 requirement

Req. Id	Requirement Description	
FNR.STL.DP.10	Data Provenance **MUST** provide efficient mechanism to access the data. It should be scalable, adaptable to different data types and formats, and support the querying of provenance metadata.	
	Affected components	Data Provenance
	Contributing Partner	FIWARE
	Comment	//

Table 4-331: FNR.STL.DP.10 requirement

Co-funded by the European Union

4.2.4.4 Attack Mitigation Functional Requirements

Req. Id	Requirement Description	
FNR.STL.AM.1	Verification tools **COULD** ensure the detection of smart-contract bugs and vulnerabilities at runtime	
	Affected components	Attack Mitigation
	Contributing Partner	NEC
	Comment	//

Table 4-332: FNR.STL.AM.1 requirement

Req. Id	Requirement Description	
FNR.STL.AM.2	Attack Mitigation **COULD** include intrusion detection and prevention services in order to detect and indicate potential mitigation strategies	
	Affected components	Attack Mitigation
	Contributing Partner	K3Y
	Comment	//

Table 4-333: FNR.STL.AM.2 requirement

Req. Id	Requirement Description	
FNR.STL.AM.3	Attack Mitigation **COULD** be able to guide SDN mechanisms to mitigate efficiently potential cyberattacks, considering network and infrastructure characteristics	
	Affected components	Attack Mitigation
	Contributing Partner	K3Y
	Comment	//

Table 4-334: FNR.STL.AM.3 requirement

Req. Id	Requirement Description	
FNR.STL.AM.4	Attack Mitigation **MUST** use harmonized representation of the data defined through a data model and using well-known ontologies	
	Affected components	Attack Mitigation
	Contributing Partner	FIWARE
	Comment	//

Table 4-335: FNR.STL.AM.4 requirement

4.3 Non-functional requirements

The non-functional requirements characterize the properties that the developed (i.e., functional) solutions must have when the overall system is seen as a final product. More specifically, the non-functional requirements specify the criteria that can be used to judge the operation of a system, rather than specific behaviours of the implemented function, components, or modules. These requirements are essential for the product quality evaluation and therefore play an important role in the release of a final product at high TRL. According to ISO/IEC 25010, the quality characteristics are summarised in Figure 4-1.

Co-funded by the European Union

Figure 4-1: Non-functional requirements (ISO/IEC 25010) taken from https://iso25000.com/images/figures/en/iso25010.png

Here the non-functional requirements are introduced with the scope to determine which quality characteristics should be considered when evaluating the properties of the INCODE platform solution, especially when this solution will move to higher TRLs at the end of the project.

In the following table, we aim to provide a mapping between the functional requirements, and the non-functional ones, by highlighting the key quality characteristics that must be considered. The characterization and mapping are performed per group of functional requirements, since each group shares several functional requirements towards the same object.

Each of the non-functional requirements is explained and discussed with respect to the INCODE functional requirements in the following paragraphs.

Table 4-336: Relationships between Functional and Non-Functional requirements.

List of INCODE **Functional** requirements	Prevailing **Non-functional** Requirements (ISO/IEC 25010)							
	Functional Suitability	Performance Efficiency	Compatibility	Usability	Reliability	Security	Maintainability	Portability
FNR.GPR (Generic platform impl.)	○	○	○	○	○	○	○	○
FNR.GRML (Generic RM-L impl.)	○	○	○		○	○	○	
FNR.GSML (Generic SM-L impl.)	○	●	○	○	○		○	○
FNR.GUIR (User Interfacing and sharing)	○		●	●	○	○	○	○
FNR.GPROG (Programming)	○	●	○	●	○		○	○
FNR.GSEC (Security)	○					●		
FNR.GDAT (Data processing and regulations)	○		●					

4.3.1 Functional Suitability

According to [46], the functional suitability "represents the degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions." In other words, it is the quality that rates the degree to which the implemented product meets the functional requirements set initially at the development phase and derived by the end user (i.e., targeted customer needs). This characteristic provides a direct link to the functional

Co-funded by the European Union

requirements and therefore it is marked as a quality that applies for all functional requirement groups in INCODE.

4.3.2 Performance Efficiency

The performance efficiency “represents the performance relative to the number of resources used under stated conditions.” These quality characteristics relate mainly to the processing efficiency and the use of programming resources.

In INCODE, the Resource and Services Management Layers, as well as the IDP are the main modules that require adequate performance efficiency since they include intense processing functionalities. The related SL-M part concerns the processing of the monitoring results in its analytics engine and the quick response to required reconfigurations over deployed applications. The processing requirements are more intense in the case of RM-L and in the efficient identification of the slice resources according to the service requirements. In addition, the timely response to reconfiguration requests is important. However, the RM-L is deployed and executed in the core part of the network over the network orchestrator and with resources provided typically by the operator/infrastructure owner; thus, no strict requirements apply in terms of resource usage or deployment capacity. Finally, the Internal Developer Platform also has performance-related requirements to provide developers with the necessary tools and environments to streamline their work and deliver high-quality applications. Therefore, for INCODE to optimize IDP prioritizes parameters related to response time, concurrency, resource utilization, development time, and integration performance.

4.3.3 Compatibility

Compatibility refers to the “degree to which a product, system or component can exchange information with other products, systems or components, and/or perform its required functions while sharing the same hardware or software environment.” For software solutions that apply in network and communication systems, this requirement translates into compatibility with common standards, including already released and established standards or planned standardization activities for applied research-oriented solutions. A valid alternative is where the developed software operations are proven to be independent of any standards, e.g., using compatible or adaptable programmable interfaces.

INCODE promises the development of an SM-L deployment and management platform over an RM-L network orchestrator. This sets a key functional requirement that denotes the development of the appropriate API for the IoT, edge/cloud computing resources, and networking platforms so as to allow for full programmability and reconfigurability of these resources. A functional constraint to achieve this is that any interfaced proprietary network orchestrator exposes all its programmability aspects to the RM-L. Furthermore, the project examines compatible solutions in the fields of slice specification, and service specification models. At the SM-L, the compatibility requirements are more relaxed due to the diverse nature of the end-user interfacing options. Finally, with respect to data management compliance standards, a key compatibility requirement arises to align any future high TRL solution to the specific EU regulations identified.

4.3.4 Usability

Usability refers to the “degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.” The usability feature has particular meaning for solutions with end user interfaces or

Co-funded by the European Union

platforms that are destined to be adopted as services and integrated over existing infrastructures.

This characteristic is of paramount importance for the INCODE platform's user interface and the capabilities it offers towards a) IDP users for the proper boarding, update, and reconfiguration of the application components, b) the vertical end users for deploying and managing their features and tools but also reconfiguring on demand c) the operator and end users for accessing the monitored information. Furthermore, the interface must be easily understandable as well as operatable by different end users, provide the means to avoid critical human-oriented mistakes, and provide a pleasing and satisfying environment to work with.

4.3.5 Reliability

Reliability as a non-functional requirement refers to the "degree to which a system, product or component performs specified functions under specified conditions for a specified period of time." This quality characteristic relates with the ability of the system to be constantly operational and accessible and deliver its destined functionalities under normal operation. It also extends to the ability to quickly recover from faults and failures or equally to continue its operation when associated underlay hardware systems recover without the need for re-initiation and long reconfiguration processes.

In INCODE the reliability feature is connected to the modules that perform the main platform operations (RM-L and SM-L). This can be achieved by contentiously keeping track of the module status and established configurations. The monitoring functions in the INCODE platform have the additional goal to identify potential changes and resource failures and then act through the reconfiguration mechanism, thus providing a higher layer-oriented recovery. For a final high TRL, specific recovery mechanisms must be also considered for the RM-L and SM-L that allow quick recovery of data through updated images when this is required.

4.3.6 Security

Security refers to the "degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization." This is an important feature for any platform that hosts multiple end users and is prompt to potential attacks that target the interception of application data or the performance of malicious attacks to harm critical end user services.

In INCODE, increased cybersecurity needs have been identified primarily for the RM-L and the IoT collected datasets. Although the focus of the project is not on the establishment and enhancement of strong security protocols, user authentication, components' verification and attack mitigation mechanisms are by default implemented several parts of the INCODE system. Regarding the data security, the overall data provenance concept relies on potential enhancements based on tools like blockchain either on demand or in an automated manner, once a potential treat is identified (e.g., by the attack mitigation tool). On the other hand, threats on the RM-L are not likely to occur since these modules are within the network operator environment and protected from unauthorised external access. Therefore, they are not directly addressed by the project. However, there are solutions from the project's consortium members that tackle security on the network functions level through risk assessment and enhanced attestation mechanisms and can be evaluated for potential high TRL releases beyond the project duration.

Co-funded by
the European Union

4.3.7 Maintainability

Maintainability “represents the degree of effectiveness and efficiency with which a product or system can be modified to improve it, correct it or adapt it to changes in environment, and in requirements.” The key feature in this quality characteristic is the ability of the software product or individual software components to adapt to any potential updates and changes, thus contributing to the longevity of the solution and consequently its market quality and presence.

INCODE has adopted a modular approach that enables parts of the overall platform to be easily updated, modified or even replaced without affecting other modules. The modularity applies also in the development of specific processes by following a micro-services deployment approach. Therefore, additional features, monitoring and analytic engines, optimisation processes and adopted data models can be easily tailored to end-users' need and modified or updated accordingly.

There are two main fields identified within INCODE where maintainability is required. The first relates to the underlay compatibility with the network orchestration. In this case, INCODE establishes a modular approach separating the RM-L functionalities from the underlay SM-L layer through a specific API that communicates the programmability capabilities of the infrastructure for the optimum allocation of the slices. The second field relates to the adaptability of the INCODE platform to different Application Areas. This requires adaptations only at the user interfacing part, to accommodate potential applications' features that are not considered in the targeted project release, as well as end user tailored analytics and reconfiguration options; the reconfiguration, monitoring and visualisation engines remain though the same.

4.3.8 Portability

Portability shows the “degree of effectiveness and efficiency with which a system, product or component can be transferred from one hardware, software or other operational or usage environment to another.” The key feature of portability that mostly relates to networking environments is the adaptability of the offered solution to the underlay hardware (and supported software) infrastructure.

In INCODE, portability is closely related to maintainability at this feature is described above. By adopting a modular approach, INCODE targets the adaptation of the platform to different infrastructure environments through easily modifiable interfaces. Hence, the key targeted requirement here is to provide a platform solution that is seamlessly operable over diverse orchestration environments offered by different operators and infrastructure owners as will be demonstrated in the different Application areas.

Co-funded by
the European Union

5 Definition of Application Areas and Link with Requirements

This section defines the four (4) application areas relevant to the INCODE project as well as the use cases that are designed and will be evaluated throughout this project.

5.1 AA1: Logistics and Transport Quality Value Chain

The AA1 of the INCODE Project aims to deliver technology solutions specifically focused on supply chain management and logistics. In this context, the AA1 consists of two (2) different UCs providing a holistic deployment and utilization of the platform capabilities offered in this project, such as 5G networks, far-edge computing, vertical service deployment, component reusability, and both research and business applications. The AA1 use cases are summarized in Table 5-1 and fully described in Section 5.1.3 (AA1 UC1) and Section 5.1.4 (AA1 UC2) respectively.

Table 5-1: AA1 – Summary of use cases.

Use Case Acronym	Use Case Name
AA1-UC1	Smart Industrial Safety and Asset Orchestration (Internal Logistics)
AA1-UC2	Route Optimization and Cargo Monitoring (External Logistics)

5.1.1 Objective

The objectives of AA1 are classified in the several important categories, in which the designed use cases provide substantial contributions.

Enhanced Service Optimization

✔ Optimize services throughout the value chain by streamlining processes and improving efficiency.
✔ Implement advanced data analytics to enable data-driven decision-making for service enhancements.

✔ Optimised services across the value chain through a super-slice among core and edge services, with edge intelligence diffusion

5G Infrastructure Deployment

✔ Deploy services and applications leveraging 5G infrastructure to ensure high-speed connectivity, low latency, and increased network reliability.

✔ Deployment over 5G infrastructure

✔ Rule definition for optimising user-guided application deployment

Edge Intelligence Integration

Co-funded by the European Union

✔ Integrate edge intelligence into core and edge services to enable real-time data processing and decision-making at the network edge.

Security and Trustworthiness

✔ Establish a trusted framework by implementing robust security measures, including IoT attestation and verified drivers, to protect data and ensure system integrity.

✔ Trusted framework enabled (incl. IoT attestation, verified drivers)

Monitoring and Analytics

✔ Implement comprehensive monitoring and analytics tools to gain insights into system performance, user behaviour, and service effectiveness, enabling continuous improvement.

Time and Event-Triggered Applications

✔ Develop and demonstrate time or event-triggered applications, particularly those activated at the time of vehicle arrival, to enhance user convenience and efficiency.

✔ Demonstrate programmable application features over IoT nodes

✔ Demonstrate time/event triggered apps (at time of vehicle arrival)

User Experience Enhancement

✔ Continuously improve the user experience by collecting feedback and iterating on services and applications.

Collaboration and Partnerships

✔ Establish partnerships and collaborations with relevant stakeholders, including technology providers, industry experts, and government agencies, to leverage expertise and resources.

Community Engagement

✔ Engage with the community and end-users to gather input, foster innovation, and build a user-centric ecosystem.

5.1.2 Why is it relevant for INCODE?

The INCODE project aspires to revolutionize the IoT landscape by creating a secure, trusted, and open IoT-to-edge-to-cloud compute continuum, unlocking the full potential of edge intelligence. Central to this vision is the development of an open platform for deploying and dynamically managing end-user applications across distributed, heterogeneous, and trusted IoT-Edge node infrastructures, enriched with advanced programmability features. Within this context, Application Area 1 is committed to optimizing services, integrating edge intelligence, deploying over 5G infrastructure, and ensuring robust security and trustworthiness. These objectives are fundamental to the INCODE mission, as they align with the project's overarching goal of reinforcing Europe's position in the next-generation smart systems market while fostering innovation, collaboration, and a superior user experience within the evolving Internet of Things and cyber-physical ecosystems. Through these objectives, Application Area 1 endeavors to create a future-ready, sustainable, and user-centric IoT ecosystem with a strong edge-computing focus, delivering tangible benefits to stakeholders and end-users that INCODE may pursue.

Co-funded by
the European Union

5.1.3 AA1-UC1: Smart Industrial Safety and Asset Orchestration (Internal Logistics)

AA1-Use Case 1: This UC aims to position awareness of moving staff (i.e., workers), moving infrastructure (i.e., forklifts), and potentially moving assets/entities (e.g., packaging, cargo) within industrial facilities. The assets' internal position cannot be calculated with compatible GPS technology, so this raises the need for the utilization of other technologies, such as ultra-wide band (UWB) anchors and wearable tags. Those IoT sensors (transmitters), combined with static anchors (receivers) can provide accurate positioning data, while utilizing state-of-the-art Network and Software Architectures (i.e., 5G, Edge computing).

The Data Collection and Analysis will examine (i) **data logging** by capturing a diverse range of data during real-world operations, including measurements from UWB sensors, environmental conditions, and ground truth positioning information, (ii) **data pre-processing** by forming the preliminary steps for the filtering, outlier removal, and synchronization in order to ensure data quality, and (iii) **feature analysis** that will collect data to determine the impact of different features on positioning accuracy. We will investigate techniques such as statistical analysis, correlation analysis, and feature importance ranking algorithms (e.g., feature importance from random forests) that can be applied to identify the significant features.

Identifying the important features of such AI algorithms and understanding how environmental conditions can alter the results and outcomes are crucial steps in designing an effective plan optimization system. This use case will try to go a step further in the feature selection, by calibrating and refining the positioning measurements relying on relevant features. However, determining which features are important can be challenging as it depends on the signal strength (the signal strength of UWB measurements can vary based on distance, obstacles, and interference), the Time of Flight (ToF) that provides distance information between UWB anchors and tags (factors such as multipath reflections and non-line-of-sight conditions can affect the accuracy of ToF measurements), Environmental Factors (such as the presence of obstacles, reflective surfaces, or electromagnetic interference, may affect the performance of the positioning system), and most importantly data collection and analysis, which will be the core component that will try to understand the importance of features and their relationship with positioning accuracy.

Further, this UC aims to the establishment of a safer industrial working environment. The movement correlation of internal assets (workers, forklifts) is constantly monitored and feeds a "Collision- Avoidance Algorithm" which "decides" whether there will be an imminent collision and provides alert notifications to the end users involved via a mobile App. This module significantly contributes to smart industrial safety. Apparently, the specific algorithm cooperates with the location of internal entities. Collision avoidance with enhanced location data (via ML). By incorporating real-time positional data, the algorithm can analyse the movement trajectories, speeds, and distances between workers and forklifts more effectively. This enriched data allows for a better understanding of the proximity and potential collision risks. The algorithm can also consider factors, such as the layout of the working environment, the presence of obstacles or restricted areas, and the characteristics of the assets involved. This comprehensive analysis enhances the algorithm's ability to accurately assess collision risks and identify potentially hazardous situations. On top of that, the AI algorithm can leverage positioning data to optimize collision alerts and notifications. By considering the precise locations of workers and forklifts, the algorithm can determine the urgency and severity of a potential collision. For example, if a worker is near a forklift, but moving away from the potential collision path, the algorithm can differentiate between critical collision alerts and less urgent warnings. This intelligent alert system ensures that workers and forklift operators receive timely notifications that reflect the real-time risk level, enabling them to take appropriate actions to prevent accidents. The algorithm can also consider contextual information, such as the speed and direction of assets, to provide additional insights for collision avoidance strategies. Part of

Co-funded by the European Union

the research is to investigate the peer-to-peer information exchange in the information layer as well as in the network layer.

By combining the service from the positioning system described earlier, with the AI algorithm used in the collision-avoidance module, the existing service can be upgraded to further enhance industrial safety. The AI algorithm can leverage the real-time positioning data of workers and forklifts to provide more accurate collision predictions and alerts. Initially, it will utilize the precise location information obtained from the positioning system to improve collision detection accuracy. Another functionality of this specific UC will be the provision of an extra module that significantly contributes to smart safety within industrial environments. In detail, the onboarded mobile device (on forklifts) will also run computer vision models capable of identifying moving persons and estimating the precise distance between the camera and the person (worker). This way, an audio-visual alert will be generated to the driver in the context of additional awareness of other moving entities. This extra module covers additional scenarios where there are implications in main CA functioning, while utilizing the capabilities of far-edge computing with very low or no network dependencies. (extended functionality).

The integration of the AI algorithm with the positioning system not only enhances the accuracy and effectiveness of collision detection but also enables a more intelligent and adaptable collision-avoidance solution. By continuously analysing and learning from the real-time positioning data (a distributed Reinforcement Learning algorithm may be used for that case), the AI algorithm can improve its predictions and adapt to changing conditions or specific working environments. This integration enables a holistic approach to industrial safety, combining precise positioning data with intelligent collision-avoidance algorithms, to create a safer and more productive working environment. On top of this component, an additional AI feature will be provided to the UC. More precisely, forklifts will be equipped with a local processor (e.g., a Raspberry Pi 4 model B) combined with a camera that collects video input for a computer vision model responsible for identifying other moving or static entities (workers and forklifts) and estimating their distance to the camera. This way, the collision avoidance module is further enhanced by providing an additional alerting system even in situations where the network is down and gives significant value to the far-edge computing capabilities.

This use case also collects local data from multiple sensors, IoT nodes, and inventory data at the warehouse. Those are combined with dynamic data from trucks, loading vehicles, and workers. Considering the constraints which are applicable to the resources, an additional output of this use case is the optimized loading/unloading scheduling, as well as the allocation of human and vehicle resources. Additionally, the assets' actual position and other dynamic data can also act as rescheduling factors, e.g., a new task needs to be done and the software assigns it to the closest available entity.

Finally, this use case uses advanced analytics applications to provide insights, for example regarding the scheduling optimization at the warehouse, potentially on an eligible time basis (weekly, daily, real-time correction), or a heatmap of possible collision areas inside the warehouse to allow the human operators to take decisions on how the set-up of the warehouse can be optimized.

An illustrated overview of the AA1 is shown in Figure 5-1, where the hardware and software components of the AA1-UC1 are co-designed within the blue frame. A similar design is made for the AA1-UC2, enclosed in the red frame in Figure 5-1.

Co-funded by
the European Union

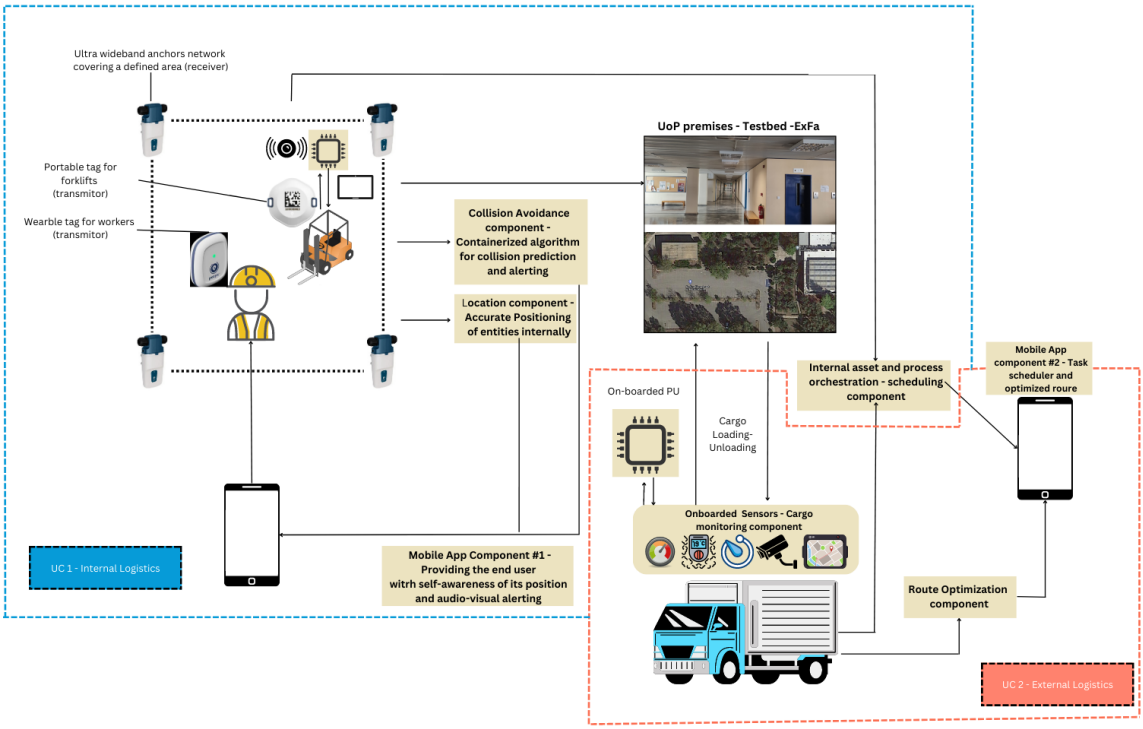

Figure 5-1: AA1 UC1 (Blue frame) – Overview.

5.1.3.1 Objective

This UC aims to position awareness of moving (Workers, Forklifts) and potentially moving (Packaging, Cargo) assets/entities within industrial facilities. The assets' internal position cannot be calculated with compatible GPS technology, so this raises the need for the utilization of other technologies, such as UWB anchors and wearable tags. Those IoT sensors (transmitters), combined with static anchors (receivers) can provide very accurate positioning data while utilizing state-of-the-art Network and Software Architectures (5G, Kubernetes, Edge computing).

The Data Collection and Analysis will examine the **data logging** by capturing a diverse range of data during real-world operations, including measurements from UWB sensors, environmental conditions, and ground truth positioning information, the **data pre-processing** by forming the preliminary steps for the filtering, outlier removal, and synchronization in order to ensure data quality, and the **feature analysis** that will collect data to determine the impact of different features on positioning accuracy. We will investigate techniques such as statistical analysis, correlation analysis, and feature importance ranking algorithms (e.g., feature importance from random forests) that can be applied to identify the significant features.

5.1.3.2 End-user Service Components

Table 5-2 introduces a list of (software-based) end-user service components used to drive the business logic of this use case.

Co-funded by the European Union

Table 5-2: AA1 UC1 – End-user service components.

End-User Service Component				
ID	**Name**	**Description**	**Type**	**Deployment Domain**
APP-01-a	MongoDB	NoSQL DB used for the internal positioning system UC1	Centralized	Edge Cloud
APP-01-d	Location	Backend interacts with the IoT/UWB sensors, stores data into MongoDB, and exposes services to end users via Open2.0 REST API	Centralized	Edge Cloud
APP-01-e	Collision	Internal location data fuel the algorithm to provide real-time and precise alerts	Centralized	Edge Cloud
APP-01-f	Scheduler	Takes inputs from internal and external assets to provide operations/tasks orchestration		Edge Cloud

5.1.3.3 Initial Use Case State

This section outlines the initial state of the system, before executing the AA1-UC1 workflow. This state is captured by the entries in Table 5-3.

Table 5-3: AA1 UC1 – Initial state of the system prior to the use case execution.

