D5.1 Initial integration and proofs of concept plan

Document Summary Information

Project Identifier	HORIZON-CL4-2022-DATA-01. Project 101093129		
Project name	Agile and Cognitive Cloud-edge Continuum management		
Acronym	AC^3		
Start Date	January 1, 2023	**End Date**	December 31, 2025
Project URL	www.ac3-project.eu		
Deliverable	D5.1 Initial integration and proofs of concept plan		
Work Package	WP5		
Contractual due date	M21: 30th September 2024	**Actual submission date**	3rd February 2025
Type	R (Report, Document)	**Dissemination Level**	PU (Public)
Lead Beneficiary	CSG		
Responsible Author	John Beredimas (CSG)		
Contributors	John Beredimas (CSG), Asimakis Lykourgiotis (CSG), Ray Caroll (RHT), Eduardo Ojeda (IQU), Abdelhak Kadouma (FIN), Sara Madariaga (ARS), Dimitrios Amaxilatis (SPA) MEKKI Mohmed, MOKHTARI Ayoub (EUR)		
Peer reviewer(s)	Vrettos Moulos (UPR), Abd Elghani Meliani (EUR)		

Funded by the European Union

AC^3 project has received funding from European Union's Horizon Europe research and innovation programme under Grand Agreement No 101093129.

Revision history (including peer reviewing & quality control)

Version	Issue Date	% Complete	Changes	Contributor(s)
v0.1	27/05/2024	5%	Initial Deliverable Structure	John Beredimas (CSG) Asimakis Lykourgiotis (CSG)
V0.2	4/10/2024	50%	First round of contributions	Ray Caroll (RHT) Eduardo Ojeda (IQU) Abdelhak Kadouma (FIN)
V0.3	14/10/2024	60%	Initial review	John Beredimas (CSG)
V0.4	15/10/2024	65%	Executive Summary, Conclusions	Sara Madariaga (ARS) John Beredimas (CSG)
V0.5	14/10/2024	70%	UC1, UC3 updates	Ray Caroll (RHT) Eduardo Ojeda (IQU) Ayoub MOKHTARI (EUR)
V0.6	18/10/2024	80%	Second review, ToC	John Beredimas (CSG)
V0.7	22/10/2024	85%	Peer review (1/2)	Vrettos Moulos (UPR) John Beredimas (CSG) Asimakis Lykourgiotis (CSG)
V0.8	30/10/2024	90%	Peer review (2/2)	Abd Elghani Meliani (EUR) Dimitrios Klonidis (UBI) Amadou BA (IBM) Ryan Jenkins (RHT) Ben Capper (RHT) Eduardo Ojeda (IQU)
V0.9	31/10/2024	95%	Document cleanup	John Beredimas (CSG) Asimakis Lykourgiotis (CSG)
V0.9.2	13/11/2024	96%	Technical coordinator review comments (in-progress)	John Beredimas (CSG) Ray Caroll (RHT) Eduardo Ojeda (IQU) Dimitrios Amaxilatis (SPA) Ryan Jenkins (RHT)
V0.9.5	18/11/2024	98%	Final technical coordinator review	Adlen Ksentini (EUR)
V0.9.6	19/12/2024	99%	Project coordinator review comments	Christos Verikoukis (ISI/ATH) John Beredimas (CSG)

				Ben Capper (RHT) Eduardo Ojeda (IQU) Dimitrios Amaxilatis (SPA) Abdelhak Kadouma (FIN) Asimakis Lykourgiotis (CSG)
0.9.7	9/1/2025	99%	Final project coordinator review	Christos Verikoukis (ISI/ATH) John Beredimas (CSG) Ryan Jenkins (RHT) Eduardo Ojeda (IQU) Dimitrios Amaxilatis (SPA) Abdelhak Kadouma (FIN) Asimakis Lykourgiotis (CSG)
1.0	31/1/2025	100%	Final version	Asimakis Lykourgiotis (CSG) John Beredimas (CSG)

Disclaimer

Copyright message

Table of Contents

List of Figures

List of Tables

Glossary of terms and abbreviations used

Abbreviation / Term	Description
AC^3	Agile and Cognitive Cloud edge Continuum management
AI	Artificial Intelligence
API	Application Programming Interface
AppD	Application Descriptor
CECC	Cloud Edge Computing Continuum
CECCM	Cloud Edge Computing Continuum Manager
CRUD	Create, Read, Update, Delete
DNN	Deep Neural Network
GUI	Graphical User Interface
IoT	Internet of Things
JWST	James Webb Space Telescope
KPI	Key Performance Indicator
LCM	Life-Cycle Management
LISO	Lightweight Edge Slice Orchestration
LMS	Local Management System
ML	Machine Learning
NSD	Network Service Descriptors
OSM	OpenSource MANO
OSR	Ontology and Semantic aware Reasoner
PaaS	Platform as a Service
SD-WAN	Software-Defined Wide Area Network
SLA	Service Level Agreement
SOTL	Semantic-Aware and Ontology Templating Language
SSP	Single Stellar Population

UAV	Unmanned Aerial Vehicle
UC	Use Case
VLT	Very Large Telescope
VoD	Video on Demand
WP	Work Package
ZSM	Zero-touch Service Management

1 Executive Summary

The present document **D5.1 Initial integration and proofs of concept plan** provides a comprehensive description of the testbeds setup, deployed for the three use cases (UC) proposed in the project (Internet of Things (IoT) and Data, Smart Monitoring System using Unmanned Aerial Vehicles (UAVs), and Deciphering the universe: processing hundreds of TBs of astronomy data), as well as the initial plans for the integration and testing of the Cloud Edge Computing Continuum Manager (CECCM) components within these scenarios.

It is worth noting that a UC based approach to integration is followed, in which we integrate all the CECCM components as they appear in the architecture and then fit the UC. Starting from UC allows for earlier component testing and more flexibility to adapt to specific requirements and needs, in line with agile methodologies.
Section 2 contextualizes this document within the project, relating it to the work done in tasks *T5.1 AC^3 components integration* and *T5.2 Testbed integration*. This deliverable is key for the successful realization of the AC^3 vision, as it establishes the foundation for the validation of the proposed architecture in the subsequent phases of the project.
Section 3 recalls the updated AC^3 architecture detailed in deliverable D2.1 [1] incorporating the results of the work performed in Work Package (WP) 3 and WP4. This section also explains the three-planes model designed to facilitate seamless flows of information and control to effectively manage complex, data-driven and microservice-based applications:

1. The Infrastructure Plane encompasses the physical and virtual resources that form the backbone of the Cloud Edge Computing Continuum (CECC).
2. The User Plane serves as the interface for application developers, enabling them to interact with the system and define their application requirements.
3. The Management Plane constitutes the intelligent core of the CECCM that orchestrates and manages the CECC resources and applications.

Sections 4, 5 and 6 describe the three UCs in detail, explaining their challenges and how they can be solved by the AC^3 architecture and software components. These sections also provide comprehensive information about the UC mapping to AC^3 architecture, the experimental platform deployment, the AC^3 objectives that will be demonstrated, experimental real-world scenarios, and the key milestones. The three UCs have already developed their core application and successfully deployed them in their respective local environments. They are currently in different testing phases and planning the integration of the AC^3 components, in coordination with WP3 and WP4.

UC1, ***IoT and Data***, strives to optimize resource allocation in office buildings. It involves deploying sensors to monitor environmental factors and human presence, with the goal of maximizing occupant health and comfort by adjusting lighting and heating systems in real-time.

The primary challenges of UC1 include potential data transmission delays, ensuring data integrity and synchronization across multiple sources, efficient resource management to meet Service Level Agreements (SLAs), scalability to accommodate a growing number of devices, and robust data processing for accurate decision-making. By integrating AC^3 components such as Application gateway (Graphical User Interface - GUI), Ontology and Semantic-aware Reasoner (OSR), Local Management System (LMS) Edge, LMS Cloud, LMS

Networking, Catalogues, Application and Resource Management (Artificial Intelligence (AI)-based Life-Cycle Management (LCM), AI-based CECC Resource Profile, AI-Based Application Profile, and Monitoring), and Data Management (Data Connector, Data Mapper, Data Manipulator, and Message Broker), UC1 aims to demonstrate a scalable and efficient solution for optimizing building resource usage based on real-time data and AI-driven insights.

UC2, ***Smart Monitoring System using UAVs***, revolves around the development and implementation of a Smart Monitoring System utilizing UAVs, AI/ Machine Learning (ML) technologies, and edge computing to enhance video surveillance and environmental monitoring. Its primary goal is to optimize urban security, traffic management, and environmental tracking through the CECCM.

UC2 leverages AC^3 to address several challenges, including but not limited to hardware heterogeneity, unreliable network connectivity in far edge locations and large-scale video processing by distributed hardware that may lack the respective hardware accelerators required for said tasks. Towards this end, it will leverage the following AC^3 components to integrate the EURECOM edge facility as well as far-edge devices: Application gateway (GUI), OSR, LMS Edge, LMS Far Edge, LMS Cloud, LMS Networking, and Application and Resource Management (Monitoring, AI-based LCM, AI-Based Application Profile, and AI-Based Resource Profile).

UC3, ***Deciphering the universe: processing hundreds of TBs of astronomy data***, analyses large 3D data cubes of astronomy data, which contain a detailed image and spectral information about galaxies, to extract key insights such as stellar kinematics and population characteristics. Its main challenge is that handling these vast datasets requires significant computing power, memory and efficient data management, as well as scalable and distributed processing capabilities. Traditional single cluster systems struggle with processing and analysing such massive amounts of data. This UC has a significant relevance, as its results will have societal impact by enabling other researchers and organizations worldwide to adopt newer and more efficient ways of executing specific scientific software, and it is also a good example of a multidisciplinary work thanks to the contribution of the Group of Extragalactic Astrophysics and Astronomical Instrumentation from the UPM, leader of the UC.

The main objective of UC3 is to integrate the AC^3 network operator, which leverages Skupper to provide secure cross-cluster communication between multiple OpenShift (Kubernetes) clusters, to enable scaling across multiple environments and testbeds and escape the limited resources of single cluster environments. Towards this end, UC3 will integrate the following AC^3 components: Application Gateway, Network LMS, Application and Resource Management (Monitoring, AI-based LCM, AI-Based Application Profile, and AI-Based Resource Profile), and Data Management (Data connector, Catalogue).

Section 7 provides the conclusions of the deliverable and a summary of the AC^3 components utilized by the different UC, as well as a note about the validation of the resource federation.

In summary, this integration plan demonstrates AC^3's commitment to developing an innovative and adaptable Cloud Edge Computing Continuum Management system capable of addressing diverse, real-world challenges.

2 Introduction

This section establishes the context of the work detailed in this deliverable. It will outline the main purpose of the deliverable, its core objectives, and how it connects to the broader project framework and associated deliverables. Additionally, this section will delineate the anticipated outcomes and their alignment with the commitments outlined in the Grant Agreement. It concludes by providing the structural organization of the deliverable.

2.1 Purpose and objectives

The CECC paradigm integrates cloud and edge resources into a unified, synergistic infrastructure. The Agile and Cognitive Cloud-edge Continuum Management (AC^3) architecture facilitates this integration, which is crucial for unlocking the full potential of modern computing, enabling modern microservice-based applications to adapt dynamically to changing conditions. At the core of this architecture lies the CECC Manager, which intelligently orchestrates the deployment and migration of microservices across the continuum, delivering optimal performance, and minimizing energy consumption. Additionally, AC^3 data management capabilities, offered as a Platform as a Service (PaaS), ensure efficient and secure data handling throughout the CECC, further enhancing the overall performance and agility of the system.

The purpose of this deliverable (D5.1) is to report on the initial integration and testbed validation plans within the project. It outlines the updated AC^3 architecture, detailing how the various components, algorithms, and enablers from different work packages are to be integrated to form the CECCM. We drafted an agile integration strategy, focused on providing the UC with the required CECCM capabilities as they are needed, allowing also for earlier component validation. The document further explores the intricacies of each UC, presenting the experimental platform deployments, experimental scenarios, and key milestones. This deliverable establishes a strong foundation for the subsequent phases of the project, paving the way for the successful realization of the AC^3 vision. It finally demonstrates how the requirements and architecture of these UC map to the AC^3 CECCM, validating its architecture.

2.2 Link with other project activities

This deliverable provides insights into the initial developments, integrations, and proof of concept activities related to Task T5.1. The work done in the UC adheres to the requirements defined in D2.4, "Business Analysis of CECC and Use Case Requirements" [2]. Regarding the data management mechanism, D3.3, "Initial Report on Data Management for Applications in CECC" [3], provides direct feedback for the design of the overall system and its interface with AC^3 service deployment and management mechanisms. D4.1, "Initial Report on Mechanisms that Enable Green-Oriented Zero Touch Management of CECC Resources" [4], informs the work on resource discovery and monitoring, AI/ML models for resource management, green-oriented LCM decisions for resource management, and networking programmability of CECC. D3.1, "Initial report on the Application LCM in the CEC" [5], navigates through the various components related to the user plane of the CECCM, namely the User Interfaces, Application Profiles, Ontology Modeling Tools and Application Descriptor Models. Moreover, D2.3 “Report on technological tools for CECC” [6] provides a comprehensive overview of the technological tools, laying the groundwork for their integration within the AC^3 framework. The careful selection and analysis of these tools in D2.3 was a crucial initial step toward ensuring a smoother integration process in subsequent work packages. Finally, D2.1 “CECC framework and CECCM” [1] provides the architecture framework for all UC.

2.3 Mapping AC³ Outputs

The purpose of this section is to map AC³ Grant Agreement commitments, both within the formal Deliverable and Task description, against the project’s respective outputs and work performed.