Initial State of the System prior to this UC
A private 5G network that will provide high-speed, low-latency data transmission must be configured at ExFa premises.
IoT sensors (UWB beacons & Anchors) must be appropriately installed, configured, and connected to the private 5G network. Interconnectivity of IoT location sensors with the edge components of the Network Application.
Wearable and portable tags must be carried by the workers and placed on the forklifts.
Mobile and Admin Apps must be up and running. Users must have successfully logged in the App.
Methods to ingest data FROM sensors, IoT nodes and inventorying data at the warehouse TO the scheduling optimisation module are in place (e.g., APIs)
Methods to share the results of resource optimisation and/or the results of analytics methods are in place (e.g., APIs)

Co-funded by the European Union

5.1.3.4 Sequence of Steps

Given that the AA1-UC1 environment is in the state described in the previous section, this section outlines the steps to execute the business logic of this use case. For every step we describe the required input, objective, and expected output, summarized in Table 5-4.

Table 5-4: AA1 UC1 – Set of steps necessary for the use case execution.

Step Number	Input	Objective	Output
S1	Asset's X, Y coordinates	Real-time location awareness and collision alerting	Depiction of every asset's location
S2	Type of each asset (Forklift, Worker)	Identify the moving assets for the accurate collision prediction	Each asset holds an ID with qualitative specs
S3	Assets' vector velocity, vector acceleration	These metrics also affect the collision prediction algorithm	Collision alerting
S4	Data from sensors, IoT nodes and inventorying data at the warehouse.	Number of available staff, vehicles, to-be-delivered cargo, and any other qualitative parameters which can act as constraints such as maintenance activities, infrastructure malfunction, etc	Operations and warehouse management
S5	Video input from cameras mounted on the forklifts (attached to the raspberry Pi or mobile device)	Smart safety extension by the deployment of CV model that identifies other moving entities and calculates the intermediate distance	Audio Visual Alert and object identification and classification

5.1.3.5 Data Models

This section provides a preliminary data model for the ultra-wide band sensor to be used in AA1-UC1. This data model is shown in Figure 5-2.

Co-funded by the European Union

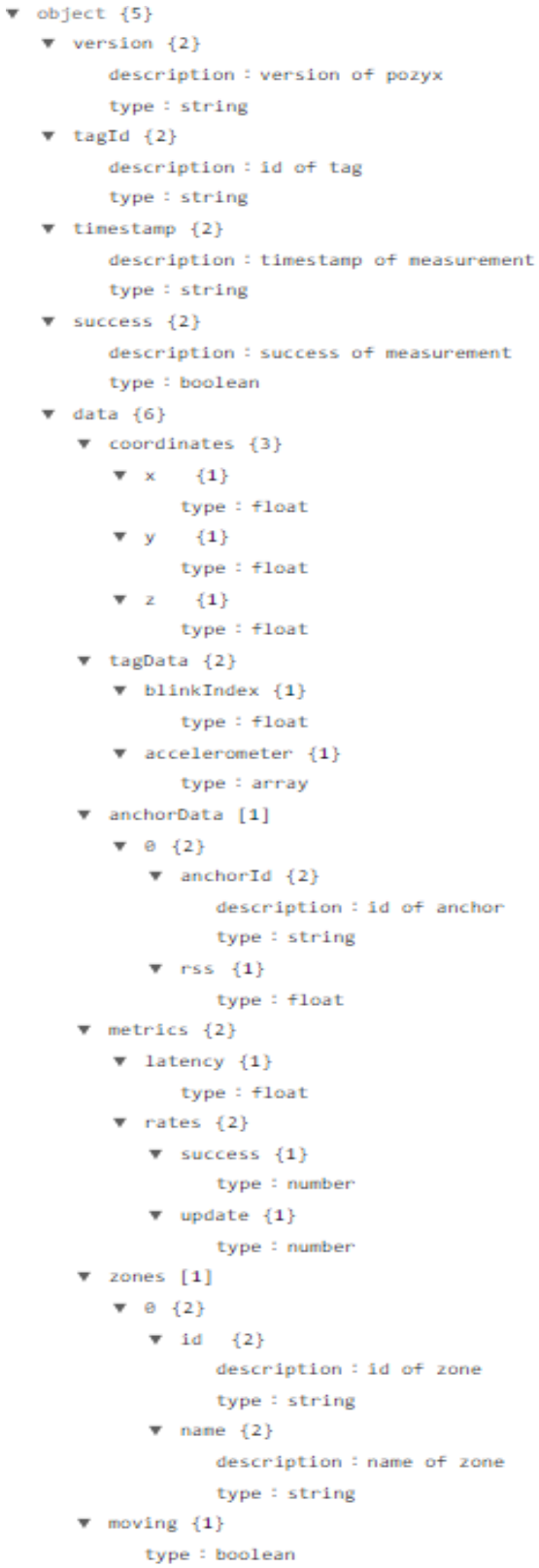

Figure 5-2: AA1 UC1 – Data model for the ultra-wide band sensor to be used in this use case.

5.1.3.6 Workflows

This section provides the steps to reproduce the AA1 UC1 and summarizes these steps through an illustrative - yet preliminary - workflow diagram shown in Figure 5-3.

Co-funded by
the European Union

Steps

1. Installation of anchors in the optimum spots aiming for the highest positioning accuracy of the assets.
2. Position data from the assets inside the facilities is collected from the sensors.
3. The actual position is calculated for further exploitation in the context of this ongoing Application Area or for other UCs within the INCODE Project.
4. Each asset is assigned qualitative characteristics (type of asset).
5. Each asset is visualized on the Mobile App.
6. Each asset is visualized on the Admin.
7. The collision algorithm communicates with the Location component and takes each asset's location as input.
8. The collision algorithm is executed repetitively in a defined time interval, projecting any potential imminent collisions.
9. Alerts are initiated on the devices based on the probability of an incident.
10. The incident is stored at the core part of the network for statistical analysis and heat map provision.
11. Operator initiates a request for the resolution of a resource optimization problem.
12. Resource Scheduling module verifies the necessary input data is available.
13. Resource Scheduling module verifies the necessary constraint parameters are available.
14. Resource Scheduling module resolves the resource optimization problem.
15. Operator is notified of successful resolution.
16. The solution of the resource optimization problem is shared with another module in a predefined technical manner.

Co-funded by
the European Union

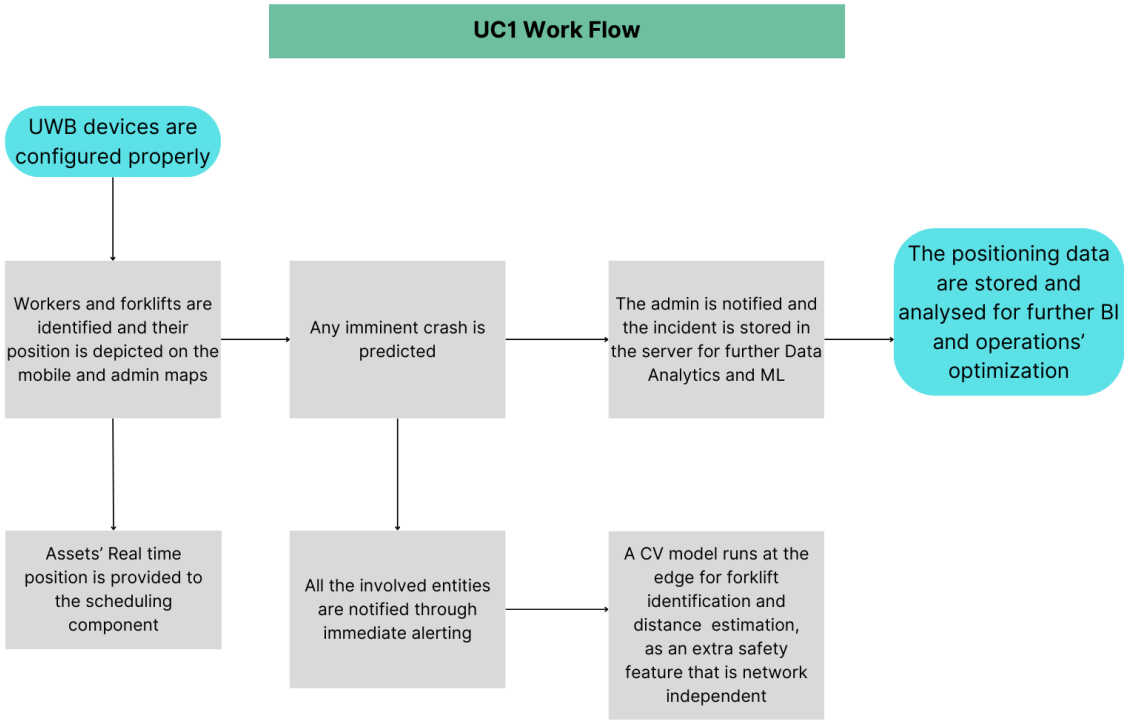

Figure 5-3: AA1 UC1 – Workflow diagram.

5.1.3.7 Requirements

A list of AA1-UC1-specific requirements is provided in the following tables.

Req. Id	Requirement Description	
AA1.UC1.01	Testbed and lab deployment **MUST** provide efficient 5G network connectivity between IoT devices and the AA1 UC1 services.	
	Affected components	IoT devices
	Contributing Partner	iLink, UoP
	Comment	//

Table 5-5: AA1.UC1.01 requirement

Req. Id	Requirement Description	
AA1.UC1.02	Indoor positioning **COULD** utilize UWB IoT sensors with high precision of entities positioning	
	Affected components	IoT devices
	Contributing Partner	iLink
	Comment	//

Table 5-6: AA1.UC1.02 requirement

Req. Id	Requirement Description	
AA1.UC1.03	5G coverage **MUST** provide bandwidth availability >99%	
	Affected components	IoT devices
	Contributing Partner	iLink, UoP

Co-funded by the European Union

	Comment	//

Table 5-7: AA1.UC1.03 requirement

Req. Id	Requirement Description	
AA1.UC1.04	Application responsiveness, uptime, and speed **COULD** be optimized through Dockerized software architecture	
	Affected components	Service components
	Contributing Partner	iLink, ARHS, SUITE5
	Comment	//

Table 5-8: AA1.UC1.04 requirement

Req. Id	Requirement Description	
AA1.UC1.05	Computing/processing **MUST** be implemented at the edge (5G) and far-edge	
	Affected components	RM-L RO, service components
	Contributing Partner	UoP, iLink, ARHS, SUITE5
	Comment	//

Table 5-9: AA1.UC1.05 requirement

Req. Id	Requirement Description	
AA1.UC1.06	Component to component communication latency **MUST** be <500ms	
	Affected components	RM-L, service components
	Contributing Partner	UoP, iLink, ARHS, SUITE5
	Comment	//

Table 5-10: AA1.UC1.06 requirement

5.1.3.8 Potential Future Extensions

The Application components referring to AI-ML algorithms will provide the enhancement of internal smart safety features. AI analytics will also provide optimised spatial and internal routing planning.

5.1.3.9 GDPR Issues

The UC is designed by taking into consideration the GDPR standards related to working ethics. The potential risk could be the exposure of workers' personal data as they log in to the system. Additionally, the position monitoring will not be linked to the worker's identity since each entity will be depicted as "worker" and "forklift" without any personal information registered.

5.1.3.10 Risk Assessment & Mitigation Plan

This section discusses risks that may arise while developing the AA1 UC1 and propose workarounds for mitigating those risks (see Table 5-11).

Co-funded by the European Union

Table 5-11: AA1 UC1 – Foreseen risks and mitigation measures.

Risk	Description of risk	Likelihood	Severity	Proposed risk-mitigation measure(s)
AA1-UC1-R01	Unauthorized access or data breaches that compromise the privacy of workers and sensitive business data.	Medium	Medium	Implement robust encryption protocols, access controls, and regular security audits. Adhere to relevant data privacy regulations and guidelines.
AA1-UC1-R02	Difficulty in integrating various technologies like UWB anchors, 5G infrastructure, and edge computing seamlessly.	Low	High	Conduct thorough testing and validation of technology integration before deployment. Establish a dedicated integration team and ensure adequate collaboration and communication.
AA1-UC1-R03	Potential hardware malfunctions or failures of IoT devices, sensors, or 5G infrastructure.	Medium	High	Employ redundancy and failover mechanisms. Regularly monitor and maintain hardware components to ensure their proper functioning.
AA1-UC1-R04	The system may struggle to handle a larger volume of assets or a sudden increase in workload, affecting performance and responsiveness.	Medium	High	Design the system with scalability in mind, utilizing cloud-based solutions and optimizing software architecture for efficient scaling. Perform stress tests to assess system behaviour under varying loads.
AA1-UC1-R05	Failure to comply with industry-specific or regional regulations related to IoT, data privacy, or industrial safety.	Low	Medium	Conduct a thorough regulatory analysis and ensure the system and its operations adhere to all applicable laws and standards.
AA1-UC1-R06	Resistance or reluctance from end-users to adopt and effectively use the system due to a lack of understanding or training.	Low	Medium	Implement an effective training program and provide comprehensive user documentation. Gather feedback and make necessary improvements to enhance user experience.

Co-funded by the European Union

5.1.4 AA1-UC2: Route Optimization and Cargo Monitoring

Use Case 2: In this UC, we aim to establish a dynamic process that can provide the real-time location of trucks by utilizing GPS sensors. Additionally, this module will act as a dynamic route optimizer by considering diverse data such as the available time windows, delivery addresses, cargo specifications, vehicle location, cargo transport conditions, etc. Undeniably, this function contributes to a more efficient and high-quality value chain system.

The cargo is another critical aspect of the logistics value chain. Apart from real-time location awareness, many other critical cargo-related data need to be monitored. IoT sensors' deployment can provide a continuous data acquisition flow which is an essential factor in terms of accurate and swift decision-making actions. Among the various data expected to be monitored is the temperature, humidity, sudden drop, pressure, etc.

Additionally, data standardization and prototyping will be implemented in the AA1. The main objective of this task is to define best practices and efficient ways to design the schema of use cases. The document will also cover the definition of the schema for storage, which can be based on block storage, queue, or even RPC calls. The primary goal of the document is to provide a comprehensive guideline for data standardization and schema definition that can be used throughout the project development lifecycle.

An illustrated overview of the AA1 is shown in Figure 5-4, where the hardware and software components of the AA1-UC2 are co-designed within the orange frame.

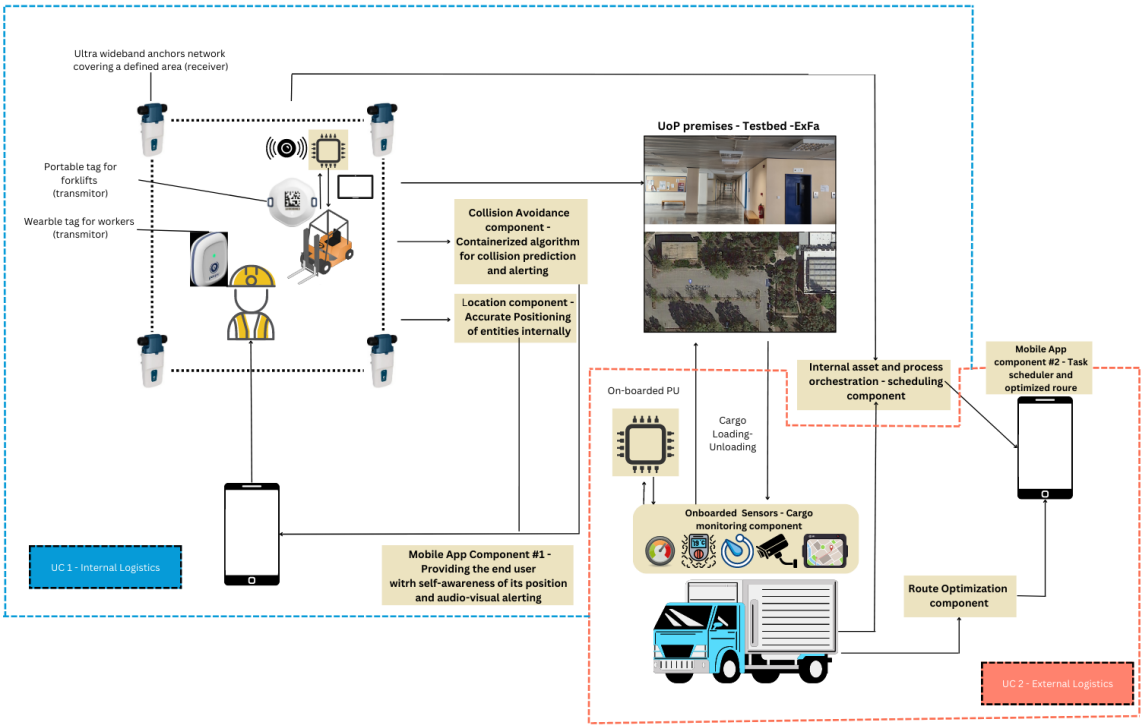

Figure 5-4: AA1 UC2 (orange frame) – Overview.

5.1.4.1 Objective

The second use case of AA1 (i.e., AA1-UC2) shares a common objective with the first use case in this application area, as they both contribute to the grant objective of AA1. More details about this objective are provided in Sections 5.1.1 and 5.1.3.1.

Co-funded by the European Union

5.1.4.2 End-user Service Components

Table 5-12 introduces a list of (software-based) end-user service components used to drive the business logic of this use case.

Table 5-12: AA1 UC2 – End-user service components.

Application Component				
ID	**Name**	**Description**	**Type**	**Deployment Domain**
APP-02-a	MariaDB	SQL DB used for the external asset positioning and the cargo sensing components	Centralized	Edge
APP-02-b	Vehicle Location	Stable component - Backend interacts with the sensors, stores data into MariaDB, and exposes services to end users via Open2.0 REST API	Centralized	Edge
APP-02-c	Routing optimization	Optimization of the routing process, offering dynamic outputs (real-time re-routing capabilities)	Centralized	Edge
APP-02-d	Cargo Monitoring	Monitoring of the loaded cargo for dynamic decision-making, loss reduction, and quality maximization	Centralized	Edge (Far-Edge)

5.1.4.3 Initial Use Case State

This section outlines the initial state of the system, before executing the AA1-UC2 workflow. This state is captured by the entries in Table 5-13.

Table 5-13: AA1 UC2 – Initial state of the system prior to the use case execution.

Initial State of the System prior to this UC
This UC assumes an available K8s cluster
A connectivity service between the available cluster and the GPS and Raspberry Pi sensors is required
Data flow among the different application components
A private 5G network that will provide high-speed, low-latency data transmission must be configured at ExFa premises.
Mobile and Admin App is properly installed and there is access to the 5G

Co-funded by the European Union

Methods to ingest data FROM sensors, IoT nodes and inventorying data at the warehouse TO the scheduling optimisation module are in place (e.g., APIs)
Methods to share the results of resource optimisation and/or the results of analytics methods are in place (e.g., APIs)

5.1.4.4 Sequence of Steps

Given that the AA1-UC2 environment is in the state described in the previous section, this section outlines the steps to execute the business logic of this use case. For every step we describe the required input, objective, and expected output, summarized in Table 5-14.

Table 5-14: AA1 UC2 – Set of steps necessary for the use case execution.

Step Number	Input	Objective	Output
S1	*Initial and real-time position of the vehicle.*	*Real-time location monitoring and input for the route optimization*	*Optimized route and data output for the internal scheduling component*
S2	*Destination of the vehicle with the intermediate stops.*	*Route optimization*	*Optimized route*
S3	*IoT sensors*	*Raspberry Pi compatible sensors measuring temperature, pressure, humidity*	*Real-time cargo monitoring. The output will be the actual metrics and the alerts in case of unusual observations*
S4	*Optical Sensing Data*	*Cameras installed in the cargo vessel will provide data for CV models.*	*Video*
S5	*Priority is given to specific packages according to any qualitative characteristics.*	*Consider any aspects of the transferred material and prioritize the delivery accordingly*	*Optimized route*
S6	*Weather conditions can be taken into account.*	*E.g., The route is scheduled in a way that parodic storms or hail are avoided*	*Optimized route*

5.1.4.5 Data Models

This section provides preliminary data models for the GPS sensor and the Raspberry Pi sensors to be used in AA1-UC2. These data models are shown in Figure 5-5 and Figure 5-6.

Co-funded by the European Union

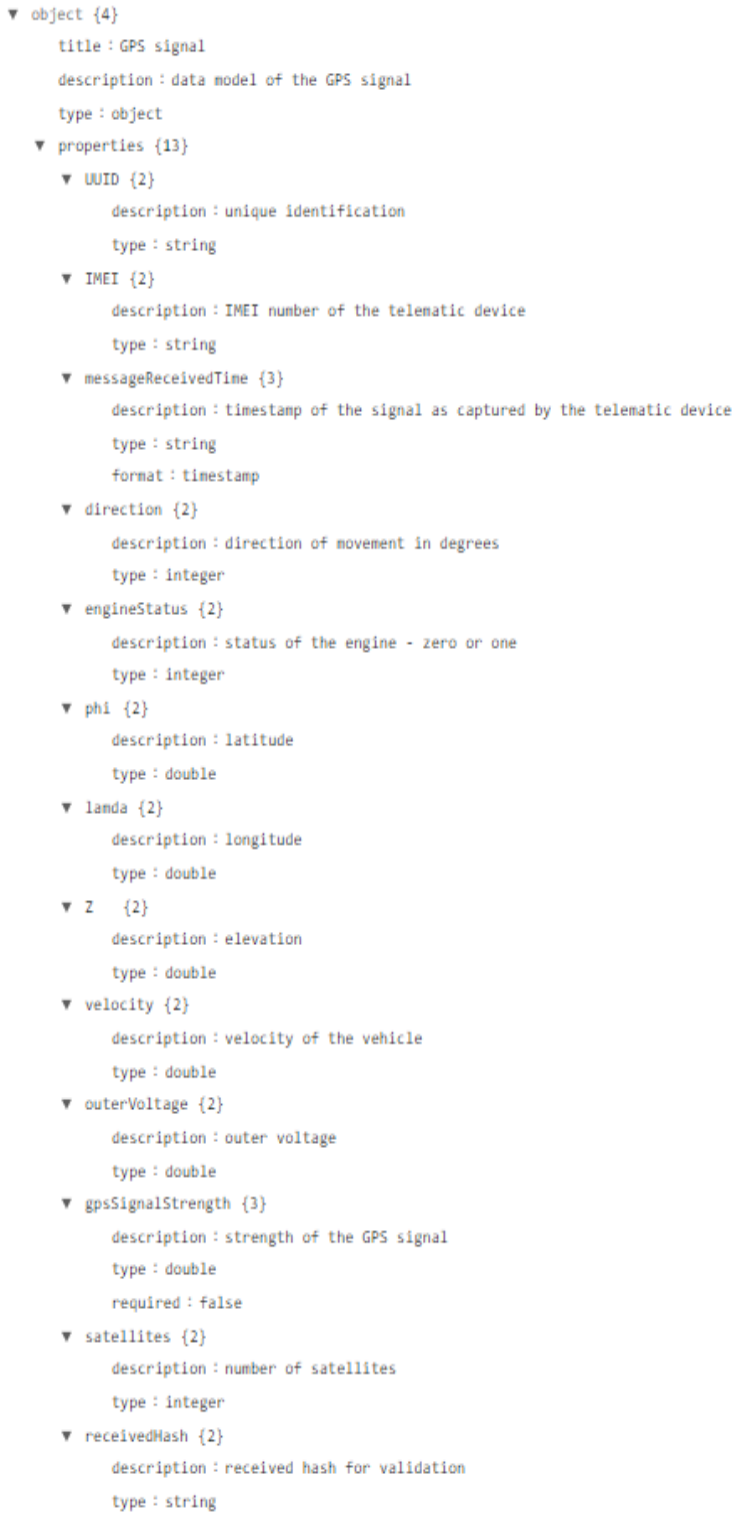

Figure 5-5: AA1 UC2 – Data model for GPS sensor to be used in this use case.

Co-funded by
the European Union

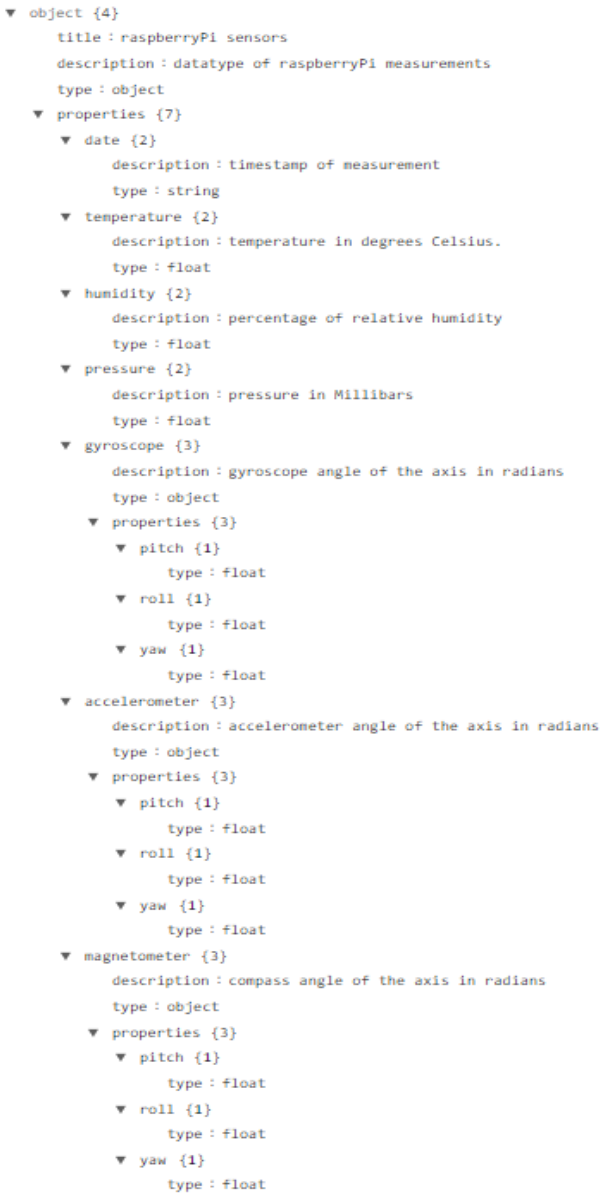

Figure 5-6: AA1 UC2 – Data model for the Raspberry Pi sensor to be used in this use case.

5.1.4.6 Workflows

This section provides the steps to reproduce the AA1 UC2 and summarizes these steps through an illustrative - yet preliminary - workflow diagram shown in Figure 5-7.

Steps

1. The position of the vehicles is taken from the GPS sensors.
2. Other parameters are loaded into the tool (type of vehicle, destination, intermediate positions, etc.).
3. The inputs are processed to the edge.
4. The optimized route is provided to the truck (driver).
5. Installation of IoT sensors on the cargo.

Co-funded by
the European Union

6. Establishing an interoperable 5G network that enables a data flow of high velocity and stability from the sensors to the edge.
7. Data processing to the edge.
8. Data transmission and storage to the core DB.
9. Data transmission and alerting system provided to the end user.

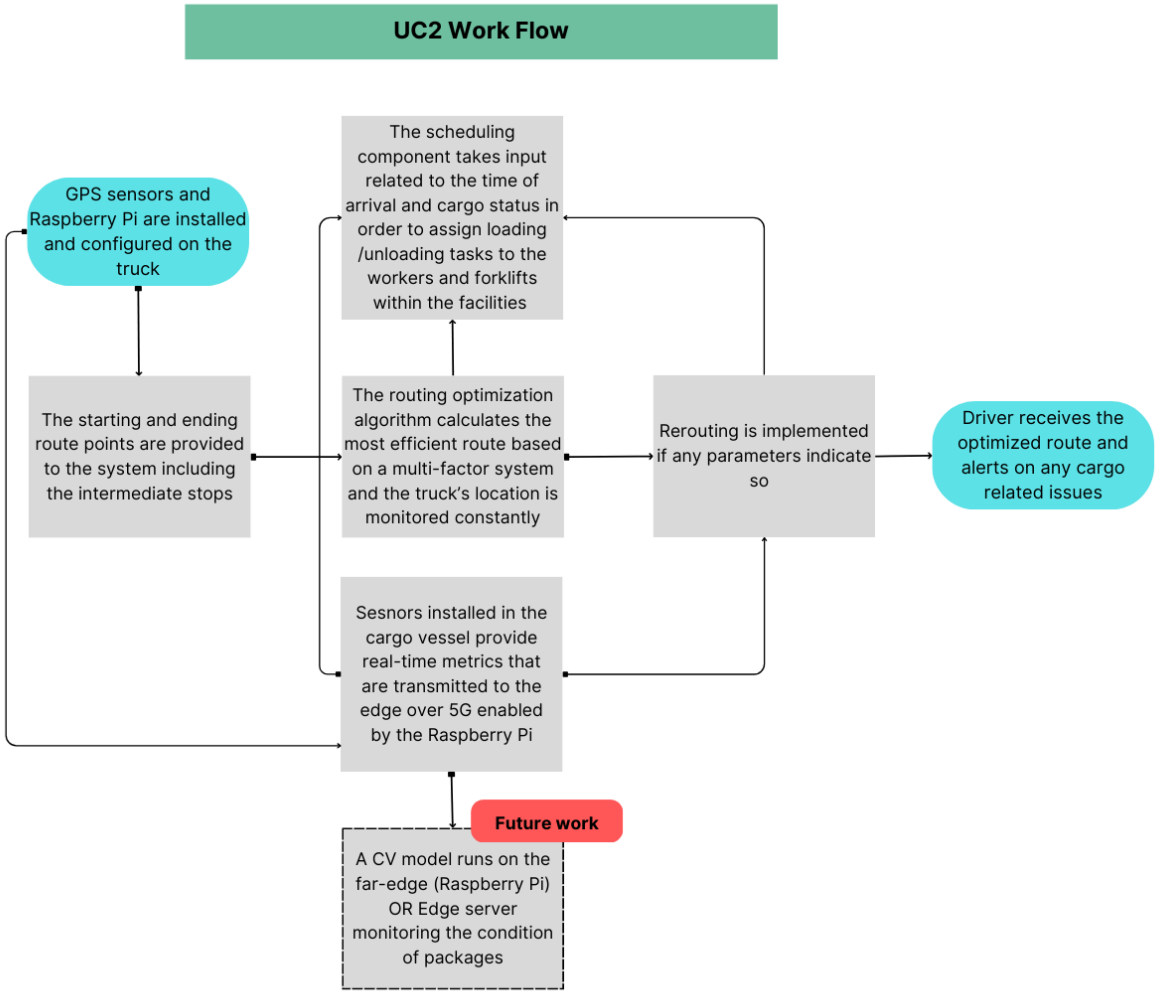

Figure 5-7: AA1 UC2 – Workflow diagram.

5.1.4.7 Requirements

This use case requires AA1.UC1.01, AA1.UC1.03, AA1.UC1.04, and AA1.UC1.05 (see Section 5.1.3.7). In addition to those, the following two requirements should also be met.

Req. Id	Requirement Description	
AA1.UC2.01	Outdoor positioning MUST utilize GPS IoT sensors with 5G capabilities	
	Affected components	IoT devices
	Contributing Partner	iLINK, UoP
	Comment	//

Co-funded by the European Union

Table 5-15: AA1.UC2.01 requirement

Req. Id	Requirement Description	
AA1.UC2.02	Component-to-component communication latency MUST be <2ms	
	Affected components	RM-L, Service components
	Contributing Partner	UoP, iLINK, ARHS, SUITE5
	Comment	//

Table 5-16: AA1.UC2.02 requirement

5.1.4.8 Potential Future Extensions

In the context of cargo monitoring, there are some potential extensions that are discussed and researched internally. One of those refers to the deployment of a CV model that is responsible for the identification of incidents where the loaded cargo drops during the transfer. For example, large boxes containing fragile equipment or devices can be closely monitored. Additionally, the volume capacity estimation within the truck through computer vision can significantly improve the cost efficiency of the delivery operations through the reduction of empty miles.

5.1.4.9 GDPR Issues

A similar approach to the UC1 design and implementation. The driver registers to the system without providing very specific information. Personal data exposure will be ensured by the optimized data collection and storage.

In particular, the purposes defined by the Customer as data processing purposes are as follows:

- Supervision of the entire fleet through an easy-to-use and ergonomic system to better organize the vehicle fleet and more efficiently manage itineraries and the operation of the traffic bureau.
- Avoidance of unnecessary stops and unnecessary movements.
- Optimal utilization of available resources and time.
- Reduction of operating costs of the corporate vehicle fleet, in particular reduction of fuel consumption.
- Direct communication with the driver.
- Checking the correct execution of the scheduled itineraries.
- Timely route update in case of need.
- Ability to adopt goals related to safe and ecological driving.
- Ability to immediately inform the driver in case it exceeds the speed limits to adopt a culture of safe driving behaviour.
- Remote control and verification that drivers, passengers, and goods move safely. Ensuring the correct temperature for refrigerated products with immediate notification in case of exceeding the limit (provided that the appropriate sensors are installed in the vehicles). Possibility for immediate intervention and coverage of emergencies.
- Protection of company property (vehicles - goods - people, e.g., possibility of finding the vehicle in case of theft).
- Ability to inform customers about the location of their service vehicle at any time.
- Checking the correct execution of what has been agreed with the customers and the drivers of the company vehicles.
- Advanced reports supporting important decisions in relation to all the above.

Co-funded by the European Union

- Preparing the company for the introduction to the new generation of tools related to travel and transportation (e.g., smart driving, etc.).

Categories of Personal Data to be processed:

The Contractor will process the following types of Personal Data in accordance with the Agreement.

Categories of simple (non-sensitive) Personal Data:

Personal data received from the telematics device and exported from the system:

- position stigma - coordinates
- possibility of driver identification with ibutton (if the service is activated)

Personal data that the Customer already has and is given the opportunity if it wishes it, to enter it in the system:

- email and cell phones to receive notifications.
- information on invoices and insurance policies.
- leasing supervisor details, insurance (code, description, name, phone, email).
- details of gas stations and workshops (description, address, number, area, postal code, TIN).
- drivers
    - name
    - telephone
    - ibutton code
    - sector
    - username and password for the navigation application
    - email
- driver-vehicle matching
- vehicle - itinerary matching

data entered by the driver from the navigation application on delivery:

- photograph
- comments
- action (I delivered; I did not deliver etc.)

5.1.4.10 Risk Assessment & Mitigation Plan

This section discusses risks that may arise while developing the AA1 UC2 and propose workarounds for mitigating those risks (see Table 5-17).

Table 5-17: AA1 UC2 – Foreseen risks and mitigation measures.

Risk	Description of risk	Likelihood	Severity	Proposed risk-mitigation measure(s)
AA1-UC2-R01	Unauthorized access or data breaches that compromise the privacy of workers and sensitive business data.	Medium	Medium	Implement robust encryption protocols, access controls, and regular security audits. Adhere to relevant data

Co-funded by the European Union

				privacy regulations and guidelines.
AA1-UC2-R02	Difficulty in integrating various technologies custom model training, 5G infrastructure, and edge computing seamlessly.	Low	High	Conduct thorough testing and validation of technology integration before deployment. Establish a dedicated integration team and ensure adequate collaboration and communication.
AA1-UC2-R03	Potential hardware malfunctions or failures of IoT devices, sensors, far-edge processors or 5G infrastructure.	Medium	High	Employ redundancy and failover mechanisms. Regularly monitor and maintain hardware components to ensure their proper functioning.
AA1-UC2-R04	The system may struggle to handle a larger volume of assets or a sudden increase in workload, affecting performance and responsiveness.	Low	High	Design the system with scalability in mind, utilizing cloud-based solutions and optimizing software architecture for efficient scaling. Perform stress tests to assess system behaviour under varying loads.
AA1-UC2-R05	Failure to comply with industry-specific or regional regulations related to IoT, data privacy, or industrial safety.	Low	Medium	Conduct a thorough regulatory analysis and ensure the system and its operations adhere to all applicable laws and standards.
AA1-UC2-R06	Resistance or reluctance from end-users to adopt and effectively use the system due to a lack of understanding or training.	Low	Medium	Implement an effective training program and provide comprehensive user documentation. Gather feedback and make necessary improvements to enhance user experience.

Co-funded by the European Union

5.2 AA2: Utilities Inspection

Application Area 2 (AA2): Utilities inspection aims to show how a legacy High-Voltage (HV) substation can be transformed into a prototype digitalized one via the use of the unifying INCODE platform and the deployment of smart utility services on top of it. AA2 leverages all the available platform technologies, such as 5G, edge computing, vertical service deployment tools, etc. to provide the services with access to the critical infrastructure through an end-to-end secure deployment environment while dealing with the hazards of a real industrial facility. Table 5-18 summarizes the proposed use cases in the INCODE Application Area 2.

Table 5-18: AA2 – Summary of use cases.

Use Case Acronym	Use Case Name
AA2-UC1	Predictive Maintenance
AA2-UC2	Intruder Detection

5.2.1 Objective

The main objectives of AA2 are summarized below:

Showcasing Advanced Industrial Services

- ✓ Showcasing the provision of advanced industrial services on top of critical infrastructure through a super-slice among core & edge services

Smart Data Analytics and Algorithms at the Edge

- ✓ Collection of industrial data from both legacy and new IoT sensing devices into an open data management platform - tackle industrial sensor heterogeneity
- ✓ Combination of data and value extraction using smart data analytics and algorithms running at the edge of the network

Deployment Over Diverse Infrastructures

- ✓ Demonstration of a deployment over diverse infrastructures (e.g., ORAN, P4, 5GC)
- ✓ Demonstration of programmable application features over IoT nodes
- ✓ Demonstration of time/event triggered apps (e.g., in case of intruder detection)

Scalability Testing

- ✓ Conduct scalability testing to ensure that the INCODE platform can handle a growing number of IoT devices and data streams while maintaining performance and reliability.

Interoperability Testing

- ✓ Test the platform's ability to interoperate with various industrial protocols and standards commonly used in critical infrastructure environments to ensure seamless integration.

Edge Intelligence Benchmarking

- ✓ Establish benchmarking criteria for evaluating the intelligence and processing capabilities of edge nodes, allowing for comparisons and performance improvements.

Co-funded by the European Union

Fault Tolerance and Disaster Recovery

- ✓ Implement mechanisms for fault tolerance and disaster recovery to maintain critical services even in the face of hardware failures or network disruptions.

User Experience Enhancement

- ✓ Focus on improving the user experience by providing intuitive interfaces for application deployment and management, making it accessible to a broader range of users, including non-technical personnel.

Data Lifecycle Management

- ✓ Develop strategies and tools for managing the entire data lifecycle, including data ingestion, storage, processing, analysis, and archival, to meet compliance and regulatory requirements.

Community Engagement

- ✓ Foster collaboration and engagement with the industrial IoT community, including businesses, researchers, and developers, to gather feedback and insights, driving continuous improvement of the INCODE platform.

Real-World Use Cases

- ✓ Identify and showcase real-world industrial use cases (e.g., predictive maintenance, quality control, asset tracking) where the INCODE platform can bring substantial value and measurable benefits.

Market Adoption Strategy

- ✓ Develop a strategy for promoting the adoption of the INCODE platform in various industrial sectors, including strategies for licensing, support, and community building.

By achieving the above objectives, AA2 can deliver the following values:

- ✓ Improvement of predictive maintenance activities
- ✓ Improvement of failure precautions
- ✓ Improvement of safety of assets and personnel
- ✓ Improvement of application interoperability

5.2.2 Why is it relevant for INCODE?

AA2 represents a pivotal aspect of the INCODE project, serving as the embodiment of its core objectives. It goes beyond the unification of heterogeneous infrastructure, illustrating the true potential of computational intelligence at the edge of the network. AA2 not only showcases the successful integration and harmonization of diverse hardware and networking protocols but also highlights the deployment of advanced computational intelligence, including AI and machine learning algorithms, near to the data sources. This strategic placement at the edge ensures reduced latency and rapid responses to critical events, such as intruder detection. AA2's significance further lies in its ability to enable a diverse array of vertical services, tailored to specific industry needs, and its cross-infrastructure compatibility ensures its applicability across various critical infrastructure types. In essence, AA2 demonstrates how the INCODE platform can make a substantial real-world impact on industrial processes, ushering in a new era of efficiency, responsiveness, and customization while future proof industrial IoT applications against technological evolution.

Co-funded by
the European Union

5.2.3 AA2-UC1: Predictive Maintenance

The main purpose of UC1 is to demonstrate monitoring services over a HV substation for the enablement of predictive maintenance activities. Through these monitoring services, data from both legacy and new industrial sensors are collected and processed at the edge and valuable insights are gained, which can be used for prediction of events in the power grid. This information can be accessed via the core cloud from both technical personnel working at the HV substation and corporate workstations at the central offices of the Transmission System Operator (TSO).

The process starts with the sensor data collection procedure. Data from low-sampling industrial sensors (i.e., Modbus devices) are collected by the Context Broker via the Modbus TCP IoT agent. Data from high-sampling devices (i.e., Phasor Measurement Unit [PMU]) are initially collected by the OpenPDC service at the edge cloud, then they are collected by the Context Broker via the JSON IoT agent. The OpenPDC service is essentially a simulated Phasor Data Concentrator (PDC) device, responsible for collecting data from PMU equipment. All data are temporarily stored and later sent to the Smart Data Analytics service, where all the data processing algorithms run. Lastly, processed data are permanently stored at the edge cloud, while accessing them is possible via the Visualization service running at the core cloud.

Figure 5-8 presents the use case architecture as described above. In Figure 5-9, the proposed deployment scenario is presented. The main points of Figure 2 are: i) the trials will take place at the lab environment of UoP campus, ii) both types of industrial devices used as data sources will run as playback device simulators fed with real data from the HV substation of interest.

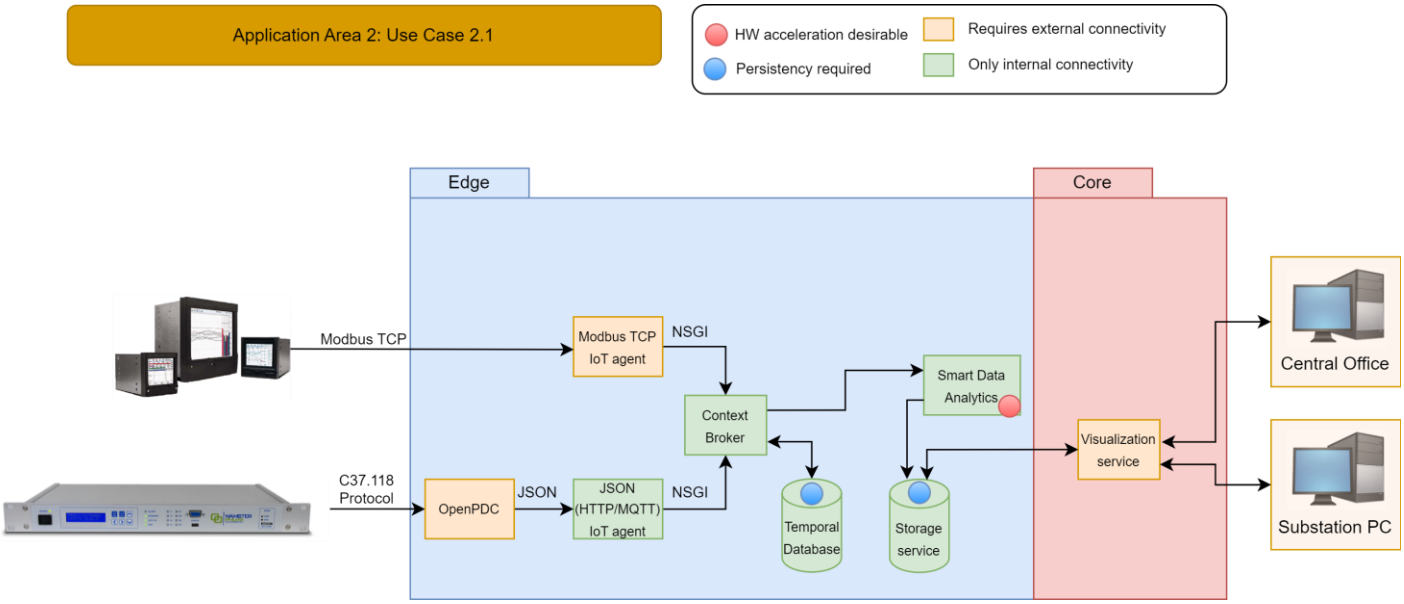

Figure 5-8: AA2 UC1 – Overview.

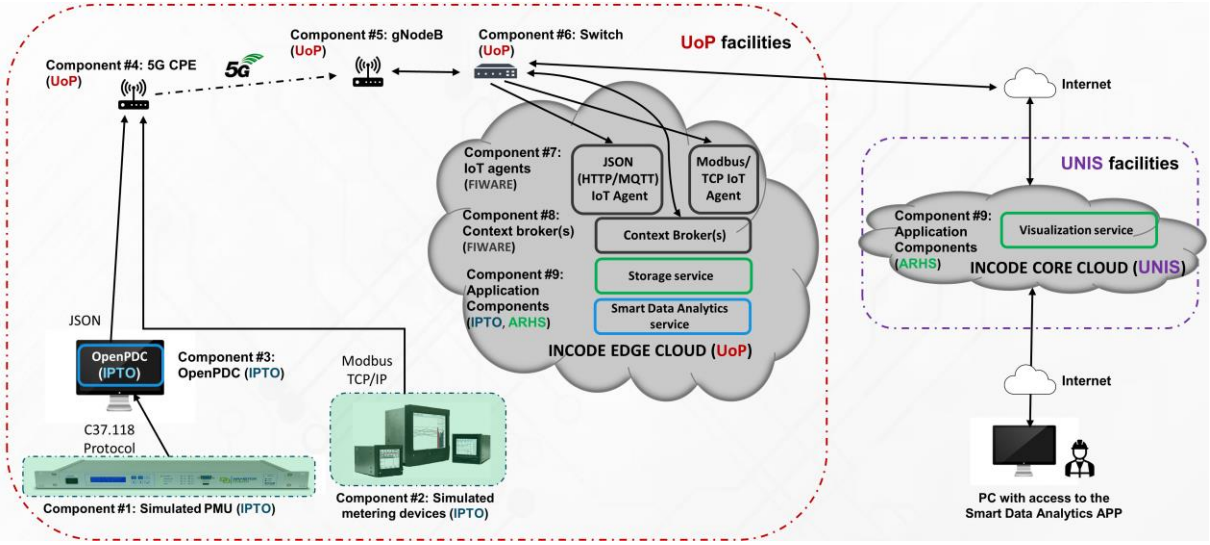

Figure 5-9: AA2 UC1 – Deployment overview.

5.2.3.1 Objective

The objective of AA2-UC1 can be summarized by the following list of sub-objectives:

1. Enabling the digitalization of legacy HV substations.
2. Real-time monitoring of industrial equipment.
3. Support for heterogeneous equipment (different communication protocols, etc.).
4. Leveraging edge intelligence to provide predictive maintenance services.

Data analytics and visualization for proper understanding of events in the power grid.

5.2.3.2 End-user Service Components

Table 5-2 introduces a list of (software-based) end-user service components used to drive the business logic of this use case.

Table 5-19: AA2 UC1 – End-user service components.

End-user Service Component				Provided by
ID	**Name**	**Description**	**Deployment Domain**	
APP-01-a	Storage service	Server to store information.	Edge cloud	ARHS
APP-01-b	Smart Data Analytics service	Service that accesses collected data from storage components, analyses them and exports output data to the storage component.	Edge cloud	IPTO
APP-01-c	Visualization service	Grafana-based UI service for data visualization	Core cloud	ARHS

Co-funded by
the European Union

5.2.3.3 Initial Use Case State

This section outlines the initial state of the system, before executing the AA2-UC1 workflow. This state is captured by the entries in Table 5-20.

Table 5-20: AA2 UC1 – Initial state of the system prior to the use case execution.

Initial State of the System prior to this UC
Interconnection between IPTO and UoP facilities must be established.
A private 5G network slice must be configured between IPTO and UoP premises.
IIoT sensors must be appropriately installed, configured and connected to the private 5G network.
Edge nodes are deployed to host the required services and their operating environment with the required resources, either pre-installed or installed on demand with the help of the INCODE platform.
The predictive maintenance services must be up and running at the INCODE edge / core cloud infrastructure.

5.2.3.4 Sequence of Steps

Given that the AA2-UC1 environment is in the state described in the previous section, this section outlines the steps to execute the business logic of this use case. For every step we describe the required input, objective, and expected output, summarized in Table 5-21.

Table 5-21: AA2 UC1 – Set of steps necessary for the use case execution.

Step Number	Input	Objective	Output
S1	*Power utility data*	*HV substation data from both legacy and newer industrial sensors are collected by the INCODE platform and are stored in a temporal database*	*Collected data temporarily stored at the edge*
S2	*Collected data temporarily stored at the edge*	*Send collected data to the smart data analytics service for processing*	*Processed data are stored in a long-term database (aka storage service)*
S3	*Processed data are stored in a long-term database (aka storage service)*	*Provide access to processed data via the visualization service for technical personnel operating in the HV substation as well as administrators*	*Visual inspection of data, graphs / plots, alarms, etc.*

Co-funded by the European Union

5.2.3.5 Data Models

This section provides a preliminary data model for an emulated Minitrend QX device to be used in AA2-UC1. This data model (see Figure 5-10) is used for the communication with other industrial devices via Modbus TCP protocol.


```
▼ object {7}
    ModbusFunctionCode : 0x03
    No_of_Measurements : 6
    deviceName : MinitrendQX_001
    ip  : ip1
  ▼ measurements [6]
    ▼ 0 {4}
        description : TR1-AMPS
        starting_address : 6337
        type : uint16
        units : A
    ▼ 1 {4}
        description : TR1-MVAR
        starting_address : 6339
        type : uint16
        units : Mvar
    ▼ 2 {4}
        description : TR1-MW
        starting_address : 6341
        type : uint16
        units : MW
    ▼ 3 {4}
        description : TR2-AMPS
        starting_address : 6343
        type : uint16
        units : A
    ▼ 4 {4}
        description : TR2-MVAR
        starting_address : 6345
        type : uint16
        units : Mvar
    ▼ 5 {4}
        description : TR2-MW
        starting_address : 6347
        type : uint16
        units : MW
    port : 502
    slaveID : 1
```


Figure 5-10: AA2 UC1 – Data model for Minitrend QX.