Table 1. Adherence to AC³ GA Deliverable & Tasks Descriptions

AC³ GA Component Title	AC³ GA Component Outline	Respective Document Chapter(s)	Justification
DELIVERABLE			
D5.1 Initial integration and proofs of concept plan: *“Description of the testbeds, as well as initial implementation and integration plans to be done during the project.”*			
TASKS			
Task T5.1: AC³ components integration	*“Each partner involved will develop their individual components and/or functions and show their project results based on the tests done in their labs, where all essential functions can be tested.”*	Sections 4.3, 5.3 and 6.3	The “Use Case Mapping to AC³ Architecture” subsection for each UC details how and which AC³ components and functions are developed by different partners in order to be integrated into the CECCM software.
Task T5.2: Testbed integration	*“This task will concern the integration of the software and hardware needed to run the three UCs.”*	Sections 4.4, 5.4 and 6.4	The “Experimental Platform Deployment for the Use Case” subsection for each of the three UCs provides detailed descriptions of the technical infrastructure and setups used for each of the three UCs in the AC³ project. It outlines the hardware, software, and networking components that enable the implementation and testing of the specific scenarios and objectives of each UC.

2.4 Deliverable Overview and Report Structure

In this section, a brief description of the Deliverable’s Structure has been provided, as follows:

- Section 3 presents a summary of the updated AC^3 architecture and details how the various algorithms and enablers from different work packages are integrated into the CECCM software. Note that an exhaustive presentation of the architecture, its components and workflows thereof is outside the scope of this document and is instead available in deliverables D2.1 and D2.2.
- Sections 4, 5, and 6 describe the implementation and integration plans for UC1, UC2, and UC3 respectively. While the UCs are separated for clarity, all three sections share a common internal structure to ensure consistency, as follows:
    1. Challenges: This subsection identifies the key challenges specific to each UC. It separately considers "Use Case Challenges" that directly relate to the UC core functionality and "Integration Challenges" that are encountered when integrating the UC with the AC^3 framework.
    2. AC^3 Objectives to Be Demonstrated: This subsection details the specific AC^3-related objectives targeted by each UC, explaining what aspects of the AC^3 framework are being validated.
    3. Use Case Mapping to AC^3 Architecture: This subsection details the mapping of the UC to the overall architecture of the AC^3 framework, showing the interactions of the different components, and the way each UC leverages AC^3 components.
    4. Experimental Platform Deployment for the Use Case: This subsection describes the infrastructure and setup of the testbed used for each UC, including hardware, software, and network configurations.
    5. Experimental Scenarios: This subsection describes the experimental setups and scenarios that will be used to test and validate the UC implementations, outlining the processes and metrics for evaluation.
    6. Milestones: This subsection outlines the key milestones for each UC, providing a structured roadmap for development and ensure a systematic approach to achieving the UC's overall goals.
- Finally, Section 7 summarizes the deliverable's key achievements and reiterates the significance of the work performed, providing a final overview of the progress made and the value proposition of the AC^3 project.

3 AC³ Architecture and Initial Integration

The final AC³ architecture is illustrated in Figure 1. This updated version reflects the initial design outlined in D2.1, incorporating the results of development work within WP3 and WP4.

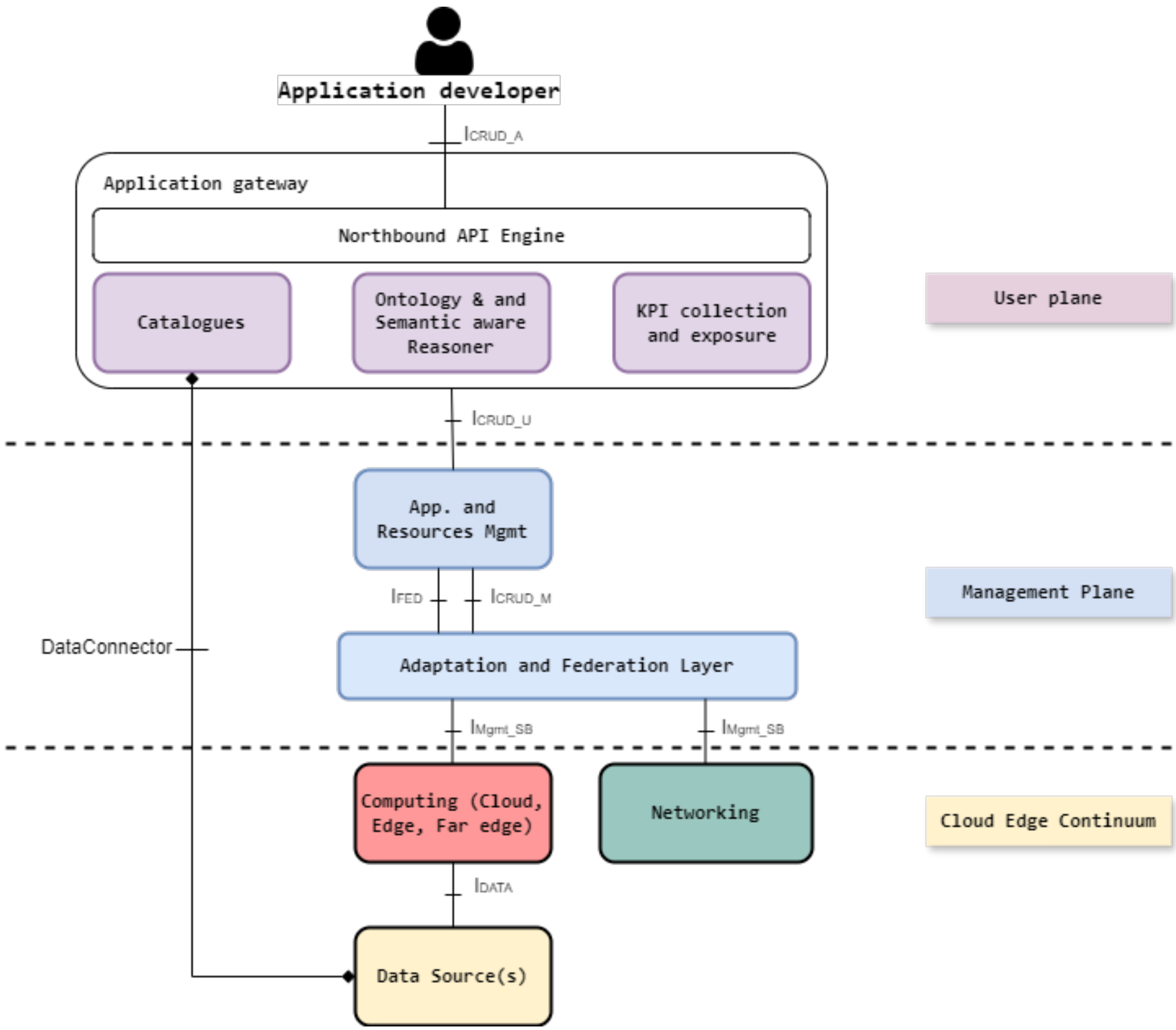

Figure 1. Updated Architecture of AC³ framework.

The AC³ architecture is a sophisticated framework designed to manage the CECC with agility and cognitive capabilities. It comprises three distinct planes: the User Plane, the Management Plane, and the Infrastructure Plane. The user and management planes comprise the CECCM. The CECC itself encompasses the Infrastructure Plane, which includes data sources, computing nodes (central cloud, edge, and far edge), and networking resources. The User Plane serves as the interface for application developers, enabling them to interact with the system and define their application requirements. Moreover, data management in AC³ is built around the

Catalogue that inventories available data resources. Data Sources, classified as Hot or Cold based on access frequency, are connected to the system via Data Connectors. These connectors facilitate data flow between the AC^3. The Management Plane acts as the intelligent core, orchestrating and managing the CECC resources and applications. It leverages AI/ML algorithms to automate and optimize resource allocation, ensuring efficient and reliable application deployment and execution. The Infrastructure Plane encompasses the CECC's physical and virtual resources, including cloud, edge, and network components. This layered structure facilitates a seamless flow of information and control, enabling the AC^3 system to effectively manage complex, data-driven, and microservice-based applications.

The User Plane is responsible for translating application developers' intents into exhaustive application descriptors. This is achieved by utilizing four key components: the Service Catalogue, the Northbound Application Programming Interface (API) Engine, the OSR, and the Key Performance Indicator (KPI) Collection & Exposure mechanism. The Service Catalogue acts as an expandable repository of blueprints for application descriptors, while the Northbound API Engine exposes all available functionalities to a user-friendly interface for developers to interact with the system. The OSR enables intelligent interpretation and translation of user intents and policies, ensuring accurate and efficient execution. The KPI Collection & Exposure mechanism gathers and presents performance metrics for developers to monitor and optimize their applications.

The Management Plane is the heart of the AC^3 system, responsible for orchestrating the CECC resources and applications. It consists of two primary components: Application and Resource Management, and the Adaptation and Federation Layer. The Application and Resource Management component oversees the deployment, scaling, and migration of applications, ensuring optimal performance and resource utilization in an energy-efficient manner. The Adaptation and Federation Layer enables seamless integration and interoperability with diverse infrastructure providers, expanding the reach and capabilities of the AC^3 system.

Finally, the Infrastructure Plane encompasses the physical and virtual resources that form the backbone of the CECC. It includes a wide range of components such as cloud servers, edge and far-edge devices, network infrastructure, and data storage systems. These resources are dynamically managed and orchestrated by the Management Plane to meet the evolving demands of applications and users, ensuring a seamless and responsive user experience.

In the subsequent sections of this deliverable, we examine each AC^3 UC, consider its challenges, and identify the relevance of the aforementioned architecture and components thereof in addressing said challenges. Finally, a proof of concept plan is presented, which details phases and milestones for integrating these UCs with the AC^3 CECC.

4 Use Case 1 Integration Setup

UC1 introduces an innovative IoT-based, smart sensing and monitoring framework designed to leverage the transformative potential of edge AI technologies within a CECC infrastructure. It aims to enhance the monitoring and management of infrastructures ranging from individual smart homes to expansive smart grids on a national scale, regardless of the underlying technologies for data collection and data communication. In this context, UC1 is engineered to integrate physical and digital realms more seamlessly than ever, thereby managing and processing a significantly larger volume of IoT data to enable timely decisions and responsive actions based on sensed conditions. It also focuses on data fusion, integrating outputs from diverse sensors to create detailed profiles and detect patterns that help in proactively managing events and minimizing their impact on infrastructure operations. This advanced functionality not only supports basic applications like air quality monitoring but also intends to enable immediate, localized decision-making through a blend of IoT innovation and edge computing intelligence.

4.1 Challenges

4.1.1 Use Case Challenges

An IoT-based smart sensing and monitoring framework with edge AI capabilities provides significant benefits for infrastructure operators. However, as outlined in D2.4 [2], it introduces several key challenges that must be addressed for the system to operate successfully:

- **Sensor Integration and Calibration**: Integrating multiple sensor types and ensuring accurate calibration can be complex. Different sensors may have varying levels of measurement accuracy, calibration procedures, and communication protocols.
- **Data Integration**: Many infrastructures utilize legacy systems with existing data sources. Integrating data from these diverse sources into a unified monitoring framework can be challenging but is often necessary to gain comprehensive insights.
- **Edge Device Reliability**: Edge devices, such as gateways or servers, must operate consistently regardless of deployment conditions. Ensuring their robustness and fault tolerance is essential for maintaining the system's uninterrupted operation.
- **Connectivity and Network Issues**: Ensuring reliable connectivity between sensors, edge devices, and the central system can be complex, particularly in remote or geographically dispersed areas. Network failures, signal interference, and limited coverage can all impact data transmission and system responsiveness.
- **Data Management and Processing**: Managing large volumes of sensor data requires effective data management and processing capabilities. Handling data in real-time, storing and retrieving it efficiently, and performing meaningful analyses can be challenging—especially when dealing with diverse data formats and sources.
- **Data Security and Privacy**: Protecting data security and privacy is paramount, especially when handling sensitive IoT data that travels across various points on the cloud-edge continuum. Robust security measures are necessary to prevent unauthorized access, ensure data integrity, and comply with privacy regulations like General Data Protection Regulation. Data security protocols must adapt to the dynamic

nature of data as it moves across environments, mitigating risks without compromising performance or accessibility.

- **System Scalability**: Scalability across the cloud-edge continuum is essential to accommodate expanding infrastructure and increasing data volumes. As more sensors are deployed and data flow rates grow, it is critical for systems to scale to prevent storage, processing, or performance bottlenecks dynamically. Scalability solutions must ensure that new devices, expanded functionalities, and heightened data demands are met without compromising system performance or operational continuity.
- **Deployment and Maintenance**: Installing and maintaining the infrastructure across different environments and locations can be logistically complex. Ensuring proper installation, configuration, and ongoing maintenance, including software updates, hardware repairs, and device replacements, requires efficient management and resource planning.

4.1.2 Integration Challenges

The integration of UC1 presents specific technical challenges that must be addressed to guarantee the successful deployment and operation of the system:

- **System Integration and Testing**: Ensuring that all components of the UC1 architecture, from IoT devices to deep learning models, are seamlessly integrated and function cohesively.
- **Scalability and Resilience**: Developing the infrastructure to be scalable and resilient, allowing for the dynamic migration of microservices across devices and platforms as needed to maintain continuity and performance.
- **Resource Optimization**: Leveraging AI-based orchestration within the CECCM to optimize the allocation and utilization of computing resources across cloud and edge nodes, reducing latency and enhancing system responsiveness.
- **Latency optimization**: Managing and analysing real-time streaming IoT data requires low latency computational pipelines. AC^3's federated architecture efficiently places workloads across cloud and edge environments to achieve this.

4.2 AC^3 Objectives to be Demonstrated

UC 1 aims to demonstrate the CECCM's capabilities focusing on the following specific objectives:

- **Intuitive application definition**: Leveraging an intuitive GUI and OSR, the application developer can efficiently define and deploy micro-service applications within the CECCM framework. This system will simplify the user experience, making it easier to harness cutting-edge technology, including cloud and edge domains and AI/ML capabilities.
- **Automatic Deployment and Zero-Touch Management**: UC1 will demonstrate the robust LCM capabilities of the CECCM developed in AC^3, enabling the automatic deployment, monitoring, and maintenance of micro-service applications. Using AI and ML-driven zero-touch management algorithms, the system can autonomously optimise and sustain application performance.
- **Microservice Deployment and Migration**: The CECCM's resilience will be further highlighted by its ability to deploy and manage microservices across cloud-edge environments. Should the application in an edge deployment become unavailable or face resource limitations, the system can automatically migrate services to alternative cloud infrastructures, ensuring uninterrupted service delivery and continuity.

- **Data Analysis and Decision-Making for Smart Building Installations**: By leveraging AI and ML techniques, the system will enable real-time analysis for the environmental and human presence detection data. This real-time data analysis will contribute to enhanced building monitoring for actionable insights and facilitating AI based decision-making.

KPIs:

- **Traffic Reduction:** Treating data locally on far-edge devices reduces the edge to cloud traffic by at least 50%, minimizing the need for data transmission to the central cloud.
- **Latency Optimization:** Edge computing allows us to reduce the time needed to process and reach to sensor data reported to under 10 milliseconds when executing the application at the edge of the network.
- **High Availability:** The system ensures a high reliability level with an uptime greater than 99.9%, leveraging the CECCM's zero-touch management and predictive availability features.

4.3 Use Case 1 Mapping to AC^3 Architecture

Figure 2 showcases both the UC1 application components as well as the AC^3 components that it leverages for its execution across the CECC. The UC1 application itself consists of the following:

1. **Monitoring and Visualization Component**: This component, based on the use of Grafana, visualizes key metrics computed from the data received from the IoT data source used in real time. It is built as a set of dashboards that showcase the operation and the condition in each room of the testbed and the actions taken by the application's AI. This component is adapted by SPA for use in the UC1.
2. **Data Manipulation Components**: These components are the core of the UC1 AI application. They are split in two categories, Anomaly Detection and Forecasting.
    a. **Anomaly Detection Component**: This component focuses on assessing the quality of the data received through the data source of the application. It checks each data point received against a ML model for inconsistencies to detect faulty data, either due to hardware malfunctions or sensor tampering. This component was developed by SPA for use in the UC1 as an AC^3 Data Management application addon.
    b. **Forecasting Component**: This component receives the data identified by the Anomaly Detection Component as valid and uses them to forecast the conditions inside the building as well as take any decisions needed for its proper operation. This component was developed by SPA for use in the UC1 as an AC^3 Data Management application addon.
3. **Data Mapping Components**: These refer to components needed to prepare the data received from the UC data source to be used by the Data Manipulation components. These include operations like format changes or data batching. This component is developed by SPA for use in the UC1 as an AC^3 Data Management application addon.
4. **Data Broker**: This component is responsible for delivering data received from the data source to the rest of the application's components. It is built using the well-established RabbitMQ message broker.
5. **AC^3 Infrastructure Plane**: This component serves as a central hub to host the UC1 AC^3 application microservices in case the data processing exceeds the capabilities of the Edge and Backend Edge Cloud.

This approach ensures that the UC1 AC^3 application can run smoothly even during periods of data deluge, seamlessly shifting between the edge and AC^3 cloud across the CECC to avoid SLA degradation. This component is adapted by SPA for use in the UC1 as an AC^3 Data Management application addon.

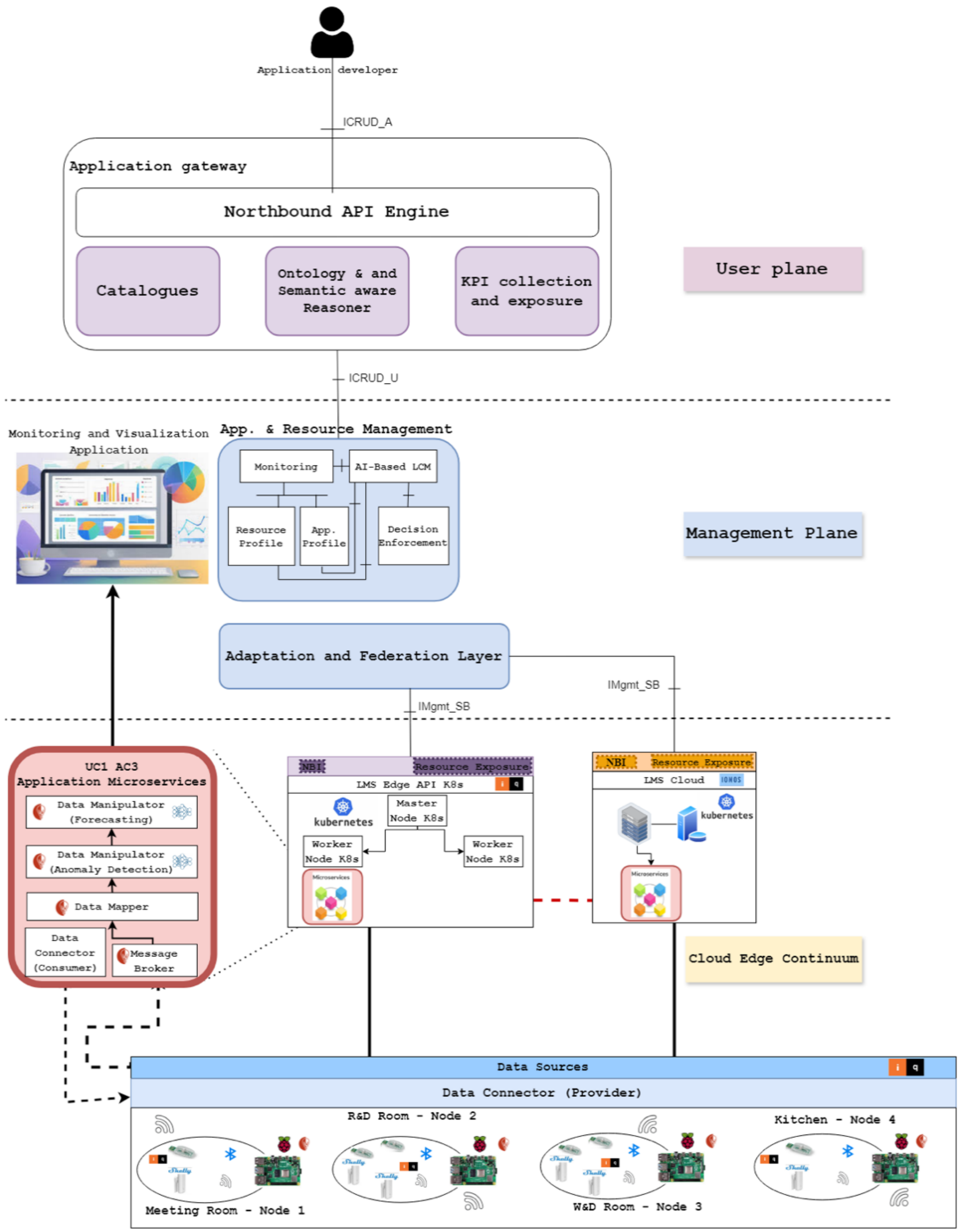

Figure 2. UC1 architecture following the AC^3 general architecture paradigm

The AC^3 components leveraged in the UC1 are the following:

1. **CECCM**: The CECCM capabilities are harnessed to enable microservice initial placement migration across the CECC infrastructure powered by AI-based orchestration. Combined with zero-touch configuration and application management, it streamlines the data life cycle, potentially accelerating application development and deployment across the CECC. Thanks to the CECCM, the UC1 AC^3 application can enhance its throughput.
2. **Application gateway (GUI):** The application gateway is used to describe and define the UC1 application, select the data sources to be used and any required placement guidelines or SLAs for its operation.
3. **Catalogues:** Service and Data Catalogues are used to store the available application services for the UC and the information for the UC IoT Data Sources.
4. **OSR:** The OSR is used to generate the application descriptor before deployment via the AI-based LCMs.
5. **Application and Resource Management**: This component uses AI-driven LCM for application deployment, runtime management, and resource optimisation. This architecture makes use of the following five elements that form this module:
    a. AI-based LCM,
    b. Decision Enforcement,
    c. AI-based resource profile,
    d. AI-based application profile, and
    e. Monitoring.

The GUI is the entry point for the application developer to interact with the CECCM framework. This GUI is designed to simplify creating and managing application descriptors, which define the UC1 AC^3 application microservices components and SLA. Moreover, the GUI facilitates the comprehensive management of the application's lifecycle from a single point. Thanks to the service and data catalogue, the application developer has application blueprints, available services, and data sources in the CECC.

The OSR is an intelligent reasoning engine that processes the application descriptor. It interprets and adapts the human-defined policies and intents provided by the application developer. The OSR translates these intents into machine-processable ontology instances and semantic rules that other CECCM components can utilise.

At this stage, the AI-based LCM is tasked with orchestrating the deployment of the microservices as dictated by the application descriptor translated by the OSR. It determines the correct level of the CEEC for initially deploying each microservice, whether on the LMS Edge or LMS Cloud, based on the intended SLA. Beyond deployment, the LCM oversees the entire lifecycle of these services, managing scaling, migration, and fault tolerance to guarantee ongoing service despite possible node failures or system changes.

The UC1 AC^3 application serves as the foundation of UC1 and is tasked with data reception, formatting, and processing through ML models. A notable characteristic of this application is its capability to seamlessly migrate between edge and cloud LMS after its initial deployment to maintain SLA compliance. In the case of computational-intensive tasks, resource constraints or demand spikes, microservices conforming the UC1 AC^3 application should migrate to prevent service degradation. This adaptability enables it to prioritise low-latency performance or increased computational power, facilitating efficient data processing and real-time decision-making under diverse conditions. Table 2 outlines the contributions of various functional components to UC1, detailing the architecture components, their sub-components, descriptions, and the technological tools or partners involved.

Table 2. AC^3 Functional Components used in Use Case 1

Architecture component	Sub-Component	Description	Technological tool/Partner
Application gateway (GUI)		Allow the application developer to define its application components and SLA.	GUI (FIN)
OSR		Allow the generation of the Application Descriptor	OSR (FIN)
LMS Edge		We will execute the micro-services that run at the network's edge for lower latency and bandwidth optimisation.	Kubernetes API (IQU)
LMS Cloud		Will execute the micro-services that cannot run at the network's edge due to high resource usage or data volumes, and latency is not a constraint. Similarly, for network unavailability at the edge.	ION Cloud (ION)
Catalogues		Hosts the component templates for the application	Catalogue from WP3 (ION)

Application and resource management	AI-based LCM and Decision Enforcement	1. Manage the microservices Life Cycle 2. Migration algorithm that adapts if the edge resource degrades or moves to the edge a micro-service	1. Service Orchestration tool over Kubernetes (EUR, UBI) 2. Proactive Stateful Container Migration Based on Resource Utilization in WP3 (EUR)
	Zero-touch configuration and application management, data management	Predict and describe infrastructure resources and implement automated corrective measures.	Using machine learning and deep learning approaches initiated in D4.1 (Section 5.3.2 and 5.3.3)
	AI-Based Resource profile	Describe the resources of the infrastructure	Resource exposer (EUR)
	AI-Based Application profile	Predicting Application Behaviour	Algorithm on app behaviour prediction from WP3 (FIN)
	Monitoring	Monitoring the micro-services KPI	Monitoring tools (EUR)
Data Management	Catalogues	Provides descriptions for the available data sources	Data Management (SPA)
	Data Provider Connector	Provides access to the data made available	Data Management (SPA)
	Data Consumer Connector	Initiates the streaming of data from the data source to the application micro-services	Data Management (SPA)

	Data Mappers	Transforms incoming data as needed	Data Management (SPA)
	Data Manipulator	Core application logic	Data Management (SPA)
	Message Broker	Responsible for transferring data between application components	Data Management (SPA)

4.4 Experimentation Platform Deployment for Use Case 1

An IoT-based smart sensing and monitoring environment needs to combine two main components. The **IoT infrastructure**, that is responsible for generating and collecting data from sensors deployed throughout the installation (e.g., a building or a factory) and the **computing infrastructure** that deals with processing the collected data to generate the expected conclusions and interventions (e.g., environmental monitoring, security control, automation etc.). In UC1 these two components are depicted as a following:

- The **IQU Offices IoT Data Source Domain** represents the **IoT infrastructure** to be used for UC1. Figure 3 depicts this deployment that combines the following: *IoT devices* that generate environmental and human presence measurements from the 4 office spaces IQU operates in Barcelona and *data collection nodes* that are used to gather and forward these measurements as a real-time IoT dataset to any application that needs them. The monitored areas within the IQU office building are detailed in Figure 2, showcasing four distinct spaces: 1) the Meeting Room; 2) the R&D Room; 3) the W&D Room; and 4) the Kitchen.