Other devices will also be used in this use case. Specifically, PMU-R1 communicate with PDC devices via C37.118 protocol, however no data model in the form of a JSON / YAML / XML file is provided. In our case, PMU data will be forwarded to an emulated PDC device which will then transform the incoming data to a JSON format. Since this is under development, no data model is included here at the moment.

5.2.3.6 Workflows

Figure 5-11 shows the workflow for AA2-UC1. First, the PMU connects to the emulated PDC device, while both the PDC and the Modbus devices are connected to a 5G CPE for enabling

Co-funded by the European Union

5G connectivity. Then, the orchestrator deploys all the needed services. Lastly, either technical personnel from the substation or an administrator from the central offices can connect to the application via the visualization service.

When all prior steps have been successfully run, the main operation begins. Data from the metering equipment are collected to the temporal database. Then, data are requested by the smart data analytics service for processing. After this step data are stored to the storage service, where they can be accessed via the visualization service. If a power event has been identified, appropriate alarms are sent to the visualization service, and the same process as before is followed.

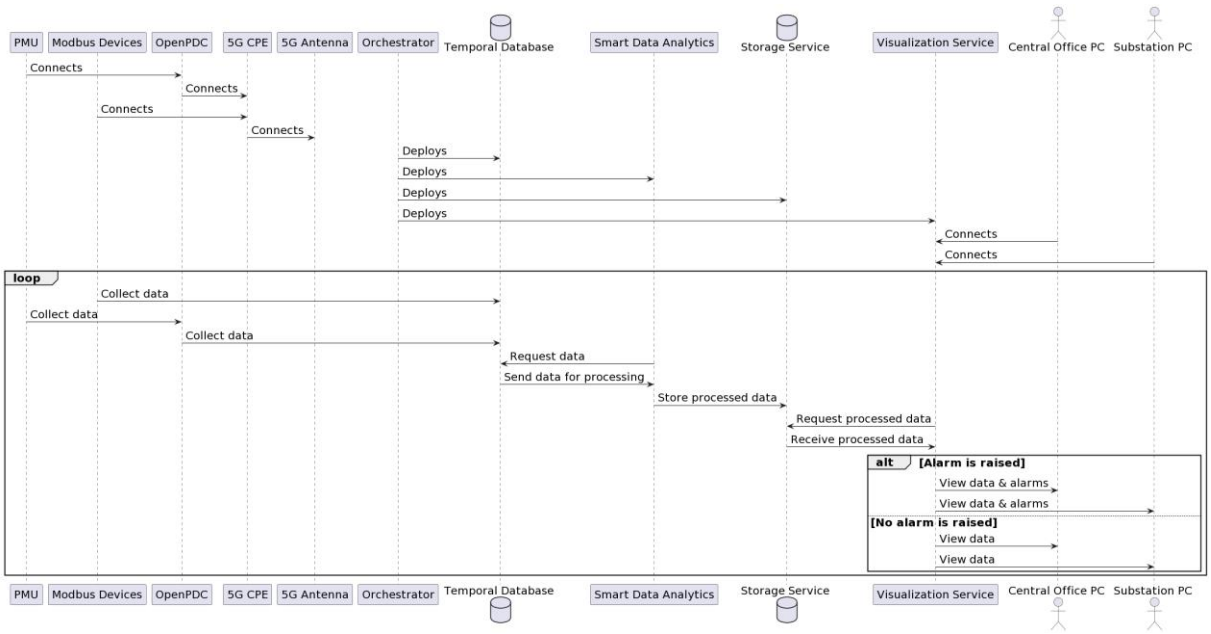

Figure 5-11: AA2 UC1 – Workflow diagram.

5.2.3.7 Requirements

Req. Id	Requirement Description	
AA2.UC1.01	Metering devices **MUST** consist of at least one legacy and one newer industrial sensing device	
	Affected components	IoT devices
	Contributing Partner	IPTO
	Comment	//

Table 5-22: AA2.UC1.01 requirement

Req. Id	Requirement Description	
AA2.UC1.02	Data from metering devices **MUST** be converted from proprietary industrial protocols to open industrial IoT protocols	
	Affected components	IoT devices
	Contributing Partner	IPTO
	Comment	//

Table 5-23: AA2.UC1.02 requirement

Req. Id	Requirement Description

Co-funded by the European Union

AA2.UC1.03	All industrial data **SHOULD** be sent at the smart data analytics application running at the UoP Edge Cloud	
	Affected components	IoT devices, Service components
	Contributing Partner	IPTO, UoP, ARHS
	Comment	//

Table 5-24: AA2.UC1.03 requirement

Req. Id	Requirement Description	
AA2.UC1.04	Smart data analytics application **MUST** be able to capture all types of industrial data, extract useful information & generate appropriate alarms for enabling predictive maintenance actions	
	Affected components	Service components
	Contributing Partner	IPTO, ARHS
	Comment	//

Table 5-25: AA2.UC1.04 requirement

Req. Id	Requirement Description	
AA2.UC1.05	Smart data analytics application **MUST** not introduce computational + networking delays exceeding 10 seconds.	
	Affected components	RM-L, Service components
	Contributing Partner	IPTO, ARHS
	Comment	//

Table 5-26: AA2.UC1.05 requirement

Req. Id	Requirement Description	
AA2.UC1.06	Smart data analytics application **MUST** prioritise the transmission of alarms to the visualization service	
	Affected components	Service components
	Contributing Partner	IPTO, ARHS
	Comment	//

Table 5-27: AA2.UC1.06 requirement

Req. Id	Requirement Description	
AA2.UC1.07	Technical personnel / Administrators **MUST** be able to access the HV substation information from the smart data analytics application via the visualization service running at the core cloud.	
	Affected components	Service components
	Contributing Partner	IPTO, ARHS
	Comment	//

Table 5-28: AA2.UC1.07 requirement

Req. Id	Requirement Description	
AA2.UC1.08	The system **MUST** be designed to allow for customization and extensibility, enabling organizations to tailor the platform to their specific industrial use cases and requirements.	
	Affected components	Service components

Co-funded by the European Union

	Contributing Partner	ARHS
	Comment	//

Table 5-29: AA2.UC1.08 requirement

Req. Id	Requirement Description	
AA2.UC1.09	User Authentication and Authorization **SHOULD** Access to HV substation information and system components SHOULD require user authentication and proper authorization. Role-based access control should be implemented to restrict access based on user roles and responsibilities.	
	Affected components	Service components
	Contributing Partner	ARHS
	Comment	//

Table 5-30: AA2.UC1.09 requirement

Req. Id	Requirement Description	
AA2.UC1.10	All components of the system, including the smart data analytics application and the visualization service, **SHOULD** adhere to relevant industry standards and protocols to facilitate interoperability and compliance with regulatory requirements.	
	Affected components	Service components
	Contributing Partner	IPTO, ARHS
	Comment	//

Table 5-31: AA2.UC1.10 requirement

Req. Id	Requirement Description	
AA2.UC1.11	Technical personnel and administrators **MUST** have remote access to the smart data analytics application for monitoring and controlling HV substation facilities. This access should include the ability to adjust settings and configurations as needed.	
	Affected components	Service components
	Contributing Partner	ARHS
	Comment	//

Table 5-32: AA2.UC1.11 requirement

Req. Id	Requirement Description	
AA2.UC1.12	Adequate user training and ongoing technical support **MUST** be provided to technical personnel and administrators to ensure they can effectively operate and maintain the system.	
	Affected components	Service components
	Contributing Partner	IPTO, ARHS
	Comment	//

Table 5-33: AA2.UC1.12 requirement

Co-funded by the European Union

5.2.3.8 Potential Future Extensions

Future developments of this use case are poised to usher in a new era of industrial IoT innovation. One promising avenue is the enhancement of edge computing capabilities, with a focus on integrating more advanced AI and machine learning algorithms directly at the edge. This will enable even more sophisticated real-time data analysis and decision-making, improving the overall responsiveness of the system. Additionally, ongoing research and development efforts may lead to the optimization of predictive maintenance techniques, resulting in more precise predictions and reduced downtime. This could involve integrating condition-based monitoring systems and advanced failure prediction models. Furthermore, as IoT technology continues to evolve, advancements in sensor technology are expected. These developments could lead to more accurate and versatile sensors, enabling a deeper level of insight and control over critical infrastructure. Overall, the future developments in this use case hold the potential to revolutionize industrial operations, making them more efficient, reliable, and cost-effective.

5.2.3.9 GDPR Issues

No GDPR issues are raised in this use case since no personal data are used.

5.2.3.10 Risk Assessment & Mitigation Plan

This section discusses risks that may arise while developing the AA2 UC1 and propose workarounds for mitigating those risks (see Table 5-34).

Table 5-34: AA2 UC1 – Foreseen risks and mitigation measures.

Risk	Description of risk	Likelihood	Severity	Proposed risk-mitigation measure(s)
AA2-UC1-R01	Smart Data Analytics service fails to achieve time constraints.	Medium	Medium	Use hardware AI accelerator.
AA2-UC1-R02	The system may struggle to handle a larger volume of assets or a sudden increase in workload, affecting performance and responsiveness.	Low	High	Design the system with scalability in mind, utilizing cloud-based solutions and optimizing software architecture for efficient scaling. Perform stress tests to assess system behaviour under varying loads.

5.2.4 AA2-UC2: Intruder Detection

This use case allows to increase the security level in the HV substation facilities by detecting all intruders. If an alarm is raised because of an intruder, an UAV could be deployed to follow the intruder and always know its position.

Co-funded by the European Union

The cameras in the facilities stream real time video to the network application. The AI models will receive that video footage and detect any people in there. If a person is detected, the face will be compared with all the ones stored in the database of authorized personnel. If the person detected is in the database, a message requesting confirmation will be sent to the Authentication App of that person. If it is confirmed the location of the worker will be placed in a map for the security cabin.

In the case that the intruder is not in the database, or the worker do not confirm its identity. An alarm will be sent to the security cabin where the guard could deploy an UAV. If it is accepted, the UAV will go to the position of the intruder and will follow him including a sound alarm.

The guards in the security cabin will have an interface where they can see the video feeds from the cameras and UAV with the detections and a map locating all the detections and their identity.

Figure 5-12 presents the use case architecture. It can be seen how the camera and the UAV send video for the intruder detector and its visualization in the security cabin. Also, the smartphone App that allows authorized worker to manage its profile and confirm its identity.

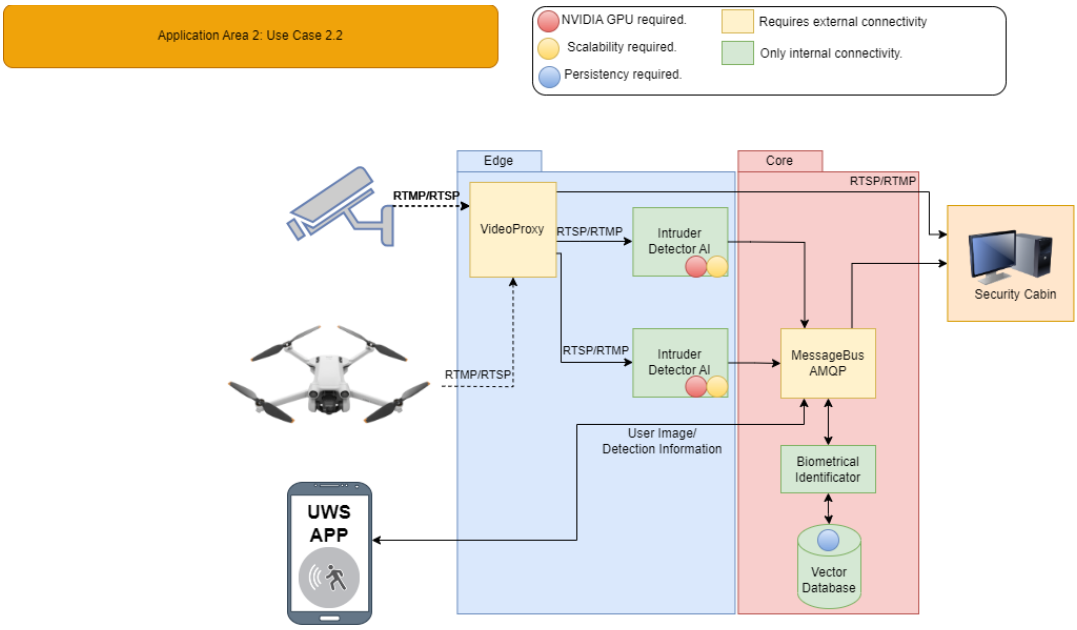

Figure 5-12: AA2 UC2 – Overview.

A complementary proof-of-concept (PoC) will also be developed in this use case to evaluate the theoretical work on the new machine learning (ML)-based resource management algorithm development. The aim is to validate the enhancement of the system's efficiency by applying a customized version of the developed resource management algorithm.

Co-funded by the European Union

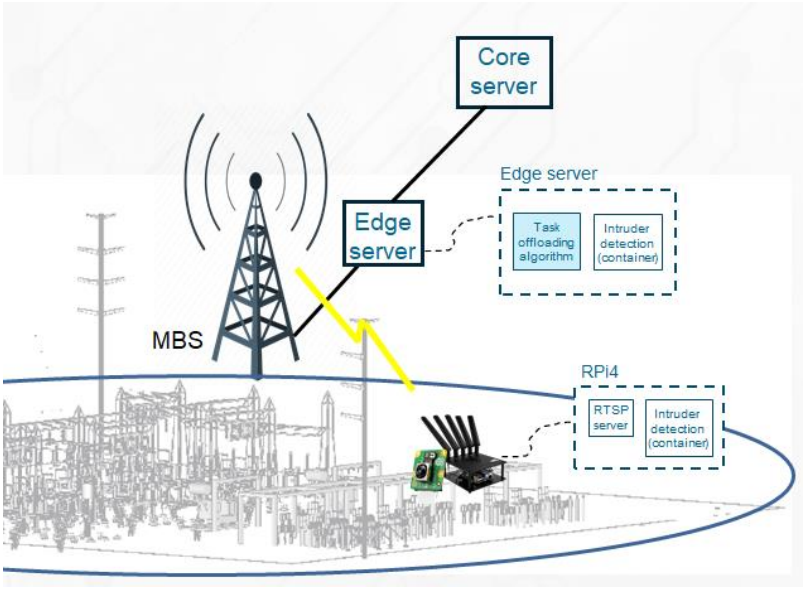

Figure 5-13: AA2 UC2 – Task offloading PoC overview.

Figure *5-13* shows the overview of the task offloading PoC. The simple human/intruder detection task can be performed on either the IoT device (RPi4) or the edge server. The task offloading algorithm, a customized version of the ML-based resource management algorithm, will decide where to run the human/intruder detection task based on the QoS requirements, as well as resource and power consumption of the IoT device and the edge server. Lastly, in Figure *5-14*, the proposed deployment scenario is presented. The main point of the figure is that the trials will take place at the lab environment of UoP campus.

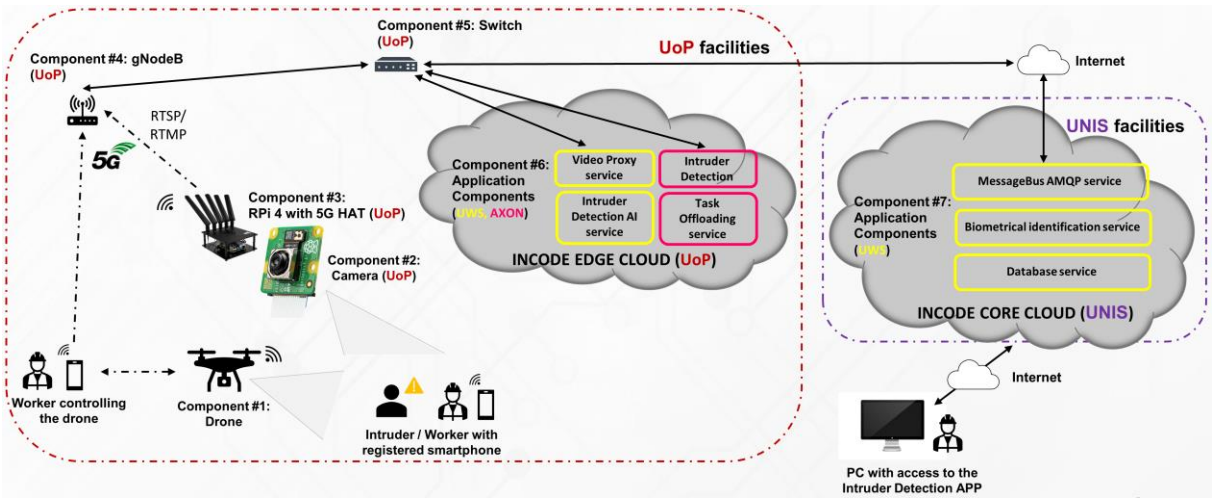

Figure 5-14: AA2 UC2 – Deployment overview.

5.2.4.1 Objective

The objective of AA2-UC2 can be summarized by the following list of sub-objectives:

1. To achieve a high-level security in the HV substation facilities by detecting and tracking any intruder.
2. Real time video streaming from the UAV and the cameras to the Intruder Detector AI service.
3. High accuracy person and face detector from the UAV and cameras.

Co-funded by the European Union

4. Creation of a smartphone App that allows authorized personnel to register, delete or update their profile and confirm their identity when they have been identified.
5. Enable real time monitoring of the facilities through video and map in the security cabin.
6. Tracking of the intruder using a UAV deployed by the security guards in the cabin.
7. To develop a task offloading PoC for evaluating the ML-based resource management algorithm.

5.2.4.2 End-user Service Components

Table 5-35 introduces a list of (software-based) end-user service components used to drive the business logic of this use case.

Table 5-35: AA2 UC2 – End-user service components.

End-user Service Component				Provided by
ID	Name	Description	Deployment Domain	
APP-01	VideoProxy	Nginx-based server. Streams video.	Edge cloud	UWS
APP-02	VectorDB	Server to store information.	Core cloud	UWS
APP-03	MessageBus	RabbitMQ-based server to exchange information: detections, user images, GPS coordinates, authentications.	Core cloud	UWS
APP-04	Intruder Detector AI	Service to detect intruders in the HV substation from video stream. GPU compliant.	Core cloud	UWS
APP-05	Biometrical Identificator	Service to identify the intruders in the HV substation comparing with the registered ones.	Core cloud	UWS

5.2.4.3 Initial Use Case State

This section outlines the initial state of the system, before executing the AA2-UC1 workflow. This state is captured by the entries in Table 5-36.

Table 5-36: AA2 UC2 – Initial state of the system prior to the use case execution.

Initial State of the System prior to this UC
The HV substation facilities are equipped with at least one security camera.
Cameras are streaming video to the UoP Edge Cloud
There is at least one UAV in the HV substation facilities.

Co-funded by the European Union

The authorized personnel have registered in the Authentication Smartphone App.
The HV substation facilities are interconnected with the UoP Edge Cloud
A 4G/5G network is existing in the search area.
Edge nodes are deployed to host the required AI models and their operating environment with the required resources, either pre-installed or installed on demand with the help of the INCODE platform.

5.2.4.4 Sequence of Steps

Given that the AA2-UC2 environment is in the state described in the previous section, this section outlines the steps to execute the business logic of this use case. For every step we describe the required input, objective, and expected output, summarized in Table 5-37 and Table 5-38.

Table 5-37: AA2 UC2 – Set of steps necessary for the use case execution.

Step Number	Input	Objective	Output
S1	*UAV, UWS App and CCTV cameras starts.*	*Connect to antenna.*	*Successful connection.*
S2	*Orchestrator deploys VNFs*	*VNFs running*	*VNFs successfully running.*
S3	*Worker sends their information to register using the UWS App*	*Register a worker in the system*	*Worker registered in the system*
S4	*Video streaming from the CCTVs cameras.*	*Video is streamed from the cameras and received by Intruder Detector and the Security Cabin*	*Video watched in the Security Cabin and received by the Intruder Detector.*
S5	*Intruder detected by the Intruder Detector*	*Check if the intruder is stored in the Vector Database (worker) or not.*	*If the intruder is a worker or not.*
S6	*Confirmation of identity by a worker*	*Confirm that the identified workers are themselves*	*Confirmation of the identity of the intruder.*
S7	*Alarm raised by the Biometrical Identifier*	*Inform the Security Cabin that an intruder has entered the perimeter.*	*Security cabin notified of the intruder*
S8	*UAV deployment authorization by the Security Cabin*	*Authorize the deployment of an UAV to start following the intruder*	*The intruder will be followed by an UAV*

Co-funded by the European Union

Table 5-38: AA2 UC2 – Set of steps necessary for the execution of the task offloading scenario.

Step Number	Input	Objective	Output
S1	*Video stream*	*Start human detection app in IoT node*	*App successfully running in IoT node*
S2	*Optimization requirement(s)*	*Set optimization objective(s)*	*Optimization objective set*
S3	*Infrastructure and application resource usage*	*Monitor resource availability and usage of IoT node*	*Resource monitoring data*
S4	*Infrastructure and application metrics; ML model*	*Dynamic execution of task offloading algorithm*	*Task offloading decision*
S5	*Decision to offload task (I.e., change deployment)*	*Migrate human detection app to edge server*	*App successfully running in edge server*

5.2.4.5 Data Models

This section provides preliminary data models for the DJI Mini 2 (UAV) and a 5G-enabled Raspberry Pi-based camera to be used in AA2-UC2. These data models are shown in Figure 5-15 and Figure 5-16 respectively.

Co-funded by the European Union

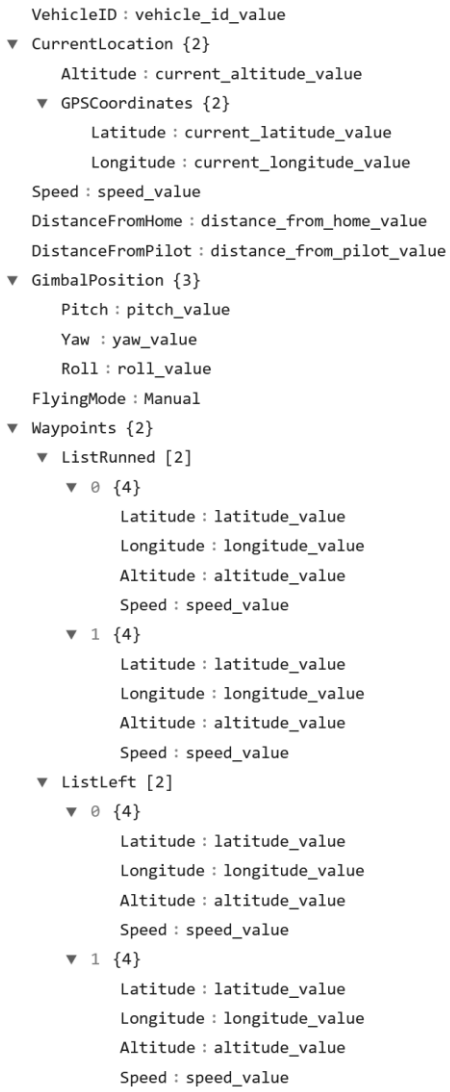

Figure 5-15: AA2 UC2 – Data model for the DJI Mini 2 (UAV) to be used in this use case.

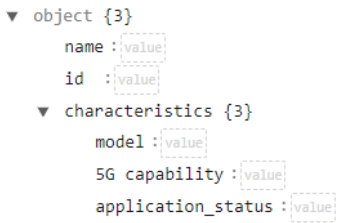

Figure 5-16: AA2 UC2 – Data model for the 5G-enabled Raspberry Pi-based camera.

5.2.4.6 Workflows

Figure 5-17 shows the main workflow for AA2 UC2: Intruder detection. First, the UAV, the CCTV camera and the UWS App connect to the 5G antenna. Then, the orchestrator deploys all the VNFs. After that, a worker can register in the system through the UWS App, that will send it to the message bus to be obtained by the Biometrical Identifier that will store in the Vector Database.

When the workers are registered in the system, the CCTV cameras can start streaming video to be received by the Video Proxy to be watched in the Security Cabin. The Intruder detector

Co-funded by the European Union

will process the streaming to detect any intruder. If any is detected, it will identify using the Biometrical Identifier. If the intruder is already in the database, a message will be sent to that worker to confirms its identity. If it is confirmed the position of the worker will be shown in the security cabin. If it is not confirmed an alarm will be raised and it will be received in the security cabin, that could authorize the deployment of an UAV to start following the intruder. If the intruder identity is not in the database, an alarm will be raised directly, and the same process as before will be followed.