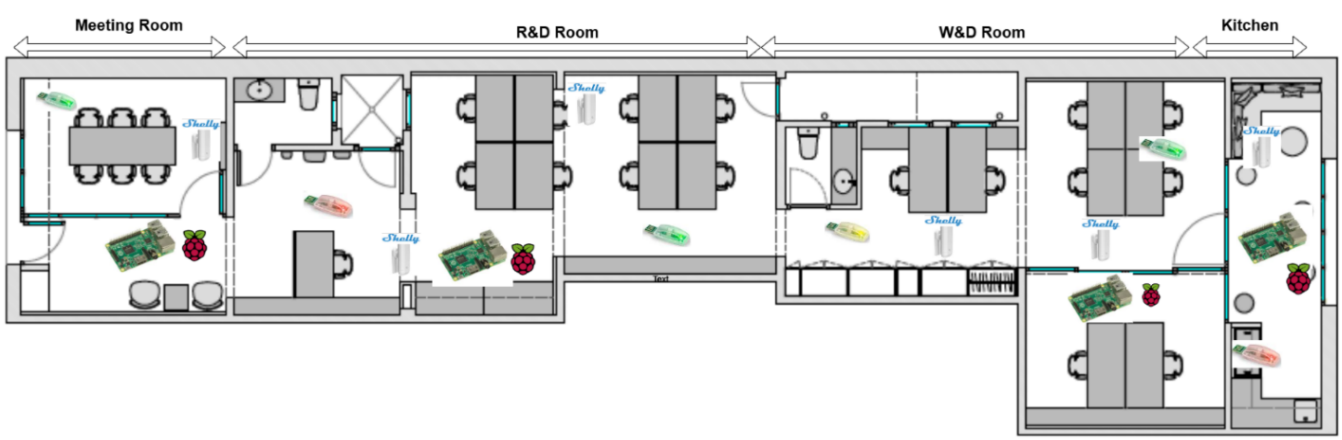

Figure 3. The IQU Offices IoT Data Source Domain that represents the IoT infrastructure of UC1.

- The Edge and Cloud domains computing infrastructure hosts the UC1 application, which consumes the IoT data generated and applies ML models to assess the occupancy of office spaces based on both environmental and presence data from the IoT infrastructure. This setup minimizes the need for constant cloud connectivity. These two computing domains will help us showcase the local processing benefits of using AC^3 for applications that consume locally produced data, namely reduced latency and real-time decision-making, as well as present how cloud resources can help provide a responsive service regardless of the volume of data we need to process.

The UC1 testbed that combines IoT, edge, and cloud is illustrated in Figure 4 and includes:

- **Edge Domain**: Edge servers form a Kubernetes cluster, serving as the edge domain of UC1. They allow for improved resource management and scalability by enabling Kubernetes to orchestrate the containerized UC1 AC^3 application as a local management system. The edge domain acts as a location where the analysis of the IoT data is as close to their source as possible reducing communication costs and processing latency. The processing capabilities of this location is limited though and in cases of excessive traffic generated by the IoT infrastructure the time to process the respective data would increase beyond the needed SLAs.
- **Cloud Domain**: UC1 will use a Kubernetes cluster hosted by ION to act as its cloud services domain. This cluster will help us showcase the capabilities of service-migration in AC^3. Cloud services offer us access to “unlimited” resources, that would be capable of handling any volume of data originating from the IoT domain. Under circumstances where the IoT infrastructures generate overwhelming data for the Edge domain, processing can be migrated to the cloud, increasing latency but ensuring that all data is processed respecting the SLAs of the defined. Additionally, specific services that are used for performance data collection and visualization of the data consumed by the UC application will be hosted in this cloud domain. These tools enable monitoring and visualisation of our local infrastructure's performance metrics. Prometheus collects data regarding the operation of our application’s components performance, and Grafana provides interactive dashboards for data visualization, analysis and alerting.
- **Data Source Domain**: Deployed primarily for sensing, collecting, and transmitting IoT data in real-time. This domain utilizes:
    - **IoT Sensors**: The IQU building monitoring uses Sensirion SCD41 CO_2 sensors, and Shelly Motion 2 that have been deployed as the building’s IoT monitoring infrastructure.
    - **Raspberry Pi4 & Pi5 devices**: These devices are the leading platforms used in the Far Edge Domain. They are compact and cost-effective single-board computers capable of running software to manage sensor data. They are equipped with SIM8200EA-M2 5G HATs, providing 5G connectivity. However, Wi-Fi can also be used as an alternative.

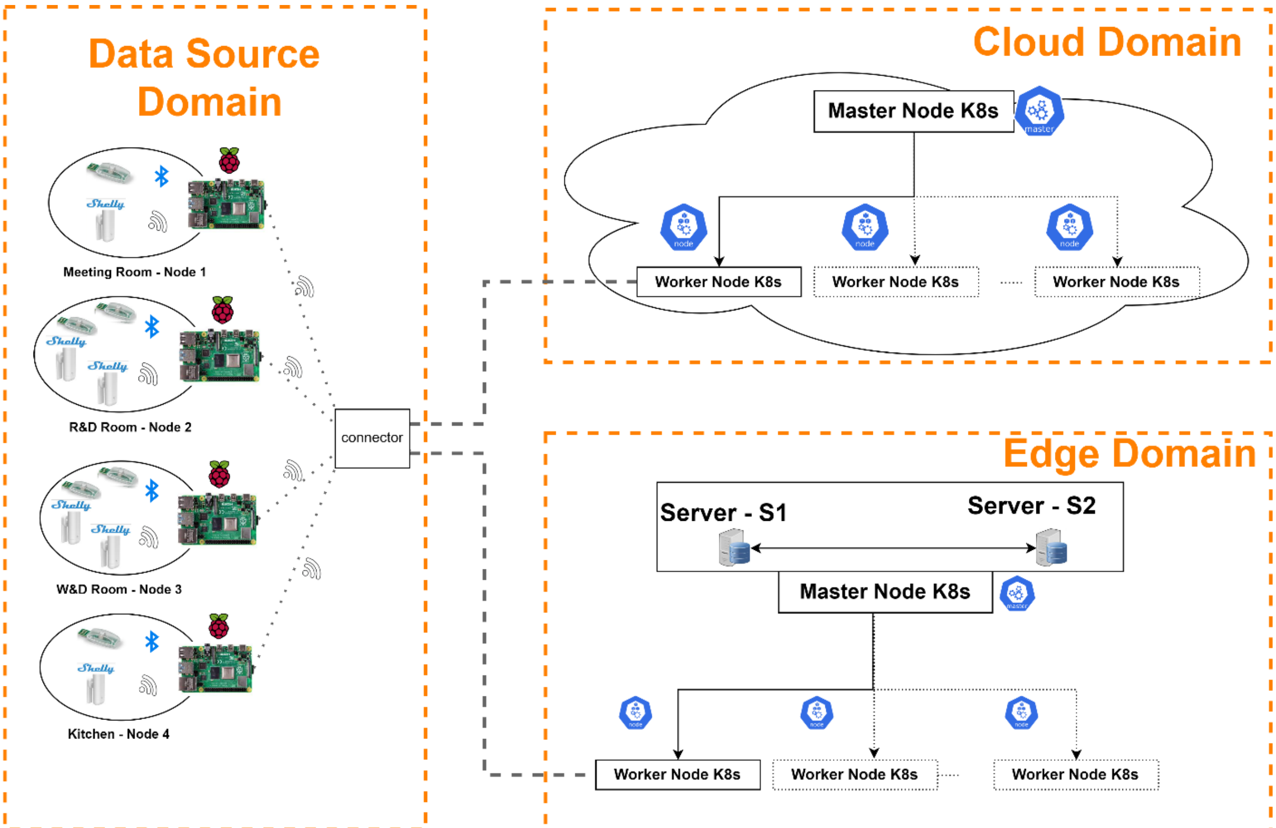

Figure 4. Depiction of the UC1 Data Source, Edge and Cloud Domains and the flow of data between them.

4.5 Experimental Scenarios

UC1 showcases the CECCM's capabilities to accelerate microservices deployment at the edge of monitored infrastructure. Moreover, the CECC infrastructure enables the development of microservice-based applications that leverage edge computing for lower latency, improved data security, and privacy. The experimental scenarios to be evaluated are the following.

Zero-touch configuration, and application management, data management: Leverage the CECCM to facilitate service configuration, deployment, and management across the cloud-edge continuum. Additionally, the framework offers data management as a service (PaaS), simplifying data operations and accelerating insights. These features accelerate the development process for developers. The CECCM's intuitive GUI and the OSR provide a platform for flexible application management, facilitating seamless configuration, deployment, and reconfiguration of microservices to adapt to changes in the environment. Data management includes the selection and deployment along with the application's business logic of the application addons needed for requesting, retrieving, transforming and disposing of the data as needed.

Application Descriptor Composition and OSR Integration: The GUI of the Application Descriptor Composer serves as the main interface for creating and managing application descriptors, allowing developers to create, read, update, and delete descriptor models. Once the descriptor is created, it is transmitted to the OSR, which interprets it using semantic web technologies (e.g., ontologies and reasoners). This step ensures that all application components, policies, and requirements are properly represented. By connecting with the POSSIBLE-X catalogue, the OSR can access data and service catalogues to identify and retrieve blueprints that meet the application's specified needs. This integration results in a comprehensive YAML file, which aligns with application requirements and is passed to the LCM for deployment across the cloud-edge infrastructure.

Time to process and react to sensor data: The CECCM empowers the UC1 to deploy and run microservices at the edge of the monitored infrastructure, enabling rapid sensor data processing. This approach offers significant benefits in the time to process and react to sensor data.

Seamless Microservice Deployment and Migration: The LCM component orchestrates the deployment of microservices across the cloud-edge continuum. Service migration across the cloud-edge continuum requires flexibility in moving services dynamically to adapt to workload changes, optimize latency, or enhance data locality. Balancing resource usage and managing migration transparently to end-users is key to maintaining service performance and user satisfaction. Effective migration capabilities enable applications to adjust to the most optimal points across the continuum, ensuring uninterrupted functionality and efficiency. The LCM ensures that, in cases of resource degradation or unavailability, micro-services can seamlessly migrate across cloud and edge domains. This functionality guarantees high availability and uninterrupted service, making real-time adjustments based on environmental conditions.

AI-powered Infrastructure Monitoring & Control Service at the Edge: Highlight an AI-powered edge service that improves the infrastructure depicted in Figure 2 by providing real-time monitoring and control. By bringing computation closer to the data source, this service can notify when conditions deviate from established standards. This capability enhances immediate corrective actions, helping to reduce downtime and mitigate potential risks.

The UC1 deployment flow will be illustrated through a series of deployment steps that showcase the functionalities and capabilities of the AC^3 framework and its components in the context of UC1:

1. **Streaming data source deployment (Initial Setup):**

Deployment of the data collection infrastructure of the UC and all the configuration needed for the definition of the data source in the AC^3 Data Catalogue. Raspberry Pi devices with Sensirion SCD41 CO_2 and Shelly Motion 2 sensors has been deployed in multiple rooms of the IQUADRAT´s headquarters´ offices, i.e. in the Meeting Room, R&D Room, W&D Room, and Kitchen, as illustrated in Figure 3.

2. **Application through the CECCM (Initial Setup):**

Deployment of the application through the CECCM of AC^3 and incorporates multiple steps that include the initial definition of the application, before any migration- or scaling-related decisions are employed, its building blocks, and its dependency to data sources. It involves the use of different core components of AC^3 the GUI, OSR, Service and Data catalogue for application composition as well as the Application and Resource Management for the initial deployment of the application.

3. **Data Access through the CECCM:**

Validation of the use of Data Connectors to transfer data from the data source's location to the AC^3 application's execution location either at the edge or at the cloud location available.

4. **UC Application:**

Operation of the business logic of the UC by demonstrating the usage of the internal components of the UC, like the Data Mapper and Data Manipulator. The Data Mappers are evaluated on the versatility to transform the received data correctly to the internal application formats, while Data Manipulators use them to apply the ML models needed for forecasting and classifying the received data. The Data Manipulator within the UC1 AC^3 application will integrate the following ML models (provided by SPA).

- An Anomaly Detection model to identify irregularities in the sensor data (CO_2 concentration, temperature, and humidity).
- A predictive model to forecast the subsequent data trends.
- Additionally, classification models will be used based on the combined data to identify characteristics and situations like occupied or unoccupied rooms.

5. **Application Migration:**

Using CECCM capabilities, the UC1 AC^3 application microservices can be dynamically migrated between edge and cloud locations. This migration is driven by specific requirements, such as low-latency processing or increased computational power, enabling the system to adjust to varying operational conditions. We will showcase how AC^3 can help easily migrate the application between different execution environments without the intervention of the application developer.

6. **Application End-to-End Operation:**

We showcase how the use of the AC^3 can help reduce the resources computed by the UC application leveraging the use of the edge computing resources. Based on the execution of the application closer to the data producers, we will be able to reduce the data traffic needed to transfer the data from the data source to the processing location and monitor these gains through the AC^3 monitoring components.

4.6 Milestones

Table 3 lists the key milestones, highlighting activities and outlining our strategic objectives. These milestones cover essential steps, such as defining UC goals and selecting appropriate hardware and software. Each milestone ensures a structured approach to the implementation and continuous improvement of the UC.

Table 3. Integration Plan Milestones for Use Case 1

S/N	Implementation
1	UC goals and objectives definition
2	UC hardware and software selection
3	UC architecture definition

4	Testing setup of the data collection testbed infrastructure
5	Testing data collection and development of the UC application components (including ML algorithms)
6	Deployment of application components to cloud location using docker containers
7	Development of a dashboard interface for monitoring the operation of the UC components
8	Deployment of the UC K8s edge location
9	Kubernetization of the UC application components
10	Refinement of the UC application based on the Kubernetes capabilities
11	Integration of the AC^3 LCM to execute the application through AC^3
12	Testing of the Edge-to-Cloud and Cloud-to-Edge application migration
13	Testing of the Edge-to-Cloud and Cloud-to-Edge application migration

5 Use Case 2 Integration Setup

UC2 integrates advanced technologies such as IoT, cameras, and UAVs to create a robust video surveillance and streaming system. This system supports both live streaming and video-on-demand (VoD) functionalities. Cameras and IoT devices are deployed both on the ground and onboard UAVs to collect various types of sensing data, such as carbon monoxide (CO), carbon dioxide (CO_2), and passive infrared sensors (PIR), enhancing the capability of video surveillance systems to cover areas without blind spots. This integration facilitates the dynamic deployment and management of IoT-driven services within the CECC framework, optimizing data transmission and processing across the cloud-edge continuum.

5.1 Challenges

5.1.1 Use Case Challenges

UC2 confronts several challenges stemming from the heterogeneity of computational power and hardware capabilities among the deployed cameras, IoT devices, and UAVs. Some devices are equipped to perform advanced on-board processing, while others are constrained to basic tasks, necessitating reliance on cloud or edge computing resources for more intensive data processing tasks. Key challenges include:

- **Computational and Hardware Heterogeneity**: Managing diverse processing capabilities and ensuring efficient operation across devices.
- **Efficient Data Gathering**: Collecting and processing data from IoT sensors and video content in real-time, especially challenging due to the resource constraints on some IoT devices.
- **Network Connectivity**: Ensuring robust and high-speed network connections for IoT devices and cameras, crucial for supporting live streaming and VoD in far-edge locations.
- **Data Management**: Handling large volumes of video and sensor data, necessitating significant resource management across distributed devices.
- **Customization and Flexibility**: Designing a system that allows users to specify their operational needs through a flexible, microservice-based platform, accommodating changes in user requirements and system configuration without extensive redevelopment.

5.1.2 Integration Challenges

The integration of UC2 within the EUR testbed introduces specific technical challenges that need to be addressed to ensure the successful deployment and operation of the system:

- **System Integration and Testing**: Ensuring that all components of the UC2 architecture, from UAVs and IoT devices to deep learning systems like Deepstream and YOLO, are seamlessly integrated and function cohesively.
- **Scalability and Resilience**: Developing the infrastructure to be scalable and resilient, allowing for the dynamic migration of microservices across devices and platforms as needed to maintain continuity and performance.
- **Operational Flexibility**: Implementing a system that can adapt its functionalities in real-time to changing conditions and requirements, such as shifting from area monitoring to movement detection or other surveillance tasks as dictated by user needs.

- **Resource Optimization**: Leveraging AI-based orchestration within the CECCM to optimize the allocation and utilization of computing resources across cloud and edge nodes, reducing latency and enhancing system responsiveness.

5.2 AC^3 Objectives to be Demonstrated

In what follows we present the objectives of the UC2 and the related KPIs:

- **Intuitive Application Definition:** The application developer can efficiently define and deploy microservice applications within the CECCM framework using an intuitive GUI and OSR. This approach simplifies the user experience, enabling easy access to advanced technology such as UAVs, far-edge devices, and AI/ML capabilities, streamlining the configuration and deployment processes.
- **Automatic Deployment and Zero-Touch Management:** Demonstrated in UC2, the robust LCM capabilities within AC^3 facilitate automated deployment, monitoring, and maintenance of microservices. This system incorporates AI-driven zero-touch management algorithms that autonomously optimize and maintain application performance. These algorithms leverage machine learning for predictive maintenance and resource allocation, significantly reducing the need for human intervention. The details of these processes are elaborated in WP3 and WP4.
- **Microservice Deployment and Migration:** The resilience of the CECCM is showcased by its capability to manage microservices across far-edge environments, including UAVs. In scenarios where a UAV becomes unavailable or encounters resource limitations, the system seamlessly migrates services to alternate UAVs or edge infrastructures, ensuring consistent service delivery and system continuity.
- **Data Analysis and Decision-Making for Enhanced Urban Security:** Utilizing AI and ML technologies like Deepstream and YOLO, the system provides real-time analysis capabilities for object detection, movement tracking, and human activity recognition. This functionality enhances urban security by delivering actionable insights and supporting rapid decision-making.
- **Environmental Monitoring and Response:** Equipped with IoT sensors for monitoring variables such as temperature and CO_2 levels, UAVs within the system enable real-time environmental monitoring. This capability allows for proactive responses to dynamic environmental conditions, enhancing urban management and public safety.

KPIs:

- **Traffic Reduction:** By treating data locally on far-edge devices, the system reduces the load on the CECC infrastructure by at least 50%, minimising the need for data transmission to the central cloud.
- **High Availability:** The system ensures a high reliability level with an uptime greater than 99.9%, leveraging the CECCM's zero-touch management and predictive availability features.
- **Latency Optimization:** Using distributed microservices for AI/ML processes, such as object detection and tracking (e.g., YOLO), the system reduces latency to milliseconds. The duplication and distribution of these micro-services across edge and cloud domains improve response times for tracking, detection, prediction, and analysis tasks.

5.3 Use Case 2 Mapping to AC^3 Architecture

The architecture of UC2 showcases the integration of smart monitoring through UAVs, IoT devices, and video analytics leveraging AI and edge computing. The UC is centred around the following components: **1) User Interface, 2) video analytics using Deepstream and ML components, 3) CECCM, and, 4) IoT devices**.

1. **UC User Interface**: The UC user interface is designed to configure and manage edge devices, ensuring efficient data processing and video streaming. It allows UC users to monitor live streams, receive real-time alerts for unusual activities, and analyse telemetry data through labelled and segmented images. Administrators can add, edit, activate, or deactivate various edge devices via this interface, which interacts with the system's frontend, backend, and video analytics microservices. Functionalities include:
    a. Monitoring live video streams for detecting security threats or unusual activities.
    b. Receiving real-time alerts (e.g., unusual vehicle behaviour).
    c. Viewing labelled and segmented images for telemetric analysis.
2. **Far Edge/IoT Devices (Raspberry Pi, Nvidia Jetson)**: The architecture includes UAVs equipped with Nvidia Jetson and Raspberry Pi devices that serve as edge nodes. The Nvidia Jetson is responsible for on-device AI/ML processing and real-time video analysis, while the Raspberry Pi manages video streaming but lacks video processing capability. These devices support various sensing functions, including CO2 monitoring, temperature, and object detection.
3. **Video Analytics System**: Using Deepstream and YOLO, the video analytics system processes streams captured by UAVs, performing object detection, human activity surveillance, and more. The system is built for real-time insights and can support both live streaming and VoD.
4. **CECCM**: The CECCM offers functionalities to optimise performance, reliability, and flexibility in the UC2 architecture. It uses AI-based orchestration to manage computing resources between cloud and edge nodes. With service relocation and offloading capabilities, CECCM ensures high resiliency and efficient management of the surveillance infrastructure. It enables seamless integration of cloud-edge technologies, reducing latency, optimising bandwidth, and improving the system's overall performance.

The UC2 architecture is structured around a microservices framework that ensures scalability and adaptability to different operational environments. This framework includes **frontend, backend, and deepstream services** deployed across various devices, such as Jetson units and x86 servers.

The Central Server, integral to the architecture, manages user interactions through a frontend API and handles CRUD operations for device and user management, with data stored in an SQL database. The Query Processing System, embedded within the microservices framework, supports dynamic query management and service deployment, enabling real-time video streaming, VoD, and effective data collection. The Authorization and Authentication Service ensures secure access control, allowing only authorized users to manage system functionalities. Additionally, the Regional Server, a component of the deepstream microservice, addresses the local processing needs of IoT devices and cameras in designated regions and supports federated learning initiatives.

Each component is precisely integrated to meet the functional demands of the system, providing robust support for video surveillance and environmental monitoring while adapting to various operational requirements.

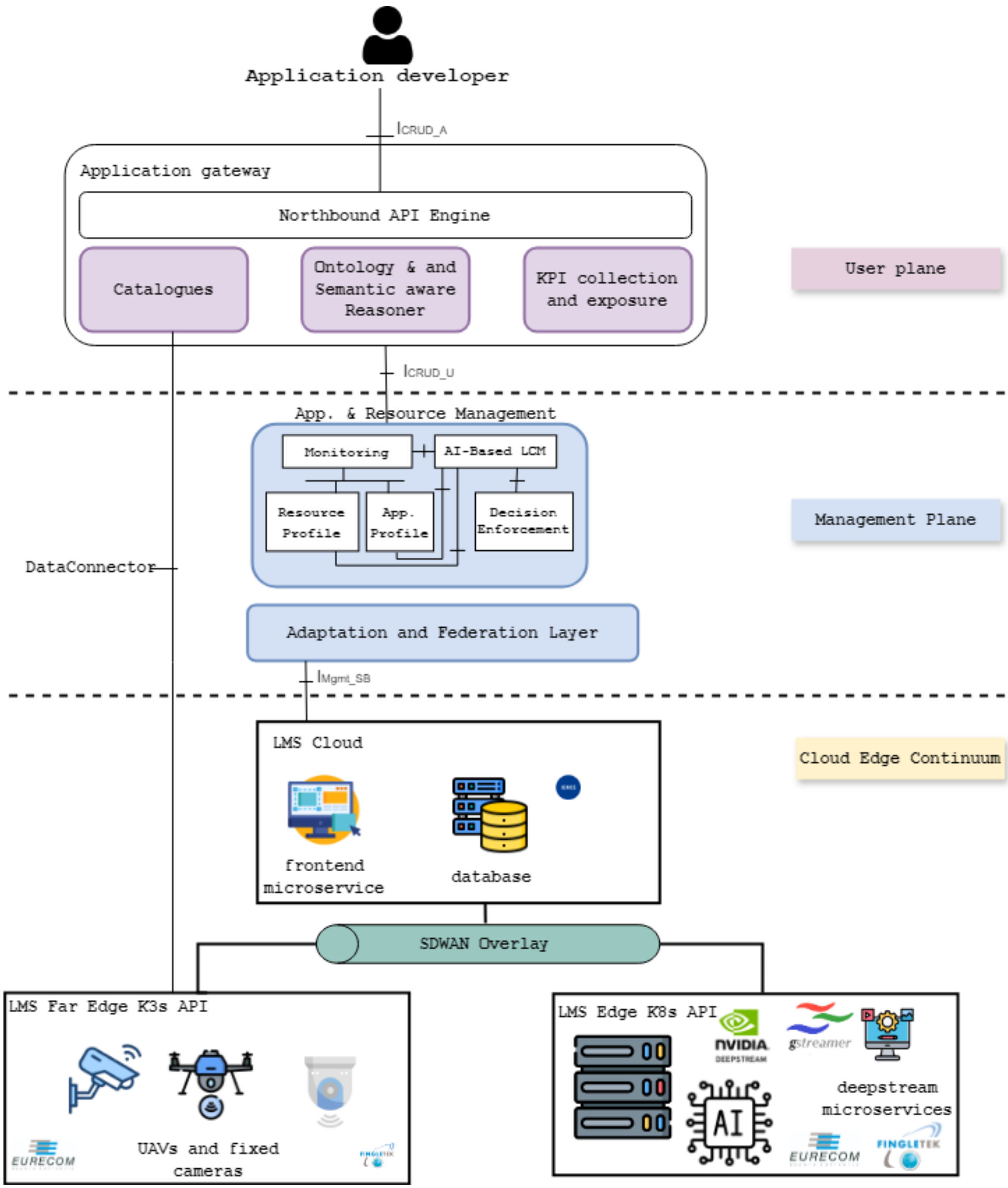

Figure 5. UC2 architecture following the AC³ general architecture paradigm

The UC2 architecture is illustrated in Figure 5. It integrates various components of the AC³ project including the GUI, OSR, LCM, and LMS. The figure provides a clear depiction of how the application developer interacts with the CECCM and the flow of application descriptors through the system for deployment and management.

The application developer uses the GUI as the primary interface to create and manage application descriptors. The GUI simplifies interaction with the system, allowing the developer to define the microservices, dependencies, and policies required for the application. Through a streamlined multi-step form or file upload, the GUI enables developers to manage the entire lifecycle of the application descriptor.

Once the descriptor is submitted via the GUI, it is transmitted to the OSR. The OSR interfaces with the Catalogue to support users in identifying relevant service blueprints. Specifically, the OSR queries the Service Catalogue to retrieve descriptors that align with user-defined application components, policies, and runtime requirements.

After retrieving the necessary descriptors from the catalogues, the OSR composes a YAML application descriptor that integrates all the application requirements. This completed YAML file is then sent to the LCM, which manages the orchestration and deployment of the application across the cloud-edge infrastructure.

The LCM then takes over, orchestrating the deployment of the microservices based on the translated application descriptor. The LCM ensures that each microservice is deployed to the appropriate level of the cloud-edge continuum—whether on the LMS Cloud, LMS Edge, or LMS Far Edge. The LCM also manages the lifecycle of the deployed services, handling tasks like scaling, migration, and fault tolerance. This capability ensures service continuity even in cases of node failure or dynamic changes in the system.

The LMS, such as Kubernetes or K3S, deployed at the Cloud, Edge, and Far Edge levels, manage the physical and virtual resources needed for the microservices. They handle container orchestration, resource allocation, and ensure that services run optimally across distributed environments.

Table 4 presents the contributions of different functional components to UC2, including details on the architecture components, their sub-components, descriptions, and the technological tools or partners involved.