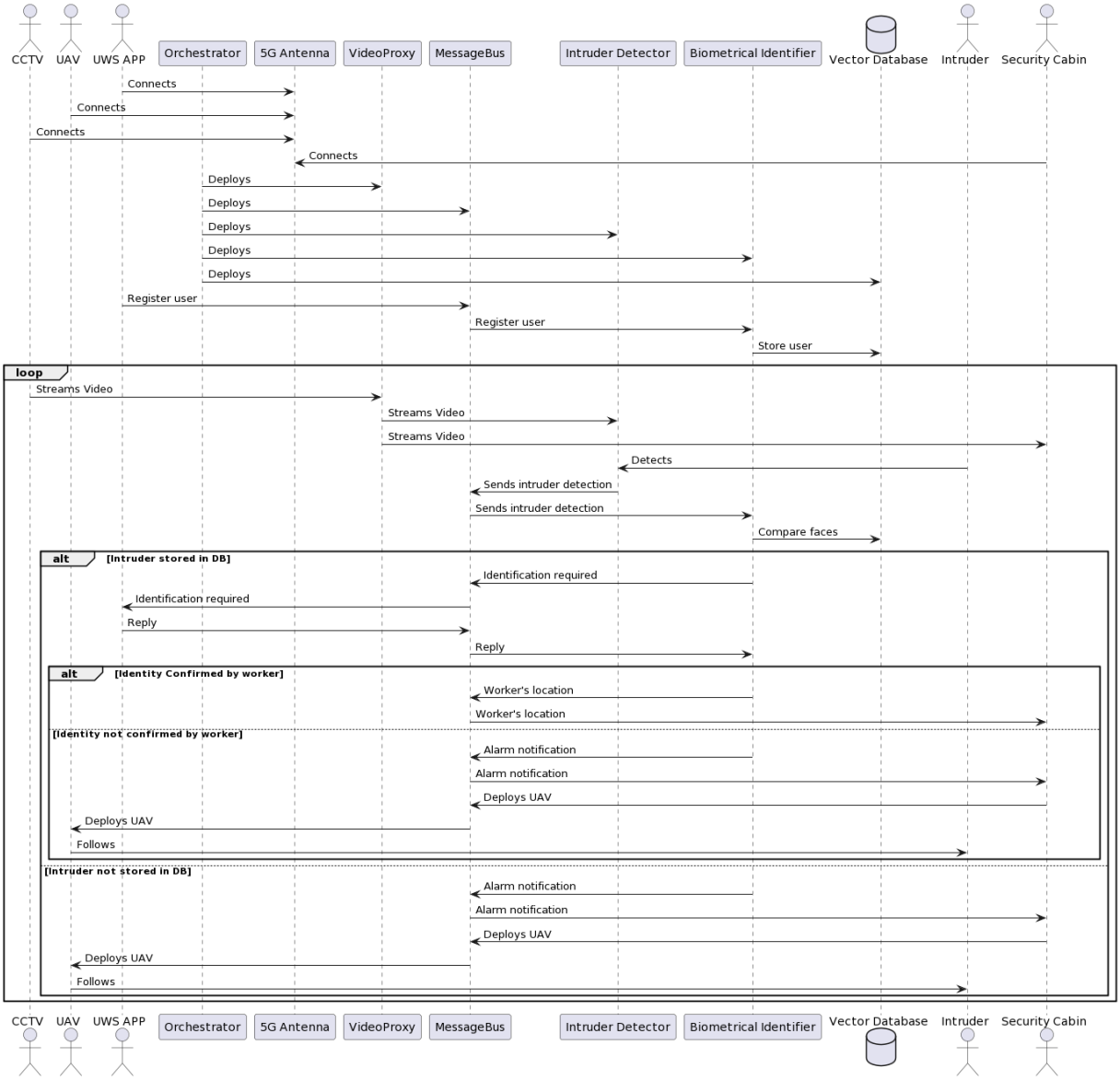

Figure 5-17: AA2 UC2 – Main workflow diagram.

Figure 5-18 shows the workflow for the dynamic task offloading scenario of this use case. First, camera streams video to the human detection app running as a container in the RPi. Input data for the current timeslot such as the resource consumption, power consumption, latency, etc. are collected by the monitoring app and fed to the task offloading decision algorithm. The trained ML-based algorithm will make decision whether to offload the task to the edge server or not. If the decision is to offload the task, video will be streamed to the edge server and the human detection application will be migrated to the edge server.

Co-funded by the European Union

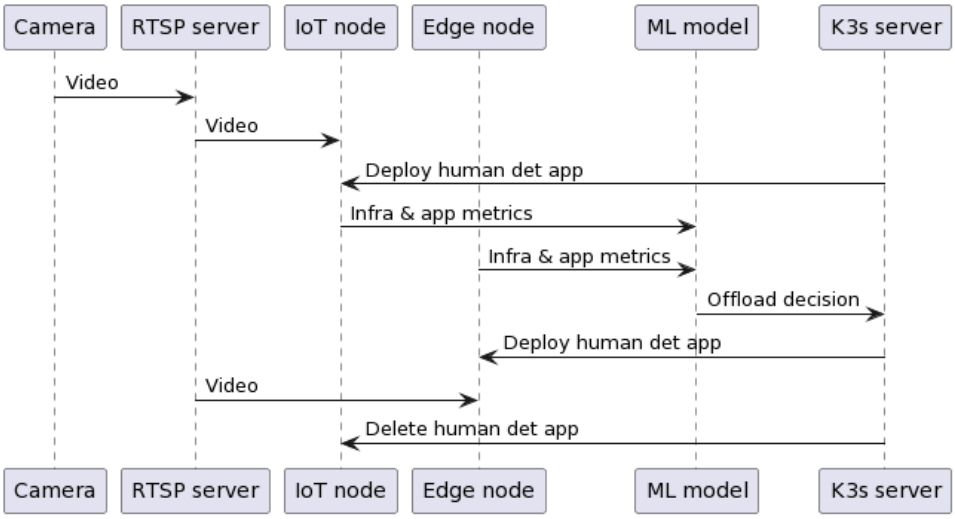

Figure 5-18: AA2 UC2 – Workflow diagram for task offloading scenario.

5.2.4.7 Requirements

Req. Id	Requirement Description	
AA2.UC2.01	The testbed **MUST** have at least one security camera	
	Affected components	IoT devices
	Contributing Partner	UoP
	Comment	//

Table 5-39: AA2.UC2.01 requirement

Req. Id	Requirement Description	
AA2.UC2.02	Biometrical Identificator **MUST** be able to generate appropriate alarms when an unauthorized person enters the HV substation facilities	
	Affected components	Service components
	Contributing Partner	UWS
	Comment	//

Table 5-40: AA2.UC2.02 requirement

Req. Id	Requirement Description	
AA2.UC2.03	Biometrical Identificator **MUST** be able to identify if the identity of a detected person is stored in the database as authorized personnel	
	Affected components	Service components
	Contributing Partner	UWS
	Comment	//

Table 5-41: AA2.UC2.03 requirement

Req. Id	Requirement Description	
AA2.UC2.04	Intruder Detector AI **MUST** be able to detect people in the testbed facilities	
	Affected components	Service components
	Contributing Partner	UWS

Co-funded by the European Union

	Comment	//

Table 5-42: AA2.UC2.04 requirement

Req. Id	Requirement Description	
AA2.UC2.05	Drones **MUST** be able to send video to the UoP Edge Cloud	
	Affected components	IoT devices
	Contributing Partner	UWS, UoP
	Comment	//

Table 5-43: AA2.UC2.05 requirement

Req. Id	Requirement Description	
AA2.UC2.06	Drones **COULD** be deployed in case an intruder detection alarm is raised	
	Affected components	IoT devices
	Contributing Partner	UWS
	Comment	//

Table 5-44: AA2.UC2.06 requirement

Req. Id	Requirement Description	
AA2.UC2.07	Technical personnel **MUST** be registered to the INCODE platform via the Authentication Smartphone App	
	Affected components	Service components, IoT devices
	Contributing Partner	UWS, UoP
	Comment	//

Table 5-45: AA2.UC2.07 requirement

Req. Id	Requirement Description	
AA2.UC2.08	Technical personnel **COULD** delete or update their profile in the Authentication Smartphone App	
	Affected components	Service components, IoT devices
	Contributing Partner	UWS, UoP
	Comment	//

Table 5-46: AA2.UC2.08 requirement

Req. Id	Requirement Description	
AA2.UC2.09	Technical personnel **MUST** be able to confirm a personal identity identification via the Authentication Smartphone App	
	Affected components	Service components, IoT devices
	Contributing Partner	UWS, UoP
	Comment	//

Table 5-47: AA2.UC2.09 requirement

Req. Id	Requirement Description
AA2.UC2.10	Security Cabin **MUST** have an interface to show the video with the detections

Co-funded by the European Union

	Affected components	Testbed
	Contributing Partner	UoP
	Comment	//

Table 5-48: AA2.UC2.10 requirement

Req. Id	Requirement Description	
AA2.UC2.11	Security Cabin **COULD** deploy drones when an intruder is detected	
	Affected components	Service components, IoT devices
	Contributing Partner	UWS
	Comment	//

Table 5-49: AA2.UC2.11 requirement

Req. Id	Requirement Description	
AA2.UC2.12	Security Cabin **MUST** have a map showing the position of the detections	
	Affected components	Service components
	Contributing Partner	UWS
	Comment	//

Table 5-50: AA2.UC2.12 requirement

5.2.4.8 Potential Future Extensions

The intelligent resource management and task offloading mechanism can be integrated into a more advanced intruder detection use-case in the future. Additional inputs such as wireless channel gain and network resources data can be included in the intelligent resource management framework.

5.2.4.9 GDPR Issues

To ensure compliance with GDPR, several measures have been implemented for this use case:

- Data Encryption: The faces of workers will be securely stored in the vector database using state-of-the-art encryption techniques, rendering the data non-readable to unauthorized personnel. This safeguards sensitive facial data from unauthorized access or breaches.
- Data Anonymization: Users' identities will be protected by storing their information in the database using unique identification codes rather than personal information. This ensures that there is no direct link between a user's facial data and their identity, enhancing privacy.
- Data Minimization: Only essential information required for the use case will be collected from workers. This includes limited and non-sensitive information, with no request for contact details, birthdates, or other confidential data. This practice adheres to the GDPR's principle of collecting only necessary data.

Additionally, in compliance with GDPR guidelines:

- Privacy Policy: Workers will be required to accept a comprehensive privacy policy during the registration process. This policy will explicitly state that their facial data will be stored in a vector database and will only be used for verification purposes within the substation facility.

Co-funded by the European Union

- Volunteer Consent: Individuals volunteering for data collection will be asked to provide explicit consent through a consent form. This form authorizes the use of their facial images for research and dissemination purposes, ensuring compliance with GDPR consent requirements.
- Data Deletion: All volunteers and workers will have the right to request the deletion of their data at any time. Such requests will be processed immediately, and their data will be permanently removed from the system.
- Notification Signs: In areas where video recording takes place, such as substations or testing facilities, conspicuous signs will be displayed to inform individuals that video recording is in progress. This practice aligns with GDPR transparency requirements and informs individuals of data processing activities.

By implementing these measures and adhering to GDPR guidelines, we ensure the protection of individuals' privacy rights while maintaining the necessary functionality for the specified use case.

5.2.4.10 Risk Assessment & Mitigation Plan

This section discusses risks that may arise while developing the AA2 UC1 and propose workarounds for mitigating those risks (see Table 5-51).

Table 5-51: AA2 UC2 – Foreseen risks and mitigation measures.

Risk	Description of risk	Likelihood	Severity	Proposed risk-mitigation measure(s)
AA2-UC2-R01	UAV is not authorized to fly in that area.	Medium	Medium	The intruder will be followed using different CCTVs cameras.
AA2-UC2-R02	A worker fails to confirm its identity and is detected as an intruder.	High	Low	Security cabin will check the intruder face and the one stored and can rectify the mistake. Or if after a time is again detected as a worker, he can confirm its identity then.
AA2-UC2-R03	Weather condition not acceptable for UAVs	High	Low	The intruder will be followed using different CCTVs cameras or the testing can be moved to another suitable date.
AA-UC2-R04	Worker detected as intruder.	Low	Low	Security cabin will check the intruder face and the one stored and can rectify the mistake.
AA2-UC2-R05	Lack of resources to run the human detection app in the IoT node	Low	Medium	Use hardware AI accelerator.

Co-funded by the European Union

5.3 AA3: Smart Factories – Intelligent Worker Assistance

Application Area 3 is deployed at the MADE Competence Centre Industry 4.0 testbed. The primary focus of the application area is to showcase the ability if the INCODE platform to be swiftly deployed in Industrial setting. The application Area is primarily focused on Human Centric Manufacturing Solutions. The Application Area will leverage data feeds from both Production assets and wearable sensors to dynamically adapt and mobilize the production assets like Collaborative Robots and assembly stations to provide in situ support to the operators thereby promoting operator health and safety while ensuring leaner manufacturing.

The Application Area hosts two use cases which are closely related to the Operator 4.0 domain and leverage wearable sensor technology. The use cases and their respective application domains are as follows:

- **Healthy operator** – Which deals with collaborative robotics.
- **Human in loop** - Which deals with operator stress based adaptive production lines and lean manufacturing.

Table 5-1 summarizes the proposed use cases in the INCODE Application Area 3.

Table 5-52: AA3 – Summary of use cases.

Use Case Acronym	Use Case Name
AA3-UC1	Healthy Operator
AA3-UC2	Human in Loop

5.3.1 Objective

The objective is to demonstrate how the INCODE platform can be quickly deployed in a smart factory to retrieve and exploit data from wearable sensors to improve the physical and mental wellbeing of the operators through timely interventions driven by AI based applications. This industrial use case presents a significant challenge because of the diversity of interfaces and protocols that currently exist in the manufacturing space. The primary objectives of the Application area are as follows:

- **Interfacing with diverse industrial Protocols**: On the Made testbed we would like to demonstrate the how potential Industrial stakeholders can Interface with the diverse Industrial devices and systems like MES, Industrial robots, etc. We also demonstrate the integration of wearable sensors with partly closed communication protocols to be used for the development of Operator 4.0/Industry 5.0 solutions.
- **Deployment of Private 5G**: Private 5G infrastructure will be leveraged as an enabling technology which will be used to provide secure and reliable connectivity between the industrial assets and IoT devices. This will also promote private 5G as an enabler of Industry 5.0 technologies in the manufacturing space.
- **Edge AI for Operator wellbeing:** The application area will demonstrate the deployment of at Edge ML/AI solutions for improvement of operator wellbeing and safety while promoting lean manufacturing at the same time.

Co-funded by the European Union

- **Data Privacy:** Since wearable sensor data from operators will be utilized the use case necessitates the use of adequate measures to anonymize the operator data. Therefore, the use case also serves as a template for the adoption of wearable sensors in industry while maintaining GDPR compliance. Details about GDPR issues are discussed in Section 5.3.3.9 and 5.3.4.9.

5.3.2 Why is it relevant for INCODE?

The MADE test bed depicts the situation of a typical smart manufacturing plant consisting of a complex mix edge infrastructure like IoT devices, Production Assets Edge computation devices etc. which in turn rely upon a wide range of interfaces and communication protocols. From an implementation and validation perspective, this makes it highly relevant in realizing The INCODE vision which aims at creating a trusted cloud-native platform which helps in solving the emerging dynamicity of distributed and heterogeneous Private edge Infrastructure.

Furthermore, the testbed is a real-life depiction of the challenges and opportunity presented by Operator 4.0 technologies. The Application address some of the key challenges namely data privacy aspect of the Operator 4.0 paradigm. As such the use cases act as templates to be used and exploited by potential stakeholders in the manufacturing sector who wish to adopt Operator 4.0 technologies in their factories. This acts as a self-driving force helping to maximize the impact and adoption of INCODE platform.

5.3.3 AA3-UC1: Healthy Operator

The proposed scenario reflects the work of a human operator, who is performing a repetitive task of material handling while equipped with an exoskeleton- that helps releasing muscle effort, and electromyographic wearable sensors- to detect muscle fatigue. As soon as a stressful condition is detected in the operator, the Cobot is reinstructed to apply more “intrusive” tasks to diminish the operator’s handling movements and/or send an alert to the operator.

Figure 5-19 shows the topology of the use case including hardware as well as application components. The alpha numeric keywords in the brackets correspond to the Component/Hardware IDs described in relevant sections of the remaining text.

Co-funded by
the European Union

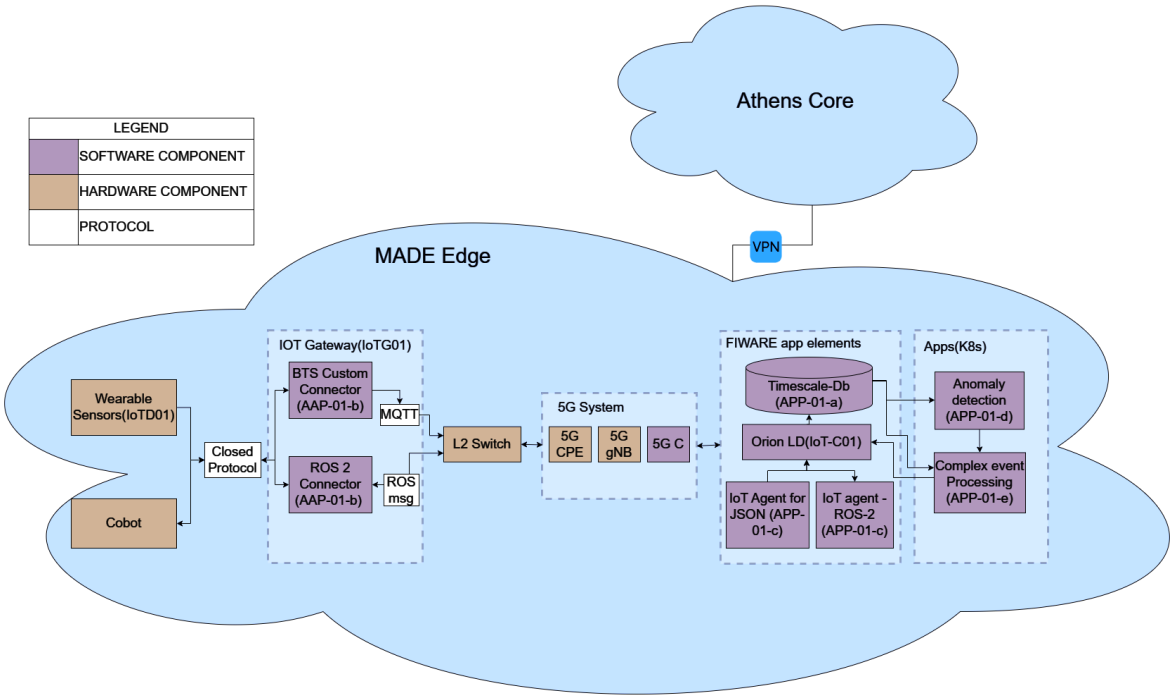

Figure 5-19: AA3 UC1 – Overview.

5.3.3.1 Objective

Apart from the general objectives of AA3 the Use case 1 specifically has the following additional objectives.

- Lays down a foundation for the use of Human Cobot collaboration under industrial setting under the umbrella of INCODE Project.

Present Operator 4.0 solution which aids and helps prevent musculoskeletal injuries on account of repetitive work situations.

5.3.3.2 End-user Service Components

Table 5-53 introduces a list of (software-based) end-user service components used to drive the business logic of this use case.

Table 5-53: AA3 UC1 – End-user service components.

End-User Service Component				Provided by
ID	**Name**	**Description**	**Deployment Domain**	
APP-01-a	Database (Timescale DB)	Stores data from the sensors and the cobot state related info	Edge	FIWARE
APP-01-b	Connectors	Set of intermediate applications which further connect to IoT drivers.	Edge	POLIMI

Co-funded by the European Union

APP-01-c	Drivers	Drivers for IoT and Robotics devices to allow communication to Orion LD - IoT-C01	Edge	FIWARE
APP-01-d	Anomaly detection	Detects anomalies in the sensor data to detect high physical stress condition of the worker	Edge	POLIMI
APP-01-e	Complex event Procession	Uses data from the sensors and the Cobot state information to generate specific instructions to the Cobot	Edge	POLIMI

*Pre-existing solution from FIWARE Catalog

5.3.3.3 Initial Use Case State

This section outlines the initial state of the system, before executing the AA3-UC1 workflow. This state is captured by the entries in Table 5-54.

Table 5-54: AA3 UC1 – Initial state of the system prior to the use case execution.

Initial State of the System prior to this UC
The sensors are fully charged to ensure no outage during the trial
The operators have been educated in advance about the work instructions.
The operators are wearing the sensors and the exoskeleton properly. Here special care needs to be taken to ensure that the sensor is in proper contact with the skin and the location of the placement is at a pre decided location which is suitable for accurate and repeatable measurements
All hardware components, like IOT gateway, edge processor etc. are booted up and ready to receive the data stream.
Connectivity between BTS EMG sensors (IoT -D01) and IoT Gateway (IoT-G01)
This UC assumes an available K8s cluster running the Application elements running on commodity server CS01
A connectivity service between the available cluster and BTS EMG sensors (IOT-D01) is required

5.3.3.4 Sequence of Steps

Given that the AA3-UC1 environment is in the state described in the previous section, this section outlines the steps to execute the business logic of this use case. For every step we describe the required input, objective, and expected output, summarized in Table 5-55.

Table 5-55: AA3 UC1 – Set of steps necessary for the use case execution.

Step Number	***Input***	***Objective***	***Output***

Co-funded by the European Union

S1	*Operator goes to his workstation and starts the trial by pressing clicking on some Virtual or physical Start button*	*Mark the beginning of trial so that Data processing and event processing application elements can differentiate the beginning of trial and setup datapoints*	*Change of state in a Boolean variable.*
S2	*Operator /Cobot start working*	*Start of trial*	*Muscle fatigue data signals from operator begin to flow and show variations.*
S3	*Muscle fatigue signals, output stress Flag from Anomaly detection, and time stamp data*	*Monitoring of Operator stress data by application elements to generate specific instructions to Cobot. For Dynamic load redistribution between Operator and robot.*	*Output program type for the Cobot – Intensive support/ Nominal support.*
S4	*End of trial Buzzer /Timer*	*Provides instruction to operator to initiate stop the trial.*	-
S5	*Operator stops the trial by pressing clicking on some virtual or physical stop button*	*Mark the end of trial so as to distinguish between actual trial datapoints and discarded datapoints.*	*Change of state in a Boolean variable.*

5.3.3.5 Data Models

This section provides a preliminary data model for the EMG sensor to be used in AA3-UC1. This data model is visualized in Figure 5-20.

Co-funded by the European Union

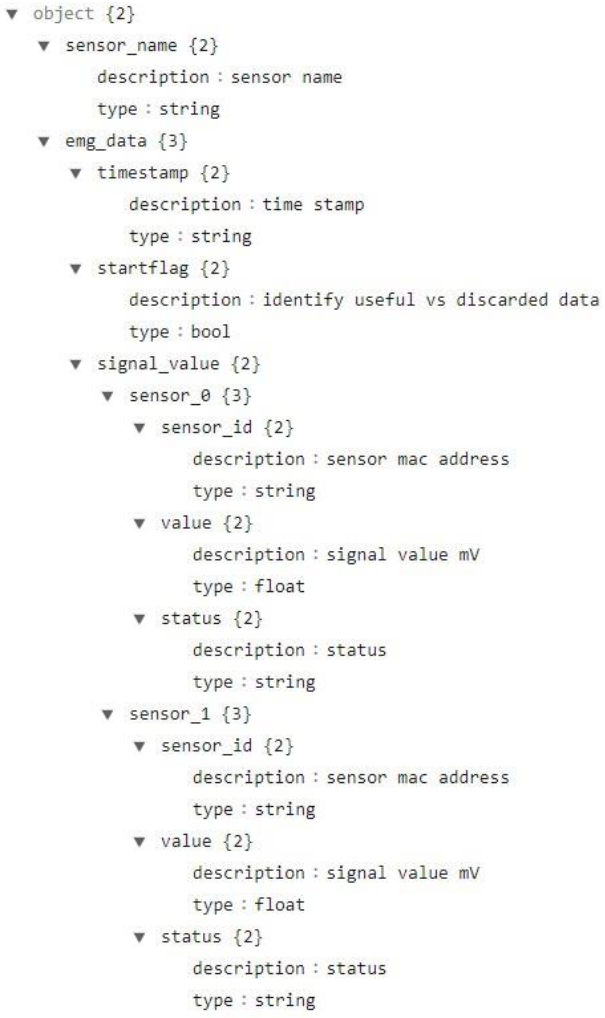

Figure 5-20: AA3 UC1 – Data model EMG sensor

5.3.3.6 Workflows

Figure 5-21 depicts the expected workflows of the trial. At the beginning of the trial, the operator starts by wearing the exoskeleton and the EMG sensors which measure physical fatigue. As the trial starts the operator first verifies that the EMG sensors connectivity to ensure that data is flowing to the application elements.

During the assembly process, the Anomaly Detection application element (AD/ APP-01-c) continuously tracks his muscular fatigue levels. If his muscular stress exceeds a threshold value continuously for a long enough period, Complex Event Processing app (CEP/ APP-01-d) which continuously receives stress state data from AD communicates with the Cobot to switch to “Intensive support state” from the “Nominal support state”. Under this Intensive support state, the Cobot takes a larger part of the work. This provides interim relief to the operator.

After sufficient time the stress levels return to normal, the Cobot reverts to its regular, “nominal support mode”. This real-time fatigue monitoring and adaptive support, ensures efficiency and operator well-being by dynamically adapting degree of support offered to the operator through the duration of the trial.

The trial comes to an end when several such cycles have been repeated and the set duration of trial time is over. In real life industrial scenarios this would potentially represent the end of shift or lunch break etc.

Co-funded by
the European Union

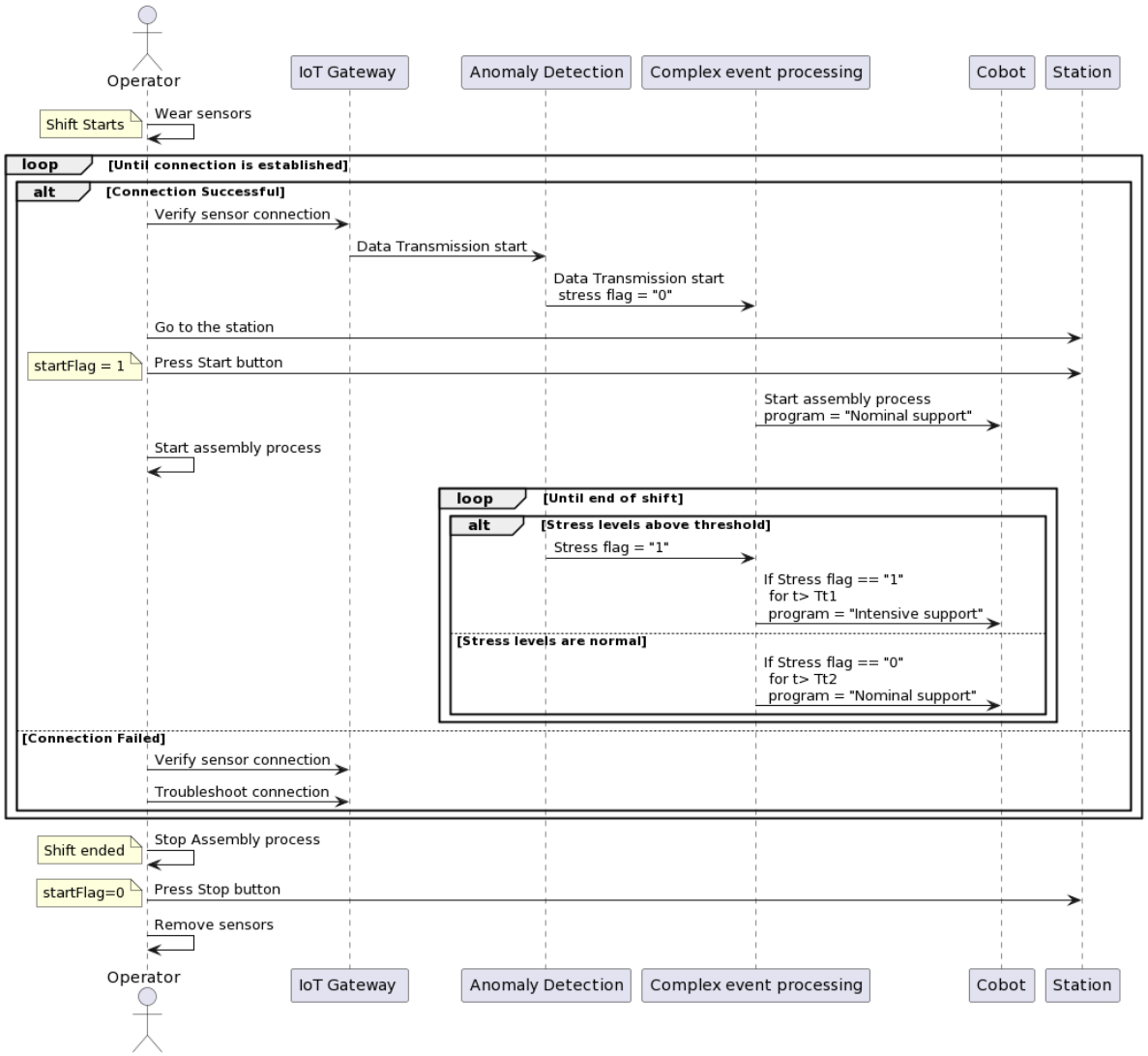

Figure 5-21: AA3 UC1 – Workflow diagram.

5.3.3.7 Requirements

Req. Id	Requirement Description	
AA3.UC1.01	Testing factory **MUST** provide private 5G network	
	Affected components	IoT Devices
	Contributing Partner	MADE
	Comment	//

Table 5-56: AA3.UC1.01 requirement

Req. Id	Requirement Description	
AA3.UC1.02	EMG sensors **MUST** be connected to the network	
	Affected components	IoT Devices
	Contributing Partner	MADE, POLIMI
	Comment	//

Table 5-57: AA3.UC1.02 requirement

Co-funded by the European Union

Req. Id	Requirement Description	
AA3.UC1.03	Raw data **COULD** be transformed and "prepared" for the sending	
	Affected components	Service Components
	Contributing Partner	POLIMI
	Comment	//

Table 5-58: AA3.UC1.03 requirement

Req. Id	Requirement Description	
AA3.UC1.04	Collected/transformed data **MUST** be communicated to the IoT platform	
	Affected components	Service Components, Storage components
	Contributing Partner	POLIMI
	Comment	//

Table 5-59: AA3.UC1.04 requirement

Req. Id	Requirement Description	
AA3.UC1.05	Operational data from the assembly process **MUST** be available to the informative system	
	Affected components	Service components, Storage components
	Contributing Partner	MADE
	Comment	//

Table 5-60: AA3.UC1.05 requirement

Req. Id	Requirement Description	
AA3.UC1.06	Infrastructure network connection **MUST** be available from MADE to IoT platform	
	Affected components	IoT Devices
	Contributing Partner	MADE
	Comment	//

Table 5-61: AA3.UC1.06 requirement

Req. Id	Requirement Description	
AA3.UC1.07	EMG sensors **MUST** be available and operative	
	Affected components	Service Components
	Contributing Partner	MADE
	Comment	//

Table 5-62: AA3.UC1.07 requirement

Req. Id	Requirement Description	
AA3.UC1.08	Exoskeleton **MUST** be available and operative	
	Affected components	*Service components
	Contributing Partner	MADE
	Comment	//* Absence of Exoskeleton could seriously skew the wearable sensor outputs causing failed trial execution.

Co-funded by the European Union

Table 5-63: AA3.UC1.08 requirement

Req. Id	Requirement Description	
AA3.UC1.09	Collaborative robot/s **MUST** be available and operative	
	Affected components	*
	Contributing Partner	MADE
	Comment	//* Essential Hardware element of the Use case execution. Trials run without the Cobots do not add value.

Table 5-64: AA3.UC1.09 requirement

Req. Id	Requirement Description	
AA3.UC1.10	Semi-industrial production factory **MUST** be available and operative	
	Affected components	*
	Contributing Partner	MADE
	Comment	Semi industrial environment is necessary because it provides a simulated environment of an industry and at the same time allows for more flexible trial environment while providing close to real industrial setting.

Table 5-65: AA3.UC1.10 requirement

5.3.3.8 Potential Future Extensions

The polar H10 sensor which is being used for use case 2 can be added as an additional data source to incorporate fall detection functionality by utilising the accelerometers on the sensor.

5.3.3.9 GDPR Issues

Both AA3 use cases collect data from wearable sensors which measure muscle activity and ECG(EKG) of the operators. These signals are further used to train ML models which help in detecting the muscular and physiological fatigue of the operators. Since this signal strength and nature depends closely on the physical and mental conditions of the operator, it may be necessary for the ML algorithms to be able to Identify/detect which operator is wearing the sensors. Which in turn may lead to the necessity of storing information about the operators which can be eventually be backtracked to a specific operator. Such information could be potentially used against the operator.

As a preliminary measure random offset to timestamps of each shift data. so that the stored data cannot be linked with a specific shift. However more robust measures also need to be explored to further secure personal information.

5.3.3.10 Risk Assessment & Mitigation Plan

This section discusses risks that may arise while developing the AA3 UC1 and propose workarounds for mitigating those risks (see Table 5-66).

Co-funded by the European Union

Table 5-66: AA3 UC1 – Foreseen risks and mitigation measures.

Risk	Description of risk	Likelihood	Severity	Proposed risk-mitigation measure(s)
AA3-UC1-R01	5G spectrum in Italy was solely auctioned to telecom operators in 2018 as such transmitting over 5G frequencies could invite Legal consequences/ Fines.	High	High	Transmission over Telecom operator owned 5G spectrums under license from the operator like Vodafone or Fastweb operators.

5.3.4 AA3-UC2: Human in Loop

In the proposed scenario, a human operator is working on an assembly station where his performance is declining due to factors like fatigue or cognitive stress, on account of physically and mentally demanding operation. In real industrial situations high cognitive stress/overload combined with physical fatigue can not only result in defective products but it can also result in potentially hazardous/life threatening situations for the operators. In this scenario the operator is equipped with muscle fatigue and stress sensors. The data from these sensors is being continuously analyzed with anomaly detection and complex event processing applications. When the stress level consistently stays above threshold levels the Production line speed is reduced until the operator performance returns to normal or a colleague from the next shift replaces the current operator. This concept targets adaptive production lines that dynamically respond to the operator's cognitive and physical fatigue levels. Thus, ensuring safer conditions and preventing errors or hazards for operators working under high stress. This approach aligns with the objectives of lean manufacturing, fostering a safer work environment while optimizing efficiency. AA3-UC2 is visualized in Figure 5-22.

.

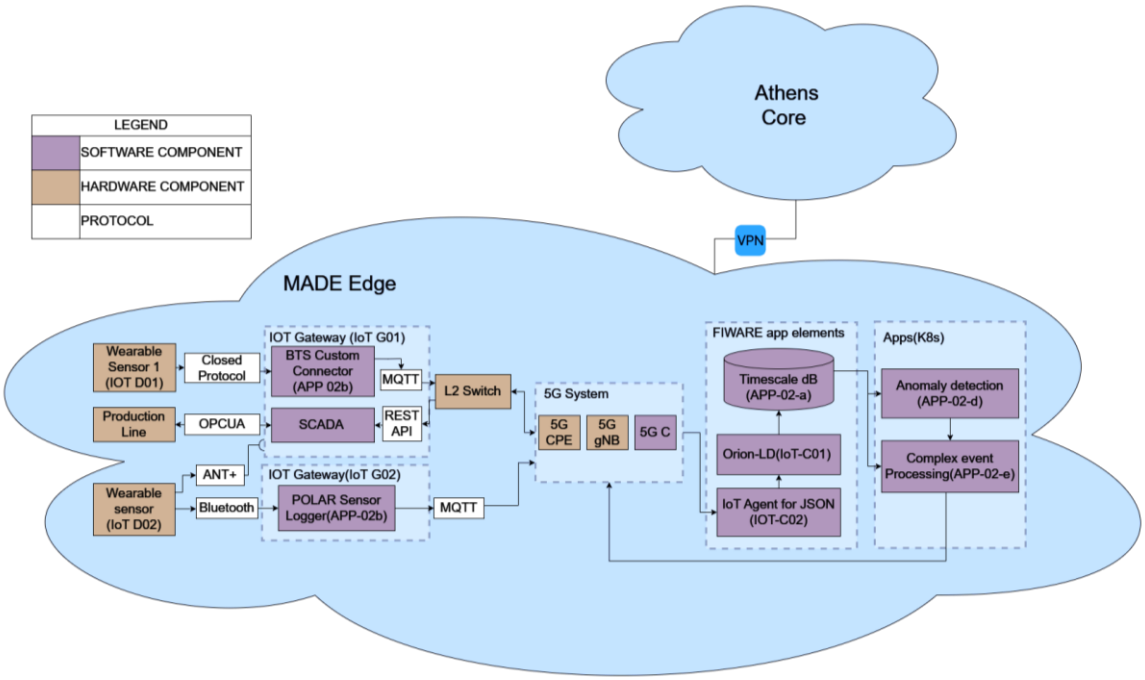

Figure 5-22: AA3 UC2 – Overview.

5.3.4.1 Objective

In addition to the general objectives of the AA3 the use case has following specific goals:

- Implement an adaptive production line which uses human biomarkers for operator load balancing to properly manage physical and mental stresses of the operators while working under the umbrella of INCODE platform.
- Promote lean manufacturing by helping to achieve zero defect assembly at the station level.

Promote a safer work environment in manufacturing settings by reducing workloads of highly stressed operators thereby preventing potentially fatal work conditions.

5.3.4.2 End-user Service Components

Table 5-67 introduces a list of (software-based) end-user service components used to drive the business logic of this use case.

Table 5-67: AA3 UC2 – End-user service components.

End-user Service Component				Provided by
ID	Name	Description	Deployment Domain	
APP-02-a	Database (Timescale-DB)	Stores data from the sensors and the Cobot state related info	Edge	FIWARE
APP-02-b	Connectors	Set of intermediate applications which further connect to IoT drivers.	Edge	POLIMI
APP-02-c	Drivers	Drivers for IoT and Robotics devices to allow communication to Orion LD - IoT-C01	Edge	FIWARE
APP-02-d	Anomaly detection	Processes data from Sensors to generate a cumulative stress state of the operator.	Edge	POLIMI
APP-02-e	Complex event Procession	Uses Processed cumulative stress state data from Anomaly detection and time stamps to vary the Assembly line speed dynamically based on defined logics.	Edge	POLIMI

*Pre-existing solution from FIWARE Catalog.

5.3.4.3 Initial Use Case State

This section outlines the initial state of the system, before executing the AA3-UC2 workflow. This state is captured by the entries in Table 5-68.

Co-funded by the European Union

Table 5-68: AA3 UC2 – Initial state of the system prior to the use case execution.

Initial State of the System prior to this UC
The sensors are fully charged to ensure no outage during the trial
The operators have been educated in advance about the work instructions. Without Initial Training to facilitate high cognitive loads.
The operators are wearing the sensors properly. Here special care needs to be taken to ensure that the BTS EMG sensor is in proper contact with the skin and the location of the placement is at a pre decided location which is suitable for accurate and repeatable measurements. Similarly for Polar H10 sensor electrode strap needs to be moist before wearing to ensure good signal capture.
All hardware components, like IOT gateways, edge processor etc. are booted up and ready to receive the data stream.
Connectivity between BTS EMG sensors (IoT -D01) and IoT Gateway (IoT-G01) and Polar H10 sensor (IoT-D02) to IoT Gateway (IoT-G02)
This UC assumes an available K8s cluster running the Application elements running on commodity server CS01
A connectivity service between the available cluster and BTS EMG sensors (IOT-D01) and Polar H10 sensor (IoT-D02) is required

5.3.4.4 Sequence of Steps

Given that the AA3-UC2 environment is in the state described in the previous section, this section outlines the steps to execute the business logic of this use case. For every step we describe the required input, objective, and expected output, summarized in Table 5-69.

Table 5-69: AA3 UC2 – Set of steps necessary for the use case execution.

Step Number	Input	Objective	Output
S1	*Operator goes to his workstation and starts the trial by pressing clicking on some Virtual or physical Start button*	*Mark the beginning of trial so that Data processing and event processing application elements can differentiate the beginning of trial and setup datapoints*	*Set of state in a Boolean variable "startFlag =1".*
S2	*Mobile cobot feeds material to station.*	*Start of trial, Operator starts working*	*Muscle fatigue data and ECG signals from operator begin to flow and show variations.*

Co-funded by the European Union

S3	*Muscle fatigue signals, ECG data signal, and time stamp data, startFlag data.*	*Monitoring of Operator Muscle fatigue and Physiological stress data by application elements to dynamically vary line speed relieve operator during high stress conditions.*	*1) HRV data from ECG signal* *2) Stress state data(bool) from combined Muscle fatigue and HRV signal* *3) Line speed instructions to MES system.*
S4	*End of trial Buzzer /Timer*	*Provides instruction to operator to initiate stop the trial.*	-
S5	*Operator stops the trial by pressing clicking on some virtual or physical stop button*	*Mark the end of trial so as to distinguish between actual trial datapoints and discarded datapoints.*	*Set of state in a Boolean variable "startFlag =0".*

5.3.4.5 Data Models

This section provides preliminary data models for the EMG sensor and Polar H10 sensor to be used in AA3-UC2. These data models are visualized in Figure 5-23 and Figure 5-24.

Co-funded by the European Union

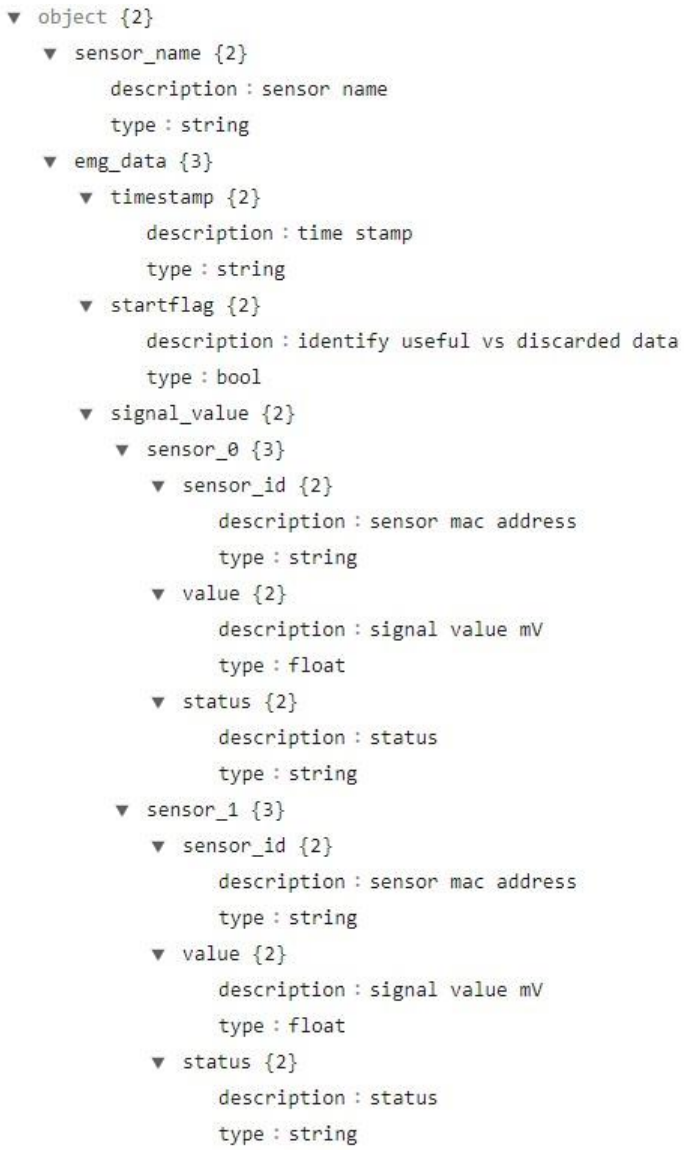

Figure 5-23: AA3 UC2 – Indicative data model for the EMG sensor to be used in this use case.

Co-funded by
the European Union

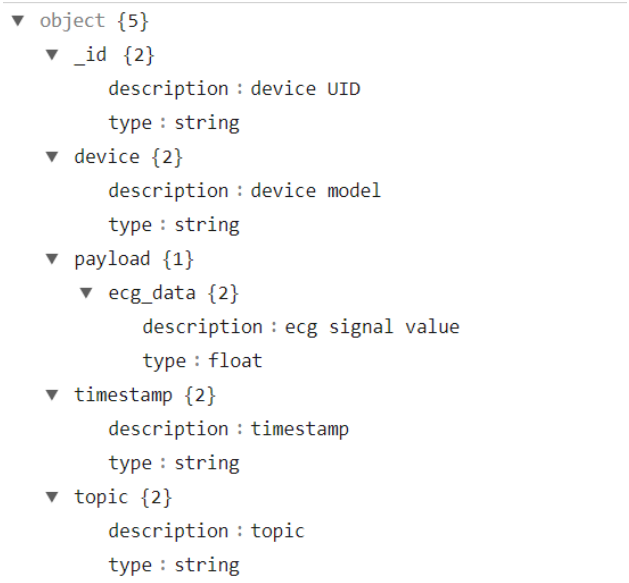

Figure 5-24: AA3 UC2 – Indicative data model for the Polar H10 sensor to be used in this use case.

5.3.4.6 Workflows

Figure 5-25 represents the workflow diagram of AA3-UC2. At the start of the trial the Operator wears the sensors that measure his stress levels. These sensors are connected to IoT Gateways, to initiate data transmission. With sensors in place and connected, the Operator heads to their workstation. The IoT Gateway transmits sensor data as the Operator begins their assembly job. Simultaneously, a Mobile Cobot feeds material to the workstation.

Throughout the trial, the Anomaly Detection app element (AD) continuously monitors the Operator's stress levels and forwards to the Complex Event Processing application (CEP). If stress levels exceed a threshold value for more than a certain fixed duration of time, the CEP communicates with the Manufacturing Execution System to adjust the line speed, offering relief to the operator when the operator is in high stress state for too long time.

When the Operator's stress levels return to normal and remain low for a duration more than a threshold value (Tt2), the system restores the line speed to normal. This process of real-time stress monitoring and adjustment ensures both efficiency and Operator well-being.

Co-funded by
the European Union

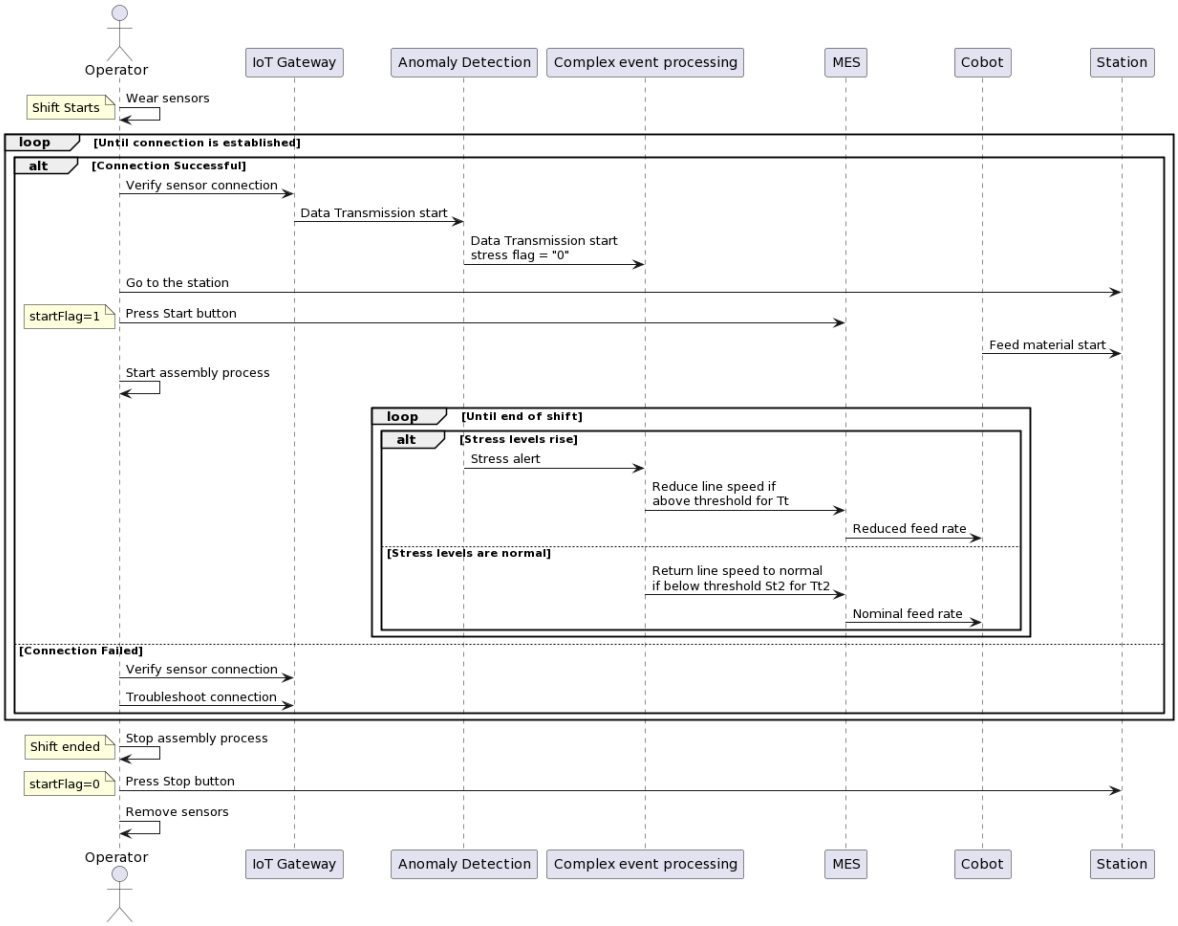

Figure 5-25: AA3 UC2 – Workflow diagram.