Table 4. Contribution of Functional Components to UC2

Architecture component	Sub-Component	Description	Technological tool/Partner
Application gateway (GUI)		Allow the application developer to define its application components and SLA.	GUI (FIN)
OSR		Allow the generation of the Application Descriptor	OSR (FIN)
LMS Edge		Will manage the micro-services that cannot run at the far edge (due to low computing resources) and centralised cloud for low latency or bandwidth optimization.	Kubernetes API (EUR)
LMS Far Edge		Will manage the micro-services that will run on the far edge	K3S API (EUR)

		device (i.e., UAV). Video capture and object detection microservices.	
LMS Cloud		Will manage and run the front-end micro-service.	ION Cloud
LMS Networking		Will interconnect the clusters (K3S, Kubernetes and ION Cloud).	SD-WAN controller (EUR)
Application and resource management	Monitoring	Monitoring the micro-services KPI	Monitoring tools (EUR)
	AI-based LCM and Decision Enforcement	1. Manage the micro-services Life Cycle 2. Migration algorithm that adapts if the far-edge resource degrades or moves to the far-edge a micro-service	1. LISO NFVO (EUR) 2. Proactive Stateful Container Migration Based on Resource Utilization in WP3 (EUR)
	Zero-touch configuration and application management (predict drones' availability)	Predict and describe infrastructure resources and implement automated corrective measures.	Using machine learning and deep learning approaches initiated in D4.1 (Section 5.3.2 and 5.3.3)
	AI-Based application profile	Predicting Application Behaviour	Algorithm on app behaviour prediction from WP3 (FIN)
	AI-Based resource profile	Describe the resources of the infrastructure	Resource exposer (EUR)

5.4 Experimentation Platform Deployment for Use Case 2

The Edge facility at EURECOM that is used for the UC2 deployment, illustrated in Figure 6, provides a robust platform for deploying cloud-native applications and microservices. This includes edge applications, Virtual

Network Functions (VNFs), and Container Network Functions (CNFs). The platform incorporates management and orchestration components that adhere to ETSI standards for Network Virtualization [7] and Multi-access Edge Computing [8].

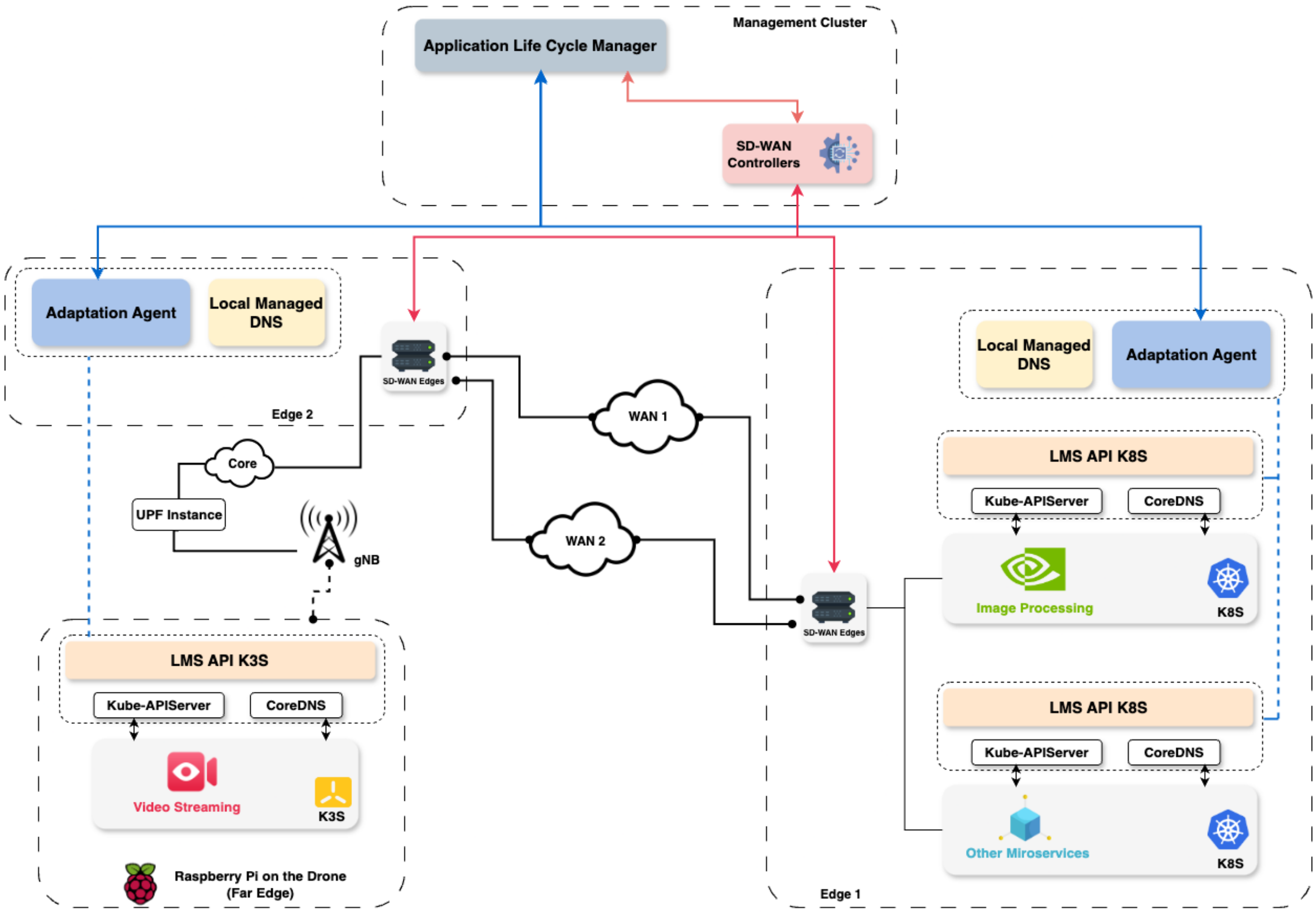

Figure 6. Experimentation platform for UC2 using EURECOM's testbed

The orchestrator, known as Lightweight Edge Slice Orchestration (LISO), oversees the lifecycle of services using Network Service Descriptors (NSDs). These NSDs comprise a set of cloud-native applications, microservices, or functions (i.e., CNFs) described by an Application Descriptor (AppD). These cloud-native functions or microservices are deployed across various container management platforms or LMSs, such as OpenShift, Kubernetes, and K3S.

The interconnection of these clusters is achieved through Software-Defined Wide Area Network (SD-WAN), with each edge site hosting an SD-WAN edge node at its boundary. By creating overlay links that abstract the heterogeneous WAN infrastructure, SD-WAN edges connect each edge site's local network as if they were part of the same network.

These overlay topologies, along with other networking capabilities such as microservices' traffic redirection rules, are created and managed by the network LMS or the SD-WAN controller. By interacting with the Application Life Cycle Manager (LISO), the SD-WAN controller enables the interconnection of separated application's microservices while ensuring that SLAs are met during traffic routing.

These SLA or Quality of Service (QoS) requirements that an application is demanding, are defined by the LISO at the NSDs level, defining a microservice-to-microservice connection needs.

In terms of infrastructure, we have an edge cluster powered by an Intel-based machine and managed by Kubernetes. Additionally, there are far-edge clusters composed of single-board computers, including a Raspberry Pi and two Nvidia Jetson devices. These far-edge clusters can be mounted on a drone and are managed using K3s, a lightweight version of Kubernetes optimised for edge devices.

The Raspberry Pi onboard the UAV connects to the 5G network through a dedicated 5G modem, allowing it to maintain high-speed, low-latency communication with other nodes in the system. This connection enables the real-time transmission of data from the UAV to the central infrastructure, supporting rapid decision-making for applications like object detection and movement tracking.

Moreover, the Raspberry Pi shares an SD-WAN edge with Kubernetes (K8s) to ensure secure and reliable connectivity across the cloud-edge continuum. The integration with SD-WAN provides network segmentation, traffic prioritisation, and optimised routing, which are essential for handling the high demands of real-time video processing and microservice management across distributed nodes. By leveraging SD-WAN alongside Kubernetes, the system ensures efficient resource utilisation and seamless service continuity across both edge and far-edge clusters. This setup enhances the robustness of the UAV infrastructure, enabling reliable, scalable, and flexible management of microservices in a dynamic environment.

This setup ensures a flexible and scalable architecture capable of supporting a wide range of cloud-native applications and microservices at the edge.

5.5 Experimental Scenarios

In the AC^3 UC Demonstration, a smart surveillance system is deployed in an urban parking loT, utilising UAVs, IoT devices, and advanced AI-driven analytics. The primary components of the system include drones, fixed cameras, Raspberry Pi/Nvidia Jetson devices for edge processing, environmental sensors, and a centralised dashboard for comprehensive monitoring and analysis. This setup enables real-time surveillance, data processing, and resource management by interacting with the CECCM.

Experiment UC2.1: Zero-Touch Configuration and Application Management: Through the CECCM, developers can use zero-touch management features, including predictive algorithms to monitor and anticipate drone availability. The CECCM's intuitive GUI and the OSR provide a platform for flexible application management, facilitating seamless configuration, deployment, and reconfiguration of microservices to adapt to changes in the environment.

Experiment UC2.2: Application Descriptor Composition and OSR Integration: The GUI of the Application Descriptor Composer serves as the main interface for creating and managing application descriptors, allowing developers to create, read, update, and delete descriptor models. Once the descriptor is created, it is transmitted to the OSR, which interprets it using semantic web technologies (e.g., ontologies and reasoners). This step ensures that all application components, policies, and requirements are properly represented. By connecting with the POSSIBLE-X catalogue, the OSR can access data and service catalogues to identify and retrieve blueprints that meet the application's specified needs. This integration results in a comprehensive YAML file, which aligns with application requirements and is passed to the LCM for deployment across the cloud-edge infrastructure.

Experiment UC2.3: Far Edge-Tailored Platform for Microservices: The AC^3 framework leverages a virtualized platform at the far edge, allowing micro-services to run on UAVs and edge devices, such as Raspberry Pi and Nvidia Jetson. This platform is optimised for handling computationally intensive tasks locally, thereby reducing the load on the CECC infrastructure by minimising the data sent to central cloud systems.

Experiment UC2.4: Seamless Microservice Deployment and Migration: The LCM component orchestrates the deployment of microservices across the cloud-edge continuum. Through integration with Zero-Touch Service Management (ZSM), microservice deployment, scaling, and migration are automated within the AC^3 framework. The LCM ensures that, in cases of resource degradation or unavailability, micro-services can seamlessly migrate across UAVs or edge infrastructures. This functionality guarantees high availability and uninterrupted service, making real-time adjustments based on environmental conditions.

Experiment UC2.5: Data Management and Processing for Traffic Reduction: The system employs local data processing techniques, enabling AI-driven object detection and tracking directly on far-edge devices. This local treatment of data reduces the volume of content transmitted to the CECC infrastructure, optimising bandwidth usage and minimising latency.

The UC2 deployment flow will be illustrated through a series of deployment steps that showcase the functionalities and capabilities of the AC^3 framework and its components in the context of UC2:

1. Initial Setup and Area Scanning

Drones equipped with high-definition cameras actively patrol the parking lot during peak hours, capturing detailed, continuous video streams of vehicles moving in and out of the premises. These aerial units are complemented by strategically placed fixed cameras that monitor the entry and exit points, ensuring full coverage. The video feeds are processed in real-time by edge devices, such as Nvidia Jetson modules, which run specialised ML models. These models are trained to identify vehicles based on make, model, and colour, eliminating the need to transmit raw video data to cloud-based systems. This local video processing drastically reduces bandwidth usage and latency, allowing for near-instantaneous identification and tracking of vehicles. The system ensures that all critical events and vehicle details are processed on-site, maintaining a swift and efficient surveillance operation.

2. Environmental Monitoring

In addition to video surveillance, drones and fixed cameras are equipped with environmental sensors that capture data on temperature, humidity, wind speed, and CO_2 levels. This real-time environmental data is streamed directly to the system's central dashboard, where it is integrated with live video feeds. The unified display provides operators with both vehicle activity and environmental conditions, enabling a holistic overview of the parking lot. For example, if environmental thresholds are crossed (e.g., a sudden spike in CO_2), alerts can be triggered to city officials or maintenance crews to respond promptly, ensuring not only safety but also compliance with environmental regulations.

3. Data Processing & Insights Generation

The system's data processing capabilities go beyond simple monitoring. By analysing video streams and sensor data, the system can generate insightful reports on vehicle congestion, parking availability, and environmental conditions. For instance, it tracks the volume of vehicles entering and exiting the lot in real-time, which allows for the detection of potential congestion points. The system can issue automated alerts when the lot nears full capacity, enabling dynamic re-routing of incoming vehicles to nearby parking facilities. Additionally, by leveraging high-resolution still images and AI-driven image labelling, the system assists security personnel in identifying unusual or suspicious activities, such as unauthorized parking or vehicles behaving erratically. This enables a proactive approach to traffic and parking management, significantly improving efficiency and safety in urban spaces.

4. Real-Time Dashboard Monitoring

At the heart of the system is a sophisticated, centralized dashboard that displays all incoming video streams, environmental data, and incident alerts in real-time. The dashboard offers a range of features, including live video feed switching, environmental data visualization, and an interactive datetime picker that allows operators to review historical footage and sensor data from specific timeframes. This interface empowers the monitoring team to not only oversee current events but also track long-term trends in vehicle activity, parking congestion, and environmental conditions. For example, they can easily access historical data to analyse peak traffic times and optimize parking lot usage accordingly.

5. Post-Survey Reporting

At the close of each day, the system compiles a comprehensive report for city administrators. This report includes a detailed summary of vehicle traffic patterns, such as the total number of vehicles entering and exiting the lot, congestion peaks, and average parking duration. It also features an analysis of environmental trends, highlighting any significant changes in conditions like air quality or temperature. Additionally, the report includes flagged security incidents, complete with timestamps and high-resolution images, providing administrators with the data needed to make informed decisions on parking operations, traffic management, and environmental impact. These reports play a vital role in enhancing urban sustainability and improving the overall efficiency of parking infrastructure.

5.6 Milestones

The key milestones, illustrated in Table 5, outlines the development phases of UC2, providing a structured overview of the project's progress and objectives for implementing the Smart Monitoring System.