5.3.4.7 Requirements

Req. Id	Requirement Description	
AA3.UC2.01	Testing factory **MUST** provide private 5G network	
	Affected components	IoT Devices
	Contributing Partner	MADE
	Comment	//

Table 5-70: AA3.UC2.01 requirement

Req. Id	Requirement Description	
AA3.UC2.02	EMG sensors **MUST** be connected to the network	
	Affected components	Service Components
	Contributing Partner	MADE/POLIMI
	Comment	//

Table 5-71: AA3.UC2.02 requirement

Req. Id	Requirement Description	
AA3.UC2.03	Wearable HR Sensor **MUST** be connected to the network	
	Affected components	Service Components

Co-funded by
the European Union

	Contributing Partner	MADE/POLIMI
	Comment	//

Table 5-72: AA3.UC2.03 requirement

Req. Id	Requirement Description	
AA3.UC2.04	Raw data **COULD** be transformed and "prepared" for the sending	
	Affected components	Service components, Storage Components
	Contributing Partner	POLIMI
	Comment	//

Table 5-73: AA3.UC2.04 requirement

Req. Id	Requirement Description	
AA3.UC2.05	Collected/transformed data **MUST** be communicated to the Storage Elements	
	Affected components	Service Components, Assembly line
	Contributing Partner	POLIMI
	Comment	//

Table 5-74: AA3.UC2.05 requirement

Req. Id	Requirement Description	
AA3.UC2.06	Operational data from the assembly process **COULD** be available to the informative system	
	Affected components	Service components, Storage Components
	Contributing Partner	POLIMI, MADE
	Comment	//

Table 5-75: AA3.UC2.06 requirement

Req. Id	Requirement Description	
AA3.UC2.07	Infrastructure network connection **MUST** be available from MADE to IoT platform	
	Affected components	IoT Devices
	Contributing Partner	MADE
	Comment	//

Table 5-76: AA3.UC2.07 requirement

Req. Id	Requirement Description	
AA3.UC2.08	EMG/stress sensors **MUST** be available and operative	
	Affected components	Service Components
	Contributing Partner	MADE
	Comment	//

Table 5-77: AA3.UC2.08 requirement

Req. Id	Requirement Description
AA3.UC2.09	Semi-industrial production factory **MUST** be available and operative

Co-funded by the European Union

	Affected components	*
	Contributing Partner	MADE
	Comment	//* Semi industrial environment is necessary because it provides a simulated environment of an industry and at the same time allows for more flexible trial environment while providing close to real industrial setting.

Table 5-78: AA3.UC2.09 requirement

5.3.4.8 Potential Future Extensions

The polar H10 sensor which is being used for use case 2 has two types of sensors, namely – Single Channel ECG and a 3-axis accelerometer. While the ECG data is currently being exploited for physiological stress determination, in future we can also exploit the accelerometer data for developing a fall detection algorithm. Secondly, it is possible to eliminate Mobile phone based IoT gateway by developing a PC based Applications and BLE or ANT+ USB dongles.

5.3.4.9 GDPR Issues

Both AA3 use cases collect data from wearable sensors which measure muscle activity and ECG(EKG) of the operators. These signals are further used to train ML models which help in detecting the muscular and physiological fatigue of the operators. Since this signal strength and nature depends closely on the physical and mental conditions of the operator, it may be necessary for the ML algorithms to be able to Identify/detect which operator is wearing the sensors. Which in turn may lead to the necessity of storing information about the operators which can be eventually be backtracked to a specific operator. Such information could be potentially used against the operator.

As a preliminary measure random offset to timestamps of each shift data. so that the stored data cannot be linked with a specific shift. However more robust measures also need to be explored to further secure personal information.

5.3.4.10 Risk Assessment & Mitigation Plan

This section discusses risks that may arise while developing the AA3 UC2 and propose workarounds for mitigating those risks (see Table 5-79).

Table 5-79: AA3 UC2 – Foreseen risks and mitigation measures.

Risk	Description of risk	Likelihood	Severity	Proposed risk-mitigation measure(s)
AA3-UC2-R01	5G spectrum in Italy was solely auctioned to telecom operators in 2018 as such transmitting over 5G frequencies could invite Legal consequences/ Fines.	High	High	Transmission over Telecom operator owned 5G spectrums under license from the operator like Vodafone or Fastweb operators.

Co-funded by the European Union

5.4 AA4: Communities and PPDR

The Public Protection and Disaster Relief (PPDR) application area focuses on utilizing the collaboration between UAVs in two distinct use cases: UAV to UGV (Ground Robot) and UAV to UAV (Swarm Drones) for the purpose of conducting search and rescue operations in disaster and emergency situations. The coordination of these operations is crucial for public safety and effective disaster relief efforts.

The 5G edge computing infrastructure plays a pivotal role in facilitating seamless communication and data transfer between the UAVs and UGVs including synchronization and decision-making. This infrastructure will allow swarms and UGV to dynamically adapt search patterns and responds to the detection of missing people. Table 5-80 summarizes the proposed use cases in the INCODE Application Area 4.

Table 5-80: AA4 – Summary of use cases.

Use Case Acronym	Use Case Name
AA4-UC1	Collaborative UAV and UGV
AA4-UC2	Drone Swarm Search

5.4.1 Objective

1. To improve emergency response in public protection and disaster relief scenarios. This includes facilitating timely and efficient search and rescue operations.
2. To enable seamless coordination and collaboration among various actors, including UAVs, UGVs, swarm drones and to foster effective communication, information sharing between devices and teams.
3. To enable real-time data collection, analysis, and dissemination to provide accurate situational awareness to decision-makers and emergency responders.
4. To develop and apply advanced AI techniques for data processing and analysis, such as image recognition, object detection, and pattern recognition, to enhance the capabilities of UAVs.

5.4.2 Why is it relevant for INCODE?

AA4 represents an emergency situation where several unmanned vehicles are managed to help in search and rescue missions. As these missions require a fast deployment and instantiation of services over the 5G network and specialized hardware, such as GPUs. AA4 in its final demo will demonstrate the capabilities the INCODE platform in terms of rapid deployment, scalability, easiness of development, uniform deployment, and optimisation of resource allocation among others.

5.4.3 AA4-UC1: Collaborative UAV and Ground Robot

This use case involves the collaboration between an Unmanned Aerial Vehicle (UAV) and a Ground Robot (UGV) with the shape of a dog to conduct efficient search and rescue

Co-funded by the European Union

operations. The UAV is responsible for searching the designated area and the UGV to rescue the victims.

The UAV streams live video footage to the network application, which are equipped with advanced artificial intelligence (AI) techniques for the detection of missing individuals in challenging environments. The video undergoes processing to detect and locate missing persons. Upon successful detection, the UAV shares the precise GPS coordinates of the missing person with the UGV.

The UGV plays a vital role in the rescue operation by carrying a first aid kit to provide immediate medical assistance. Additionally, the UGV may assist in guiding the rescue team to the exact location of the missing individual, ensuring a swift and effective rescue mission.

UAV(s) and ground robot(s) work together to enable more real-time, efficient, and effective search and rescue operations, with the support of the INCODE infrastructure and platform. The seamless collaboration between the UAV and UGV, facilitated by the INCODE platform, enhances situational awareness, optimizes resource allocation, and accelerates the overall search and rescue process. Figure 5-26 presents the AA4-UC 1 architecture where two IoT devices are connected to the 5G network and where at least 7 microservices are deployed in the INCODE architecture.

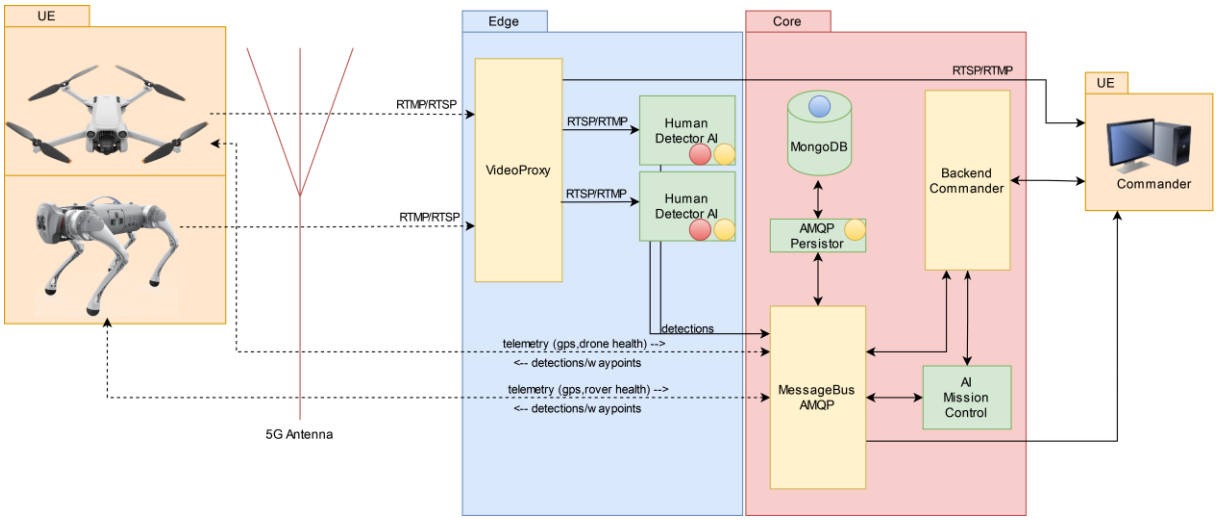

Figure 5-26: AA4 UC1 – Overview.

5.4.3.1 Objective

1. To establish a collaborative framework between a UAV and a UGV for efficient search and rescue operations.
2. Real-Time video streaming from the UAV to the human detection AI microservice.
3. High accuracy human detection in from a UAV.
4. Calculation of the location of the missing person and shared with the UGV in a form of GPS coordinates.
5. Remote piloting of the UGV to guide the rescue team to the precise location of the missing individual.
6. Enable real-time collaboration and information exchange between the UAV and UGV through the INCODE infrastructure and platform.

Co-funded by the European Union

5.4.3.2 End-user Service Components

Table 5-81 introduces a list of (software-based) end-user service components used to drive the business logic of this use case.

Table 5-81: AA4 UC1 – End-user service components.

End-User Service Component				Provided by
ID	**Name**	**Description**	**Deployment Domain**	
APP-01	VideoProxy	Nginx-based server. Streams video.	Edge cloud	UWS
APP-02	MongoDB	Server to store information.	Core cloud	UWS
APP-03	MessageBus	RabbitMQ-based server to exchange information: detections, GPS coordinates, UAV/UGV telemetry.	Core cloud	UWS
APP-04	HumanDetector	Service to detect missing people from video stream. GPU compliant.	Edge cloud	UWS
APP-05	Backend Commander	Backend for commander to manage the operation.	Core cloud	UWS
APP-06	AI Mission Planner	Main task is to calculate the location coordinates (GPS) of the missing person.	Core cloud	UWS
APP-07	AMQP Persistor	Communicates APP-03 with APP-02.	Core cloud	UWS

5.4.3.3 Initial Use Case State

This section outlines the initial state of the system, before executing the AA4-UC1 workflow. This state is captured by the entries in Table 5-82.

Table 5-82: AA4 UC1 – Initial state of the system prior to the use case execution.

Initial State of the System prior to this UC
The operation centre is equipped with UAV and ground robot, which can be deployed on demand.
The UAV and ground robot are equipped with the necessary sensors and cameras, e.g., a UAV may use a thermal camera to locate the victim in night operations, and communication interfaces.
The UAV and ground robot can communicate with each other's operator in real time, via a 4G/5G network.
Search & Rescue team including Ground personnel e.g., pilots of the UAVs/robots, are involved for operation, regulation, and other considerations.
A 4G/5G network is existing in the search area.

Co-funded by the European Union

Edge nodes are deployed to host the required AI models and their operating environment with the required resources, either pre-installed or installed on demand with the help of the INCODE platform.

5.4.3.4 Sequence of Steps

Given that the AA4-UC1 environment is in the state described in the previous section, this section outlines the steps to execute the business logic of this use case. For every step we describe the required input, objective, and expected output, summarized in Table 5-83.

Table 5-83: AA4 UC1 – Set of steps necessary for the use case execution.

Step Number	Input	Objective	Output
S1	UAV app starts.	Connect to antenna.	Successful Connection
S2	UGV starts.	Connect to antenna.	Successful Connection.
S3	Orchestrator deploys VNFs.	VNF running.	VNF successfully running.
S4	UAV sends telemetry and video to VNFs.	VNFs receives data from UAV.	VNFs process data from UAV.
S5	Commander connects to BackendCommander.	Commander watches video and information from UAV.	Commander watches video.
S6	HumanDetectorAI processes video.	HumanDetectorAI provides results with people detected.	Results sent to MessageBus.
S7	MissionPlanner receives information from detections.	MissionPlanner provides GPS coordinates to UGV.	UGV moves to designated coordinates.

5.4.3.5 Data Models

This section provides preliminary data models that describe the communication between various software and hardware components in AA4-UC1 (see Figure 5-27, Figure 5-28, and Figure 5-29).

Co-funded by the European Union

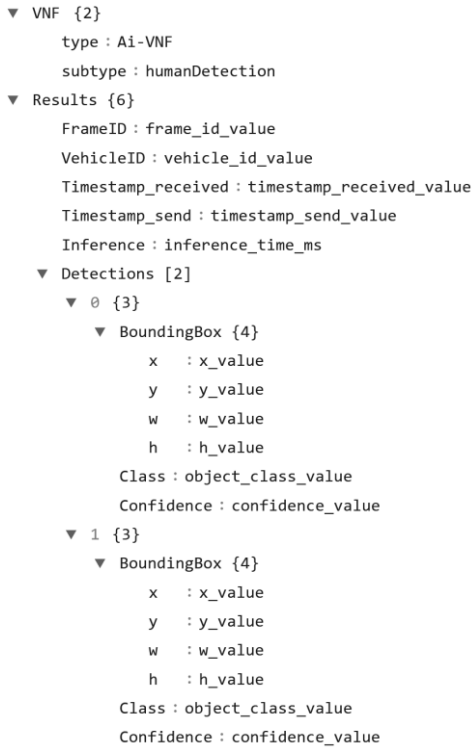

Figure 5-27: AA4 UC1 – Data model for Human Detector (APP-04) to MessageBus (APP-03) communication.

Co-funded by
the European Union

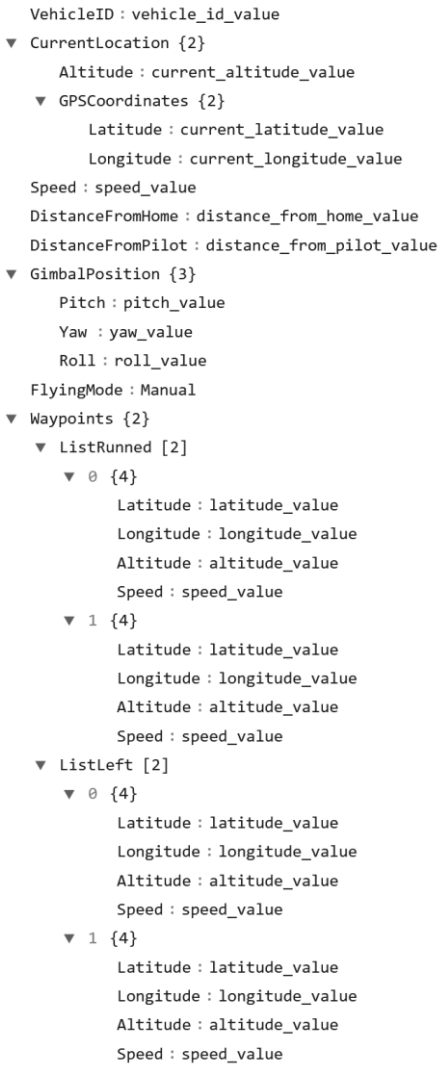

Figure 5-28: AA4 UC1 – Data model for UAV to AIMissionPlanner (APP-06) communication.

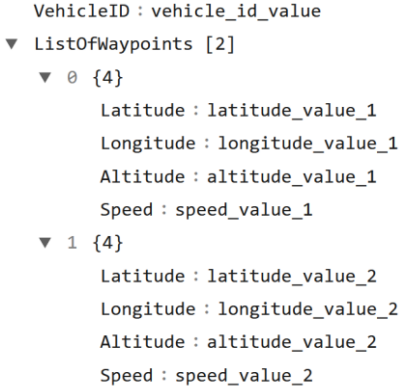

Figure 5-29: AA4 UC1 – Data model for AIMissionPlaner to UAV or UGV communication.

5.4.3.6 Workflows

Figure 5-30 visualizes the steps to execute the workflow for AA4-UC1.

Co-funded by
the European Union

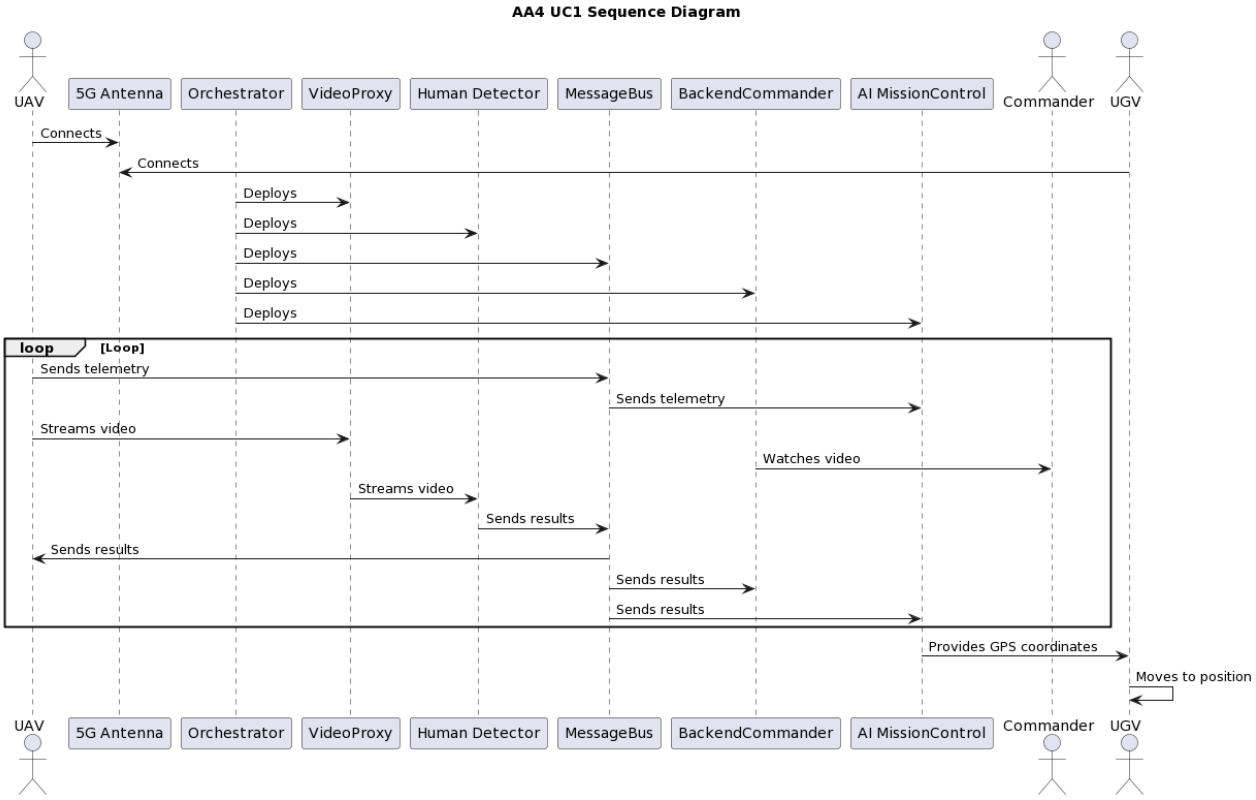

Figure 5-30: AA4 UC1 – Workflow diagram.

5.4.3.7 Requirements

This section reports use-case specific requirements. These requirements must be met before the execution of the use case.

Req. Id	Requirement Description	
AA4.UC1.01	The SAR area **MUST** meet the legal condition to fly a UAV.	
	Affected components	UAVs
	Contributing Partner	UoP (responsible partner for the SAR testing area)
	Comment	//

Table 5-84: AA4.UC1.01 requirement

Req. Id	Requirement Description	
AA4.UC1.02	The operation centre **MUST** be equipped with at least one UAV and one UGV.	
	Affected components	UAVs
	Contributing Partner	UWS
	Comment	//

Table 5-85: AA4.UC1.02 requirement

Req. Id	Requirement Description
AA4.UC1.03	UAV and UGV **MUST** be equipped with cameras.

Co-funded by the European Union

	Affected components	UAV and UGV
	Contributing Partner	UWS and UoP
	Comment	//

Table 5-86: AA4.UC1.03 requirement

Req. Id	Requirement Description	
AA4.UC1.04	UAV and UGV **MUST** have 4G/5G connectivity.	
	Affected components	UAV and UGV
	Contributing Partner	UWS and UoP
	Comment	The RAN of the UoP testbed MUST allow a WiFi-4G/5G mobile router approach for the UAV/UGV to be connected to 4G/5G, if no onboard 4G/5G interface for UAV/UGV is available.

Table 5-87: AA4.UC1.04 requirement

Req. Id	Requirement Description	
AA4.UC1.05	UAV **MUST** cover areas autonomously (waypoint capability).	
	Affected components	UAVs
	Contributing Partner	UWS
	Comment	//

Table 5-88: AA4.UC1.05 requirement

Req. Id	Requirement Description	
AA4.UC1.06	UGV **MUST** reach the identified location instructed by the personnel.	
	Affected components	UGV
	Contributing Partner	UoP
	Comment	//

Table 5-89: AA4.UC1.06 requirement

Req. Id	Requirement Description	
AA4.UC1.07	The edge node **MUST** provide at least one Nvidia GPU.	
	Affected components	Edge node at UoP testbed
	Contributing Partner	UoP
	Comment	//

Table 5-90: AA4.UC1.07 requirement

Req. Id	Requirement Description	
AA4.UC1.08	Network App **COULD** be deployed on demand.	
	Affected components	Orchestrator
	Contributing Partner	UBI and T4.2 partners
	Comment	//

Table 5-91: AA4.UC1.08 requirement

Co-funded by the European Union

Req. Id	Requirement Description	
AA4.UC1.09	AI model for human detection **MUST** be available for deployment.	
	Affected components	UWS AI model
	Contributing Partner	UWS
	Comment	//

Table 5-92: AA4.UC1.09 requirement

Req. Id	Requirement Description	
AA4.UC1.10	Mobile APP for UAV control **MUST** be available for controlling the UAV.	
	Affected components	UWS Mobile APP
	Contributing Partner	UWS
	Comment	//

Table 5-93: AA4.UC1.10 requirement

Req. Id	Requirement Description	
AA4.UC1.11	Network App framework **MUST** be available for deployment.	
	Affected components	UWS Network App framework
	Contributing Partner	UWS
	Comment	//

Table 5-94: AA4.UC1.11 requirement

Req. Id	Requirement Description	
AA4.UC1.12	Commander GUI **COULD** be available for observing and managing the operation of UAVs	
	Affected components	UWS Commander GUI
	Contributing Partner	UWS
	Comment	//

Table 5-95: AA4.UC1.12 requirement

Req. Id	Requirement Description	
AA4.UC1.13	Observer GUI **COULD** be available for observing the detection results.	
	Affected components	UWS Observer GUI
	Contributing Partner	UWS
	Comment	//

Table 5-96: AA4.UC1.13 requirement

Req. Id	Requirement Description	
AA4.UC1.14	UAV **MUST** be able to send location of the detected human (GPS coordinates) to the ground robot or its operator.	
	Affected components	UAVs
	Contributing Partner	UWS
	Comment	//

Table 5-97: AA4.UC1.14 requirement

Co-funded by the European Union

Req. Id	Requirement Description	
AA4.UC1.15	UAV and UGV **MUST** support the same communication protocol (e.g., MAVlink)	
	Affected components	UAV and UGV
	Contributing Partner	UWS and UoP
	Comment	//

Table 5-98: AA4.UC1.15 requirement

5.4.3.8 Potential Future Extensions

Not applicable to this use case.

5.4.3.9 GDPR Issues

In the context of police forces and emergency services searching for missing persons (stakeholders of this AA4), compliance with the General Data Protection Regulation (GDPR) is crucial. The core principles include a legitimate basis for data processing, such as the public interest or official authority, ensuring necessity and proportionality in data collection, practicing data minimization by only gathering pertinent information, and handling sensitive data cautiously. Additionally, compliance with data retention requirements and addressing individuals' rights under GDPR is imperative.

This use case does not store sensitive information such as person identification. There is not personal data in this use case, therefore a minimisation of it is satisfied. For trials during the scope of this European Project, all the members which information may be shared will sign a privacy policy form. No personal data is stored over the context of this application area.

5.4.3.10 Risk Assessment & Mitigation Plan

This section discusses risks that may arise while developing the AA4 UC1 and propose workarounds for mitigating those risks (see Table 5-99).

Table 5-99: AA4 UC1 – Foreseen risks and mitigation measures.

Risk	Description of risk	Likelihood	Severity	Proposed risk-mitigation measure(s)
AA4-UC1-R01	Air controller restriction to fly UAVs in the area.	Medium	High	Change scenario to other area.
AA4-UC1-RO2	Weather condition not acceptable for UAV and UGV.	Medium	High	Change testing date.
AA4-UC1-RO3	API incompatibility.	Low	High	Different UAV/UGV brand should be used.

Co-funded by the European Union

AA4-UC1-RO4	UGV cannot receive coordinates.	Medium	Low	UGV pilot moves the UGV to an estimated location.
AA4-UC1-RO5	High latency video and results transmission.	Medium	Medium	UGV and UAV moves slower.
AA4-UC1-RO6	scalability issues when they deploy HumanDetectorAI VNFs	Medium	Medium	To optimise the VNF 's in terms of memory consumption or to deploy more GPU hardware.

5.4.4 AA4-UC2: Drone Swarm Search

This use case involves the collaboration between two or more Unmanned Aerial Vehicles (UAV) efficient searching of missing people in wild environments. The UAVs are responsible for searching the designated area in different search patterns and altitudes. The UAVs stream live video footage to the network application, which are equipped with advanced artificial intelligence (AI) techniques for the detection of missing individuals in challenging environments. The video undergoes processing to detect and locate potential missing persons (named as hotspots). The UAV shares the precise GPS coordinates of the hotspots with the operation commander and flyer operator, leading to a more precise search (lower altitude) of a secondary UAV.

The UAVs work together to enable more real-time, efficient, and effective search and rescue operations, with the support of the INCODE infrastructure and platform. The seamless collaboration between all UAVs, facilitated by the INCODE platform, enhances situational awareness, optimizes resource allocation, and accelerates the overall search. Figure 5-31 presents the AA4-UC2 architecture where two IoT devices are connected to the 5G network and where at least 7 microservices are deployed in the INCODE architecture.

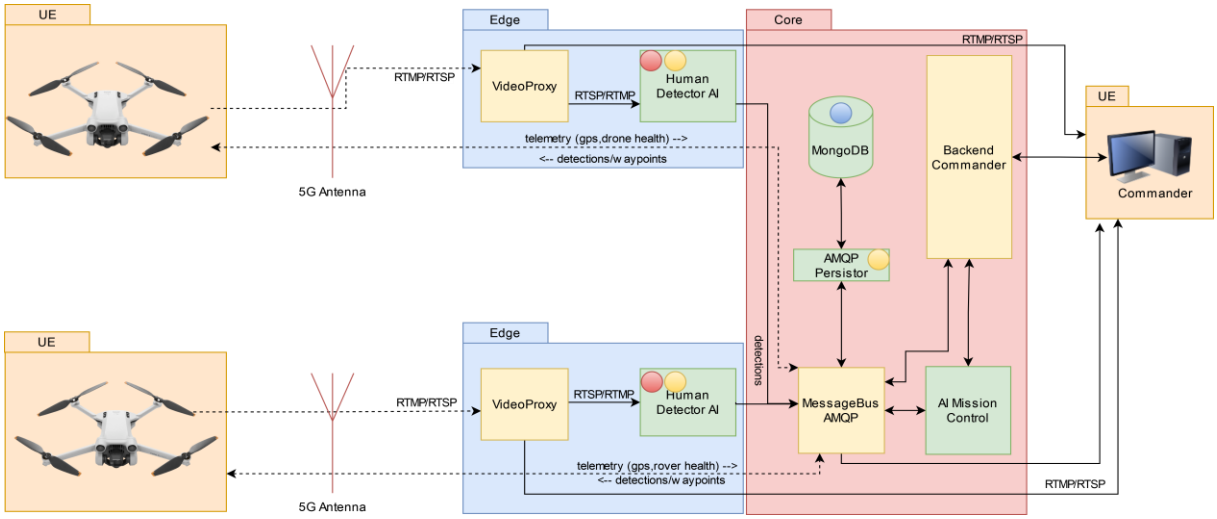

Figure 5-31: AA4 UC2 – Overview.

5.4.4.1 Objective

The objectives for this UC are enumerated as follows:

- To establish a collaborative framework between a multiple UAVs to follow efficient search patterns.
- Autonomous flight carried out by UAVs with almost not human interaction.
- Real-Time video streaming from the UAV to the human detection AI microservice.
- High accuracy human detection in from a UAV.
- Calculation of the location of the missing person and shared with a secondary UAV in a form of GPS coordinates.
- Enable real-time collaboration and information exchange between the UAV and UAV through the INCODE infrastructure and platform.

5.4.4.2 End-user Service Components

Table 5-100 introduces a list of (software-based) end-user service components used to drive the business logic of this use case.

Table 5-100: AA4 UC2 – End-user service components.

End-User Service Component				Provided by
ID	Name	Description	Deployment Domain	
APP-01	VideoProxy	Nginx-based server. Streams video.	Distributed	UWS
APP-02	MongoDB	Server to store information.	Centralised	UWS
APP-03	MessageBus	RabbitMQ-based server to exchange information: detections, GPS coordinates, UAV telemetry.	Centralised	UWS
APP-04	HumanDetector	Service to detect missing people from video stream. GPU compliant.	Distributed	UWS
APP-05	Backend Commander	Backend for commander to manage the operation.	Centralised	UWS
APP-06	AI Mission Planner	Main task is to calculate the location coordinates (GPS) of the missing person and optimise organise the search pattern.	Centralised	UWS
APP-07	AMQP Persistor	Communicates APP-03 with APP-02.	Centralised	UWS

5.4.4.3 Initial Use Case State

This section outlines the initial state of the system, before executing the AA4-UC1 workflow. This state is captured by the entries in Table 5-101.

Co-funded by the European Union

Table 5-101: AA4 UC2 – Initial state of the system prior to the use case execution.

Initial State of the System prior to this UC
The operation center is equipped with a number of UAVs, which can be deployed on demand.
The UAVs are equipped with the necessary sensors and cameras, e.g., a UAV may use a thermal camera to locate the victim in night operations, and communication interfaces.
The UAVs can communicate with each other in real time, via a 4G/5G network.
Search & Rescue team including Ground personnel e.g., pilots of the UAVs, are involved for operation, regulation, and other considerations.
A 4G/5G network is existing in the search area.
Edge nodes are deployed to host the required AI models and their operating environment with the required resources, either pre-installed or installed on demand with the help of the INCODE platform.

5.4.4.4 Sequence of Steps

Given that the AA4-UC2 environment is in the state described in the previous section, this section outlines the steps to execute the business logic of this use case. For every step we describe the required input, objective, and expected output, summarized in Table 5-102.

Table 5-102: AA4 UC2 – Set of steps necessary for the use case execution.

Step Number	Input	Objective	Output
S1	*Connections between UAV1 and Antenna1, and UAV2 and Antenna2 are established.*	*Establish connections between UAVs and antennas.*	
S2	*Orchestrator deploys various components.*	*Deploy system components.*	*Component successfully running.*
S3	*BackendCommander starts watching video feeds.*	*Begin video feed monitoring.*	
S4	*UAV1 and UAV2 send telemetry data to MessageBus.*	*Transmit telemetry data.*	*AIMissionControl receives telemetry.*
S5	*UAVs streams video*	*Video is received by components.*	

Co-funded by the European Union

S6	*Human detectors deliver results.*	*People are detected.*	*Results are received by UAV, commander and AIMissionPlanner*
S7	*AIMissionControl sends GPS coordinates*	*GPS coordinates are received.*	*UAV moves to GPS location.*

5.4.4.5 Data Models

AA4-UC2 assumes the same data models as those depicted in Section 5.4.3.5.

5.4.4.6 Workflows

Figure 5-32 visualizes the steps to execute the workflow for AA4-UC2.

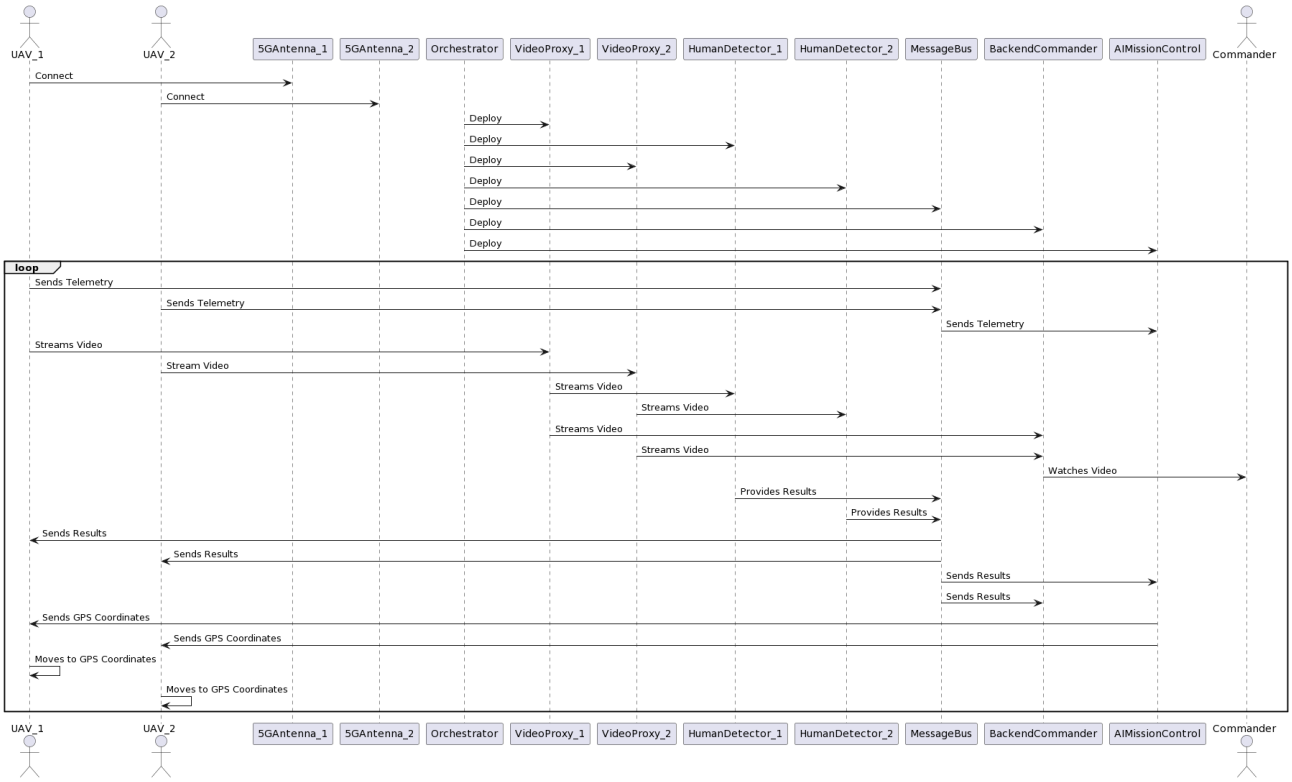

Figure 5-32: AA4 UC2 – Workflow diagram.

5.4.4.7 Requirements

This section reports use-case specific requirements. These requirements must be met before the execution of the use case.

Req. Id	Requirement Description	
AA4.UC2.01	The SAR area **MUST** meet the legal condition to fly a UAV.	
	Affected components	UAVs
	Contributing Partner	UoP (responsible partner for the SAR testing area)

Co-funded by the European Union

	Comment	//

Table 5-103: AA4.UC2.01 requirement

Req. Id	Requirement Description	
AA4.UC2.02	The operation centre **MUST** be equipped with at least two UAVs.	
	Affected components	UAVs
	Contributing Partner	UWS
	Comment	//

Table 5-104: AA4.UC2.02 requirement

Req. Id	Requirement Description	
AA4.UC2.03	UAVs **MUST** be equipped with cameras.	
	Affected components	UAVs
	Contributing Partner	UWS
	Comment	//

Table 5-105: AA4.UC2.03 requirement

Req. Id	Requirement Description	
AA4.UC2.04	UAVs **COULD** be equipped with thermal camera.	
	Affected components	UAVs
	Contributing Partner	UWS
	Comment	//

Table 5-106: AA4.UC2.04 requirement

Req. Id	Requirement Description	
AA4.UC2.05	UAVs **MUST** have 4G/5G connectivity.	
	Affected components	RAN of the testbed
	Contributing Partner	UoP
	Comment	The RAN of the UoP testbed MUST allow a WiFi-4G/5G mobile router approach for the UAV to be connected to 4G/5G, if no onboard 4G/5G interface for UAV is available.

Table 5-107: AA4.UC2.05 requirement

Req. Id	Requirement Description	
AA4.UC2.06	UAVs **MUST** cover areas autonomously (waypoint capability).	
	Affected components	UAVs
	Contributing Partner	UWS
	Comment	//

Table 5-108: AA4.UC2.06 requirement

Req. Id	Requirement Description
AA4.UC2.07	The edge node **MUST** provide at least one Nvidia GPU.

Co-funded by the European Union

	Affected components	Edge node at UoP testbed
	Contributing Partner	UoP
	Comment	//

Table 5-109: AA4.UC2.07 requirement

Req. Id	Requirement Description	
AA4.UC2.08	Network App **COULD** be deployed on demand.	
	Affected components	Orchestrator
	Contributing Partner	UBI and T4.2 partners
	Comment	//

Table 5-110: AA4.UC2.08 requirement

Req. Id	Requirement Description	
AA4.UC2.09	2 UAVs **MUST** be deployed in the Drone Swarm.	
	Affected components	UAVs
	Contributing Partner	UWS
	Comment	//

Table 5-111: AA4.UC2.09 requirement

Req. Id	Requirement Description	
AA4.UC2.10	AI model for human detection **MUST** be available for deployment	
	Affected components	UWS AI model
	Contributing Partner	UWS
	Comment	//

Table 5-112: AA4.UC2 10 requirement

Req. Id	Requirement Description	
AA4.UC2.11	Mobile APP for UAV control **MUST** be available for controlling the UAV	
	Affected components	UWS Mobile APP
	Contributing Partner	UWS
	Comment	//

Table 5-113: AA4.UC2.11 requirement

Req. Id	Requirement Description	
AA4.UC2.12	Network App framework **MUST** be available for deployment	
	Affected components	UWS Network App framework
	Contributing Partner	UWS
	Comment	//

Table 5-114: AA4.UC2.12 requirement

Req. Id	Requirement Description
AA4.UC2.13	Commander GUI **COULD** be available for observing and managing the operation of UAVs

Co-funded by the European Union

	Affected components	UWS Commander GUI
	Contributing Partner	UWS
	Comment	//

Table 5-115: AA4.UC2.13 requirement

Req. Id	Requirement Description	
AA4.UC2.14	Observer GUI **COULD** be available for observing the detection results	
	Affected components	UWS Observer GUI
	Contributing Partner	UWS
	Comment	//

Table 5-116: AA4.UC2.14 requirement

Req. Id	Requirement Description	
AA4.UC2.15	UAV **MUST** be able to send location of the detected human (GPS coordinates) to the ground robot or its operator	
	Affected components	UAVs
	Contributing Partner	UWS
	Comment	//

Table 5-117: AA4.UC2.15 requirement

Req. Id	Requirement Description	
AA4.UC2.16	UAVs **MUST** be connected to the same ground control station (GCS)	
	Affected components	UAVs and GCS
	Contributing Partner	UWS
	Comment	//

Table 5-118: AA4.UC2.16 requirement

Req. Id	Requirement Description	
AA4.UC2.17	Waypoints **COULD** be uploaded concurrently to all networked UAVs	
	Affected components	UWS Network App framework
	Contributing Partner	UWS
	Comment	//

Table 5-119: AA4.UC2 17 requirement

Req. Id	Requirement Description	
AA4.UC2.18	UAVs **MUST** support the same communication protocol (e.g., MAVlink)	
	Affected components	UAVs
	Contributing Partner	UWS
	Comment	//

Table 5-120: AA4.UC2.18 requirement

Co-funded by the European Union

5.4.4.8 Potential Future Extensions

Not applicable to this use case.

5.4.4.9 GDPR Issues

In the context of police forces and emergency services searching for missing persons (stakeholders of this AA4), compliance with the General Data Protection Regulation (GDPR) is crucial. The core principles include a legitimate basis for data processing, such as the public interest or official authority, ensuring necessity and proportionality in data collection, practicing data minimization by only gathering pertinent information, and handling sensitive data cautiously. Additionally, compliance with data retention requirements and addressing individuals' rights under GDPR is imperative.

This use case does not store sensitive information such as person identification. There is not personal data in this use case, therefore a minimisation of it is satisfied. For trials during the scope of this European Project, all the members which information may be shared will sign a privacy policy form. No personal data is stored over the context of this application area.

5.4.4.10 Risk Assessment & Mitigation Plan

This section discusses risks that may arise while developing the AA4 UC2 and propose workarounds for mitigating those risks (see Table 5-121).

Table 5-121: AA4 UC2 – Foreseen risks and mitigation measures.

Risk	Description of risk	Likelihood	Severity	Proposed risk-mitigation measure(s)
AA4-UC1-R01	Air controller restriction to fly UAVs in the area.	Medium	High	Change scenario to other area.
AA4-UC1-RO2	Weather condition not acceptable for UAVs.	Medium	High	Change testing date.
AA4-UC1-RO3	API incompatibility.	Low	High	Different UAV brand should be used.
AA4-UC1-RO4	UAV cannot receive coordinates.	Medium	Low	UAV pilot moves the UGV to an estimated location.
AA4-UC1-RO5	High latency video and results transmission.	Medium	Medium	UGV and UAV moves slower.

Co-funded by the European Union

6 Conclusions

The document provides a thorough exploration of the INCODE project from architectural and requirements perspectives. It delves into architectural foundations, examines selected logical blocks, clarifies corresponding layers, and defines application areas.

From a technical viewpoint, the INCODE project emerges as a complex and visionary endeavour, aiming to seamlessly connect the cloud-edge continuum, encompassing the radio access network to service orchestration on the edge. The well-defined architecture and logical components play pivotal roles in achieving project objectives, facilitating a clear understanding of distinct layers and logical components, thus comprehending the overall INCODE architecture.

The adoption of standardized interfaces significantly expands the INCODE ecosystem by enabling seamless integration with third parties. This approach empowers users to leverage component advantages while avoiding limitations imposed by the graphical user interface (GUI), bolstering system flexibility and interoperability. This sets the stage for adaptability to evolving requirements in the future.

Furthermore, the report outlines the different Application Areas, providing clarity regarding the envisioned systems. It acknowledges that this report represents a snapshot of the current definition of these areas, subject to refinement during implementation, especially in response to changes in the corresponding logical components of the INCODE architecture. Future steps involve the development and testing of these logical components to assess desirability, technical feasibility, and requirements demanded by the application areas.

A clear understanding of definitions and requirements is fundamental for the successful development and implementation of any project. The established process clearly defines project scope, objectives, and constraints, laying a solid foundation and minimizing ambiguity during INCODE logical component implementation. An exhaustive effort has been made to identify and document specific requirements, ensuring alignment with project needs and expectations. This clarity allows project teams to effectively plan, execute, and deliver outcomes that meet or exceed expectations. Continuous follow-up of these requirements during project execution ensures the inclusion of the defined requirements in the implementation of the INCODE logical components, ultimately paving the way for a successful project outcome.

Co-funded by
the European Union

References

[1] "NR and NG-RAN Overall Description; Stage 2", 3GPP TS38.300, Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3191

[2] "System architecture for the 5G System (5GS)", 3GPP TS23.501, Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3144

[3] "Management and orchestration; Concepts, use cases and requirements", 3GPP TS28.530, Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3273

[4] "E2E Network Slicing Architecture – Version 2.0", GSM Association Official Document NG.127, pages 48, May 10, 2022, Available: https://www.gsma.com/newsroom/wp-content/uploads//NG.127-v2.0-3.pdf

[5] "O-RAN Alliance", Available: https://www.o-ran.org/

[6] "TornadoVM", Available: https://www.tornadovm.org/

[7] "TornadoVM's source code repository", Available: https://github.com/beehive-lab/TornadoVM

[8] Pat Bosshart, Dan Daly, Glen Gibb, Martin Izzard, Nick McKeown, Jennifer Rexford, Cole Schlesinger, Dan Talayco, Amin Vahdat, George Varghese, and David Walker. 2014. P4: programming protocol-independent packet processors. SIGCOMM Comput. Commun. Rev. 44, 3 (July 2014), 87–95. DOI: https://doi.org/10.1145/2656877.2656890

[9] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. 2008. OpenFlow: enabling innovation in campus networks. SIGCOMM Comput. Commun. Rev. 38, 2 (April 2008), 69–74. DOI: https://doi.org/10.1145/1355734.1355746

[10] ETSI Open Source Group for TeraFlowSDN, Available: https://tfs.etsi.org/

[11] Open Networking Foundation (ONF): https://opennetworking.org/

[12] ONF Stratum OS: https://opennetworking.org/stratum/

[13] ETSI, "Network Functions Virtualisation: An Introduction, Benefits, Enablers, Challenges & Call for Action," 2014, Available: https://portal.etsi.org/nfv/nfv_white_paper.pdf

[14] ETSI, "ETSI unveils first cloud-native VNF management specifications", November 2020, Available: https://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/040/04.01.01_60/gs_NFV-IFA040v040101p.pdf

[15] ETSI, "Open Source NFV Management and Orchestration (MANO) software stack", https://osm.etsi.org/

[16] ETSI Software Development Group for OpenSlice, Available: https://osl.etsi.org/

[17] ETSI Context Information Management (CIM), NGSI-LD API, v1.7.1, June 2023, https://www.etsi.org/deliver/etsi_gs/CIM/001_099/009/01.07.01_60/gs_cim009v010701p.pdf

[18] 3GPP Release 17, December 12, 2020, Available: https://www.3gpp.org/release-17

[19] O-RAN alliance, 2021, Available: https://www.o-ran.org/

[20] Kok-Kiong Yap, et. al. "Taking the Edge off with Espresso: Scale, Reliability and Programmability for Global Internet Peering", ACM SIGCOMM '17). NY, USA, 432–445, Available: https://dl.acm.org/doi/10.1145/3098822.3098854

[21] Michael Dalton, et al. "Andromeda: Performance, Isolation, and Velocity at Scale in Cloud Network Virtualization" (NSDI'18). USENIX Association, USA, 373–387, Available: https://www.usenix.org/system/files/conference/nsdi18/nsdi18-dalton.pdf

Co-funded by the European Union

[22] Kubernetes: Production-Grade Container Orchestration, 2023, Available: https://kubernetes.io/
[23] AWS IoT: Unlock your IoT data and accelerate business growth, Available: https://aws.amazon.com/iot/
[24] Azure IoT: Quickly turn your vision into reality with secure, scalable, and open edge-to-cloud solutions from the Microsoft Cloud, Available: https://azure.microsoft.com/en-us/solutions/iot/#overview
[25] AWS IoT Greengrass, Available: https://docs.aws.amazon.com/greengrass/v2/developerguide/revise-deployments.html
[26] EU H2020 TERMINET: NexT GEneRation SMart INterconnectEd IoT, 2021, Available: https://terminet-h2020.eu/
[27] EU H2020 IoT-NGIN: IoT engine, to the Next Generation Internet of Things, 2021, Available: https://iot-ngin.eu/
[28] EU H2020 IntellIoT: Powering the future of humanized IoT and AI across Europe, 2021, Available: https://intelliot.eu/
[29] EU H2020 INGENIOUS: Next-Generation IoT solutions for the universal supply chain, 2021, Available: https://ingenious-iot.eu/web/
[30] EU H2020 VEDLIoT: Very Efficient Deep Learning in IoT, 2021. Available: https://vedliot.eu/
[31] EU H2020 ASSIST-IoT: Architecture for Scalable, Self-*, human-centric, Intelligent, Secure, and Tactile next generation IoT, 2021.
[32] OpenConfig: Vendor-neutral, model-driven network management designed by users, 2021, Available: https://www.openconfig.net/
[33] The YANG 1.1 Data Modeling Language, RFC 7950, August 2016, Available: https://datatracker.ietf.org/doc/html/rfc7950
[34] R. Mahmoud, "Internet of things (IoT) security: Current status, challenges and prospective measures" ICITST, 2015, pp. 336-341, Available: https://doi.org/10.1109/ICITST.2015.7412116
[35] Alfandi, O., et al. "A survey on boosting IoT security and privacy through blockchain". Cluster Comput 24, 37–55 (2021), Available: https://link.springer.com/article/10.1007/s10586-020-03137-8
[36] Tim Boivin, "Solving IoT's Access Control Conundrum", 2021, Available: https://portsys.com/solving-iots-access-control-conundrum/
[37] MoSCoW priorisation, https://www.agilebusiness.org/dsdm-project-framework/moscow-prioririsation.html
[38] TMForum 633 Service Catalog Management API, Available: https://www.tmforum.org/resources/standard/tmf633-service-catalog-api-user-guide-v4-0-0/
[39] TMForum 641 Service Ordering API, Available: https://www.tmforum.org/resources/specification/tmf641-service-ordering-api-user-guide-v4-1-0/
[40] TMForum 638 Service Inventory Management API, Available: https://www.tmforum.org/resources/specification/tmf638-service-inventory-api-user-guide-v4-0-0/
[41] The Linux Foundation, "OpenTofu: The open source infrastructure as code tool", Available: https://opentofu.org/
[42] Ceph: The future of storage, Available: https://ceph.io/en/discover/
[43] Hypershift, Available: https://hypershift-docs.netlify.app
[44] Configure multiple Kubernetes schedulers, Available: https://kubernetes.io/docs/tasks/extend-kubernetes/configure-multiple-schedulers/
[45] Kubernetes operator pattern, Available: https://kubernetes.io/docs/concepts/extend-kubernetes/operator/
[46] Kubevirt, Building a virtualization API for Kubernetes, Available: https://kubevirt.io

Co-funded by the European Union

[47] ETSI TS 119 182-1, “Electronic Signatures and Infrastructures (ESI); JAdES digital signatures”, Available: https://www.etsi.org/deliver/etsi_ts/119100_119199/11918201/01.01.01_60/ts_11918201v010101p.pdf

[48] ODRL Profile for Access Control “”, Dec. 2022, Available: https://protect.oeg.fi.upm.es/odrl-access-control-profile/oac.html

[49] ISO/IEC 25010, Available: https://iso25000.com/index.php/en/iso-25000-standards/iso-25010

[50] “Kubernetes's OpenAPI Specification”, Available: https://github.com/kubernetes/kubernetes/tree/master/api/openapi-spec

Co-funded by
the European Union