Table 5. Integration Plan Milestones for Use Case 2

S/N	Implementation
1	UC architecture

2	Development of YOLO for object detection and identification. Demonstration of deep stream and IoT devices
3	Development of UC dashboard
4	Integration of #2 and #3
5	Containerization of frontend, backend, and Deepstream
6	Demonstration of various containerization scenarios to capture the vision of AC^3
7	Kubernetization scenarios
8	Live demonstration with drones
9	Integration within AC^3 project (demonstrate the CECM's capabilities to deploy and run micro-services on top of the far edge (e.g., UAV) and anticipate drone unavailability by migrating the micro-service from one drone to another or the infrastructure edge.)
10	Further testing and validations (Reduce the load generated by produced content on the CECC infrastructure by locally treating data and guarantee high availability of the application)

6 Use Case 3 Integration Setup

Understanding the evolution of galaxies over cosmic time is a fundamental question in astronomy. To address it, we need a comprehensive understanding of the processes that drive their formation and play a role in the changes they undergo throughout their existence. Nearby galaxies provide a unique opportunity to gain insight into these processes.

6.1 Challenges

6.1.1 Use Case Challenges

UC3 presents a unique set of challenges related to the processing and analysis of extremely large astronomical datasets. These challenges stem from the sheer volume of data, the complexity of the analysis required, and the need for efficient and scalable computing resources. Addressing these challenges is crucial for extracting meaningful scientific insights from this data and maximizing the potential of astronomical research. Key challenges include:

- **Handling Large and Complex Data Cubes:** Managing the substantial data volumes generated by advanced instruments like MEGARA, which require sophisticated methods to derive actionable insights.
- **Efficient Management of Data Pipelines:** Orchestrating the influx of multiple data streams to ensure efficient processing and continuity across the system.
- **Ensuring Robust System Availability:** Guaranteeing uninterrupted data processing with a target of 100% reliability to meet the stringent requirements of astronomical research.
- **Accelerating Data Processing:** Achieving at least a 50% reduction in processing time for 5GB data cubes compared to standalone nodes through system optimization and efficient task handling.
- **Spectral analysis**: For spectral analysis, tools like pPXF, STECKMAP, and STARLIGHT are leveraged. Containerization, persistent volumes, and shared storage enable precise and scalable handling of large datasets, advancing research into galaxy evolution.

6.1.2 Integration Challenges

This section highlights the key challenges faced in managing and analyzing large-scale 3D data cubes, particularly in fields like astronomy where the data is both vast and highly complex. As these datasets grow in scale and complexity, it becomes essential to leverage advanced computational techniques and scalable infrastructure to extract meaningful scientific insights.

- **Seamless Integration of Edge and Cloud Infrastructure:** Breaking down data cubes into spaxels for efficient processing on edge infrastructure, leveraging the CECCM to unify cloud and edge operations across a federated infrastructure of OpenShift Clusters, unified by the AC^3 network operator.

- **Optimizing Edge Resource Utilization:** Identifying the most suitable edge nodes for analytics tasks based on their performance and availability, ensuring optimal use of resources.

- **Dynamic Task Distribution and Load Balancing:** Using the CECCM to dynamically allocate tasks across cloud, edge, and far-edge environments, adapting to changing demands.

- **Zero-touch deployment and scalability**: Leverage the AC3 AI-based monitoring, LCM and network operator to automate deployment, migration and scaling across the federated infrastructure, while Skupper ensures seamless and secure communication between cloud and edge resources.

Our goal is to use Integral Field Spectroscopy data (also known as data cubes) for a large sample of these galaxies, taking advantage of existing data from advanced facilities such as MEGARA Tat the 10.4 m Gran Telescopio de Canarias, MUSE installed in the Very Large Telescope (VLT) and MaNGA at the 2.5 m Sloan Telescope. In turn, this requires the application of Full-Spectrum Fitting techniques to extract essential parameters such as the age and metallicity of the stellar populations that form these galaxies as well as the kinematical state of these stars. Processing and analysing these massive data cubes present substantial challenges, demanding advanced computational resources and sophisticated data management strategies. The UC3 testbed is designed to address these needs by providing scalable and distributed processing capabilities that can efficiently handle large-scale datasets. This infrastructure enables astronomers to extract valuable insights from these data cubes, including details about stellar kinematics and population characteristics. As traditional systems often struggle with such large volumes of data, leveraging the UC3 testbed paves the way for more efficient analysis, supporting the rapid exploration of galaxy evolution and setting a foundation for future discoveries with data from observatories like the James Webb Space Telescope (JWST).

A typical application of UC3 is illustrated in the composite image of the nearby galaxy UGC 10205 in Figure 7. Overlaid on the image is the footprint of the MEGARA field of view (red rectangle). The image' bottom panel illustrates the distribution of the continuum emission of UGC 10205 derived from the MEGARA data cube whereas its top right panel provides the original spectrum for one cell of the MEGARA data cube shown in black while its best fit using "Full-Spectrum Fitting" techniques appears in red.

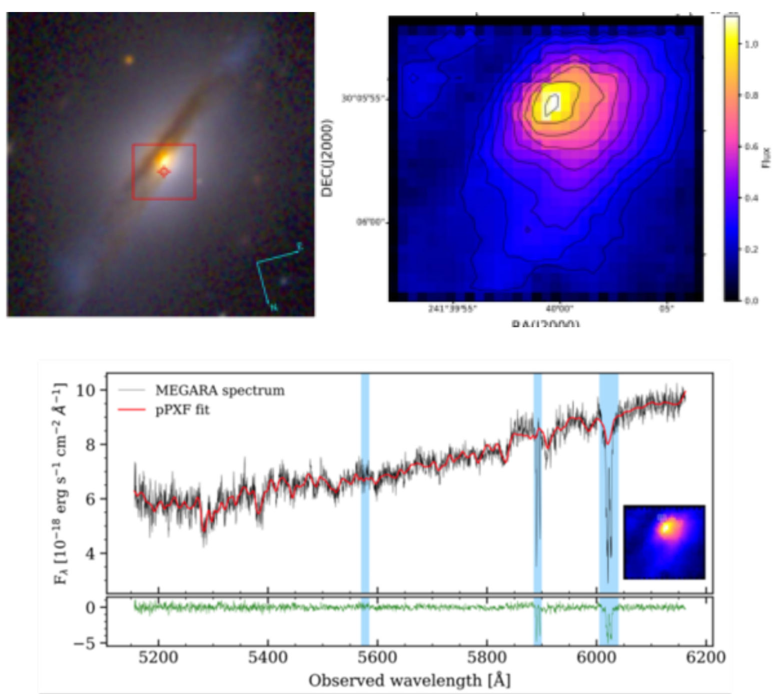

Figure 7. UC3 Astronomy Data

6.2 AC^3 Objectives to Be Demonstrated

We will demonstrate the CECCM's capabilities to deploy and run astronomical software to potentially process hundreds of TBs of data cubes. This will allow us to integrate scientific applications that will take advantage of hybrid cloud-native infrastructures to optimize the computation process based on smart AI algorithms. The implementation of this UC also enables the whole astronomy community, scientific and research teams to accelerate the analysis of the novel data gathered from newer and additional instruments and data sources, such as JWST.

We will achieve this by:

- **Efficient Data processing**: The UC aims to significantly enhance the processing speed and accuracy of 3D data cubes, which are key in analysing galaxies. These data cubes, which contain detailed spectral information, will be processed faster by employing advanced computational techniques via the CECCM infrastructure. Specifically, the goal is to reduce the processing time for a 5GB data cube by at least 50% compared to standalone systems. The application will also leverage OSR (Open Source Research) components, incorporating the AC3 (Adaptive Computing and Cloud) framework for data management. Key elements of this framework include the data collector and broker, which will efficiently manage data flow and improve the scalability of the system.
- **Improved data analysis tools**: Enhance and optimise the use of spectral tools such as pPXF, STECKMAP and STARLIGHT to extract key parameters such as stellar kinematics, the age and the metallicity of the stars that form these galaxies from 3D data cubes. These tools will work with the CECCM platform to improve the accuracy and depth of galaxy analysis, unlocking new insights from existing and future astronomical datasets. The application will be described using the GUI, allowing for intuitive interactions with the platform and enhancing the user experience for researchers. This approach will streamline data processing and analysis, making the platform accessible to a wider range of scientific users.
- **Optimal Resource management**: Ensure fine-tuned use of memory, CPU and storage through techniques like parallel processing, data compression and container orchestration, minimising resource consumption while maintaining performance.
- **Scalable infrastructure**: Develop and implement a distributed, scalable system using containerized software tools and high-performance computing to handle the increasing volume and complexity of data cubes efficiently. The system will be dynamically deployed using OSR, ensuring that the infrastructure is flexible and scalable to meet the evolving needs of the research community. This scalability will be critical for processing ever-larger datasets from next-generation instruments like the JWST.
- **Benchmarking Future Research**: This UC will establish a local benchmark for future studies on galaxy evolution across various periods in cosmic history. By processing data from the JWST and other future instruments, this platform will provide the foundation for rapid, large-scale analysis, accelerating discoveries and revolutionising the field.

UC3 which is, led by UCM and RHT, will demonstrate how AC^3's cognitive cloud capabilities enable revolutionary astronomy discoveries through its intelligent management of applications and resources. The AC^3 system can support the UC3 application in processing huge amounts of data, by intelligently scaling or migrating data processing pods in accordance with observed workload requirements, while at the same time ensuring there are sufficient resources (e.g. network links between clusters) to support the adaptation of the application. This unleashes the full potential of modern telescopes and scientific instruments while reducing processing times and resource consumption.

KPIs:

- **Data Processing Efficiency:** Reduces the time needed to process a 5GB data cube by at least 50% compared to standalone systems.
- **High Availability:** Ensures 100% reliability and uptime, supporting continuous processing and analysis of critical astronomy data.

These KPIs underscore CECCM's role in making astronomy data processing scalable, reliable, and resource-efficient, setting a strong foundation for handling future data from observatories and astronomical research.

6.3 Use Case 3 Mapping to AC^3 Architecture

The UC3 implementation will make use of a set of containerized specific-use analysis tools to process the data that is distributed by the CECC manager. The tools that will be integrated are different spectral synthesis software assets whose main goal is to decompose an observed spectrum in terms of a combination of templates of Single Stellar Population (SSP) of various stellar ages and metallicities considering velocity broadening and instrumental effects. This method produces interesting outputs. Mainly, the information about the stellar populations and the stellar kinematics. The former one allows to characterize the metallicity of the stars that compose the galaxy being an indication of the abundance of elements heavier than helium. On the other hand, the kinematic information is contained in the following parameters: (i) the line-of-sight velocity, which indicates the motion of the galaxy either toward or away from the observer; (ii) the velocity dispersion that is related to the broadening of the spectral lines and it provides information about the internal motions of stars within the galaxy, and, (iii) the high-order moments 'h3' or skewness that characterizes the asymmetry of the velocity distribution and 'h4' or kurtosis that provides information about the sharpness of the distribution's peak.

The main software assets that will be used within the AC^3 project to perform this task are the following:

- **STECKMAP** (STEllar Content and Kinematics via Maximum A Posteriori), a method to recover the kinematical properties of a galaxy simultaneously with its stellar content from integrated light spectra. The code employs a maximum a posteriori approach, which combines the likelihood of the model fit with prior knowledge or assumptions about the expected properties of the stellar populations and kinematics to obtain more robust solutions.
- **STARLIGHT** uses an optimization algorithm to adjust the weights applied to each spectral template to minimize the difference between the modelled spectrum and the observed spectrum. Common optimization algorithms include techniques based on χ^2 minimization or maximum likelihood estimation.
- **pPXF** (penalized PiXel-Fitting method) is code developed in Python to extract the stellar kinematics and stellar population from absorption-line spectra of galaxies, using a maximum penalized likelihood approach.

Figure 8 describes the overall architecture of UC3, summarizing the main architectural components and how they interact. The astronomy software is modelled as a set of interconnected microservices (Orchestrator and Processor) based in Linux containers. The users interact with the CECCM to deploy the application, where the CECCM manages the scheduling of the application, as well as ensuring availability of the required resources (data-sources, networking, storage, computing). The execution of the astronomy software can be carried out in different regions, depending on the overall system load and the requirements. During the execution of the application, both system and application metrics are monitored, and fed into the application and resource profiles. The CECCM can then use this data to reconfigure the application placement and/or scaling, as well as the supporting resources (e.g. networking) on the fly, via the appropriate LMS North Bound Interfaces. The application data is provisioned by leveraging the AC^3 data management framework, and once the software finishes its execution, the results are returned to the orchestrator and stored in the provisioned storage resources.

The architecture for UC3, illustrated in Figure 8, relies on a microservices-based design, deployed within containerized environments to ensure modularity, scalability, and flexibility. The system manages the astronomy software in a way that supports dynamic resource allocation and responsive scaling, with key components described as follows:

Orchestrator and Processor Microservices: The astronomy software itself is composed of multiple microservices, primarily the Orchestrator and Processor, which coordinate the execution and processing of astronomical data. Deployed as Linux containers, these microservices allow for scalable and efficient workload management, enabling a modular design that simplifies updates, scaling, and maintenance of individual components.

CECCM: Serving as the main interface for users, the CECCM allows administrators and scientists to deploy and monitor the astronomy application. This central management platform oversees critical functions, including application scheduling and resource provisioning (data sources, networking, storage, and compute resources) to meet the software's operational needs. CECCM also dynamically allocates resources across different regions, optimizing placement based on system load and application requirements for balanced performance across distributed environments. Additionally, it provides tools for monitoring and managing the lifecycle of the microservices, including scaling decisions and load balancing across clusters.

Metrics Monitoring and Resource Profiling: During application execution, a real-time monitoring system collects both system and application metrics. This monitoring enables CECCM to build and update detailed resource and application profiles, which contain essential data on system performance and resource utilization. By leveraging this data, CECCM can adjust application scaling and reconfigure resource allocation to ensure optimized execution, even under varying workload demands.

LMS North Bound Interfaces: To facilitate these adjustments, CECCM uses the LMS North Bound Interfaces, allowing it to modify application configurations and resource provisioning on demand. This integration supports automated scaling and adaptive resource management, which is needed for maintaining performance in highly variable conditions. The interfaces allow CECCM to seamlessly interact with underlying infrastructure, triggering necessary changes without manual intervention. The North Bound Interfaces also enable coordination with network management services to maintain connectivity between microservices across distributed clusters.

AC^3 Data Management Framework: Application data is managed through the AC^3 data management framework, which facilitates efficient data provisioning to the astronomy software. This framework ensures that the required data is available throughout execution and supports robust data flow across regions. Once the application completes its processing tasks, the results are gathered by the orchestrator and stored within dedicated storage resources, ensuring data persistence and availability for subsequent analysis or retrieval.

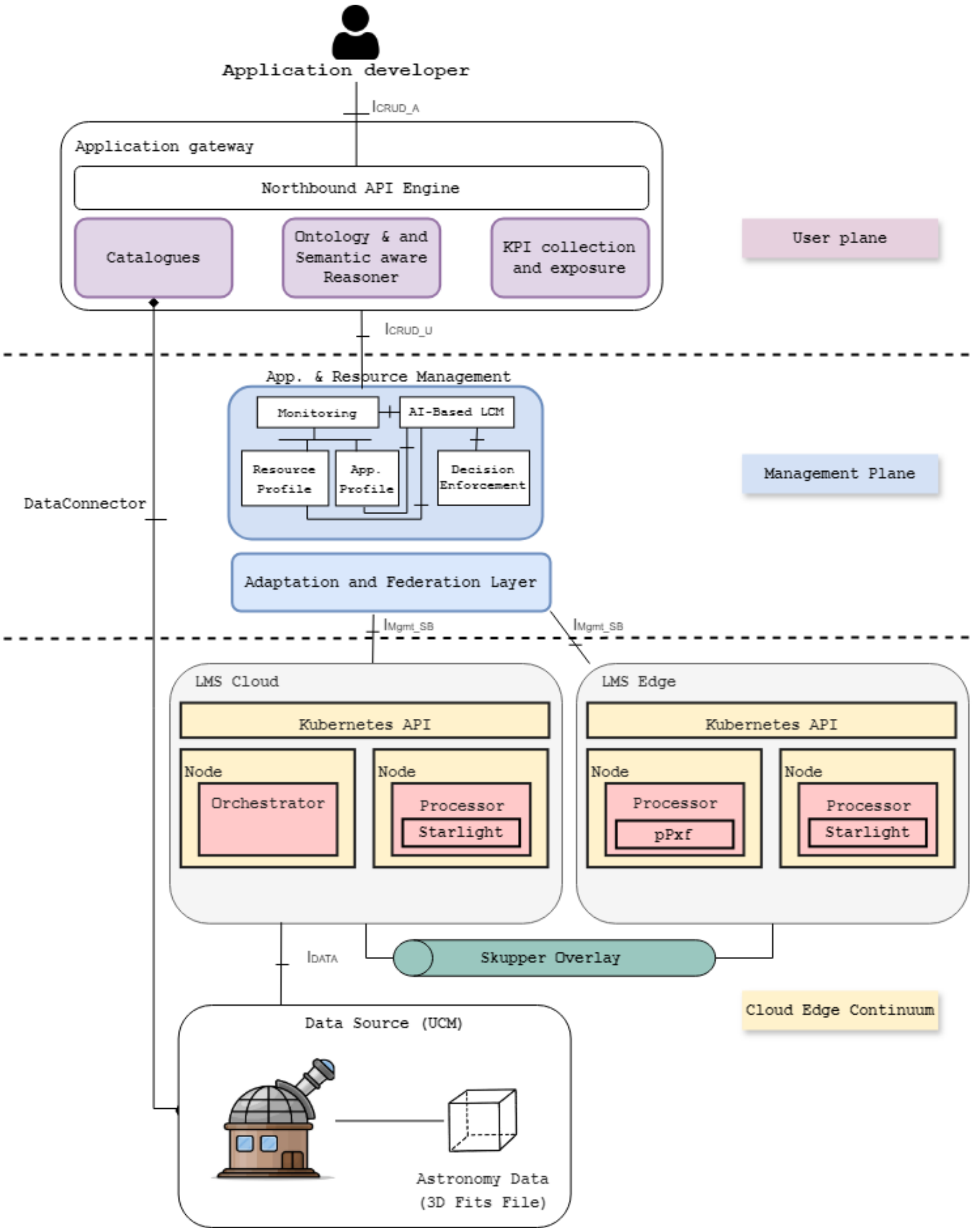

Figure 8. The architecture of UC3 following the AC^3 general architecture paradigm

Table 6 outlines the roles of various functional components in UC3, providing information on the architecture elements, their sub-components, descriptions, and the technological tools or partners associated with them.

Table 6. Contribution of Functional Components to Use Case 3

Architecture component	Sub-Component	Description	Technological tool/Partner

Application gateway (GUI)		Allow the application developer to define its application components and SLA.	GUI (FIN)
OSR		Allow the generation of the Application Descriptor	OSR (FIN)
Network LMS		Connects the OpenShift clusters running on the ARS cloud	AC³ Network Operator - Kubernetes (RHT)
Application and resource management	Monitoring	Monitors application and resource metrics	Monitoring tools (EUR)
	AI-based LCM and Decision Enforcement	1. Manage the micro-services Life Cycle 2. Horizontal autoscaling	1. Service Orchestration tool over Kubernetes (EUR, UBI) 2. Autoscaling approaches such as the Reinforcement Learning mechanism (IBM)
		Zero Touch Configuration: Predict and describe infrastructure resources and implementation of automated corrective measures.	Using machine learning and deep learning approaches initiated in D4.1 (Section 5.3.2 and 5.3.3) LCM Decisions for resource management (IQU)
	AI-Based application profile	Predicting application behavior	Algorithm on app behaviour prediction from WP3 (FIN)
	AI-Based Resource Profile	Describe the resources of the infrastructure	Resource exposer (EUR) Temporal Fusion Transformer (IBM)
Data management	1. Data Provider connector 2. Catalogue (data)	1. Astronomy data goes through the Data connector 2. Registers with the AC³ Catalogues (Piveau)	Piveau catalogue (SPA)

	3. Data Manipulator	3. Manipulate data for input to Orchestrator	
Migration		1. Manages dynamic microservice migration across cloud and edge. 2. Moves microservices between clusters based on real-time data. 3. Balances loads and avoids bottlenecks.	LCM, Kubernetes (RHT)

6.4 Experimentation Platform Deployment for Use Case 3

As referenced in Figure 8, the UC3 testbed is designed to address the challenges of processing and analysing large 3D data cubes in astronomy, which contain vast amounts of image and wavelength data. These datasets demand substantial computing resources, efficient data handling, and optimized resource management. Traditional on-premises systems are often inadequate when confronted with such data volumes. To tackle this, the UC3 testbed consists of multiple OpenShift clusters installed on 2XL servers provisioned on the ARS cloud infrastructure. OpenShift offers a reliable Kubernetes-based platform for orchestrating containers at scale and allows for flexible resource provisioning which is critical when processing terabytes of astronomy data. This testbed ensures high availability, scalability, and redundancy, enabling us to handle data-intensive workloads dynamically.

Applications within the UC3 testbeds leverage the distributed architecture by deploying processor pods across multiple OpenShift clusters. This multi-cluster deployment enables parallel processing of the astronomy data cubes, with each processor pod handling a subset of data slices. By distributing the processing load, the testbed ensures that data-intensive tasks are efficiently managed, reducing latency and optimising resource utilisation. This setup allows applications to dynamically scale processing power across clusters as needed, addressing the computational demands of analysing large astronomy datasets.

A critical aspect of the testbed is the integration of the AC^3 network operator which leverages Skupper to provide secure and transparent cross-cluster communication between multiple OpenShift clusters. This ensures that events and data can flow seamlessly between clusters, enabling distributed processing without the complexity of traditional networking configurations. The use of Skupper guarantees that the testbed remains resilient and scalable and facilitates the connection of multiple geographically dispersed clusters.

Finally, for data storage and management, data connectors copy the astronomy data into a persistent volume used by the Orchestrator pod, and the data is split into events. Once the events have been processed, the results are copied back into the Orchestrator's persistent volume which is leveraged by all processors. This ensures that processed data is centralized and readily available for analysis.

6.5 Experimental Scenarios

The workflow and deployment of the UC3 Starlight application are outlined as follows:

Zero-Touch Configuration and Data Management: CECCM manages data across the federated cloud and edge infrastructure, selecting the optimal nodes to run astronomy microservices. This minimizes manual intervention, highlighting zero-touch deployment and data location optimization for processing.

Federated Edge-Tailored Platform for Microservices: CECCM supports federated cloud and edge platforms, determining the most efficient nodes for running astronomy applications and allowing for seamless microservice duplication and migration.

Dynamic Microservice Migration: The UC3 Starlight application supports the dynamic migration of microservices between clusters to balance workloads, meet latency requirements, or address resource constraints. The LCM triggers migrations based on real-time performance data alongside AI based application and resource profiles, seamlessly redeploying microservices while maintaining network links and data accessibility.

Monitoring and Performance Tracking: The UC3 Starlight system is continuously monitored using Prometheus, which provides real-time insights into resource usage, queue length, application health, and system performance across clusters. This monitoring ensures that the system remains efficient and resilient, while tracking the impact of migrations and other adjustments.

Output Data Assembly and Finalization

Processed data is then reassembled into final output datasets, completing the data processing pipeline. The system is prepared for the next dataset, maximizing data throughput and reliability, a KPI for high availability.

The UC3 deployment flow will be illustrated through a series of deployment steps that showcase the functionalities and capabilities of the AC^3 framework and its components in the context of UC3:

1. **Application Deployment:**

The deployment of the UC3 Starlight application leverages CECCM's advanced federated cloud and edge infrastructure to optimise the execution of astronomy microservices. With a focus on zero-touch configuration, the system automatically manages the deployment and location of microservices across the cloud and edge nodes. CECCM dynamically selects the optimal computational resources for each microservice, minimizing manual intervention and ensuring that processing is done on the most efficient nodes. This approach guarantees a seamless deployment experience for researchers, allowing the UC3 Starlight application to be rapidly deployed and scaled without requiring manual adjustments, ensuring that the infrastructure adapts to the demands of large-scale astronomical data processing.

2. **Data Management and Processing:**

Efficient data management and processing are core to the UC3 Starlight application. Raw astronomical data is automatically moved to the most suitable nodes for processing, with techniques like microservice duplication and migration ensuring optimal data distribution. The application supports dynamic microservice migration, where services are relocated between clusters based on real-time performance metrics, such as resource availability or latency, enabling the system to scale efficiently with large datasets.

The system is continuously monitored to track performance, resource usage, and potential bottlenecks. Once processed, the data is reassembled into final datasets, ensuring timely delivery. The system optimises data accessibility while minimizing downtime, which is essential for ongoing astronomical research.

3. **Scalability and Efficiency:**

A key feature of the UC3 Starlight application is its ability to dynamically scale and efficiently process large datasets. As data volume increases, the system automatically deploys additional microservices across the federated cloud and edge infrastructure, minimizing processing time. Containerized microservices are duplicated and migrated between nodes to balance workloads and meet performance demands.

Real-time monitoring with tools like Prometheus ensures system responsiveness, while AI-driven resource allocation optimises resource usage based on performance data. This dynamic scaling allows the application to handle large datasets efficiently, delivering high-performance computing for astronomical data analysis without sacrificing speed or reliability.

4. **Migration:**

The UC3 Starlight application uses dynamic microservice migration to optimise performance and resource efficiency across federated cloud and edge infrastructure. As workloads shift or resources are limited, microservices are automatically moved between clusters to load balance, meet latency needs, and prevent bottlenecks. Managed by the LCM, migration is driven by real-time performance data, ensuring minimal disruption. AI-driven profiling ensures microservices run on the most efficient nodes, improving scalability and resilience as data volume and complexity increase.

6.6 Milestones

Table 7 lists the key milestones of UC3, highlighting crucial activities and delineating our strategic objectives. These milestones encompass various stages of development, from defining UC3 objectives and boundaries to designing the architecture and configuring the testbed. Each milestone represents a significant step in the implementation plan, ensuring a structured and systematic approach to achieving our goals.

Table 7. Integration Plan Milestones for UC3

S/N	Implementation
1	Define UC3 objectives, boundaries, and architecture.
2	UC3 Architecture Design
3	Testbed Installation and Configuration
4	Data preparation
5	Initial Containerisation of Analysis Software

6	Development of Orchestrator component MVP
7	Development of Processor component MVP
8	Refinement and testing of the UC3 Application
9	AC^3 Integration, Validation, and Documentation
10	Verification, Enhancement, and Rollout

7 Conclusions

Per the Grant Agreement, the purpose of D5.1 "***Initial integration and proofs of concept plan"*** was to provide "Description of the testbeds, as well as initial implementation and integration plans to be done during the project", capturing the initial work done in tasks T5.1 (assessment and planning of the CECCM integration) and T5.2 (description of the UC testbeds).

Rather than a plane-based integration approach, which would integrate all CECCM components prior to testbed integration, we opted for a more agile, UC based approach that will allow for integration of components required by each UC as they are needed. This provides us with greater flexibility, because each UC has different timelines, milestones and CECCM component requirements. It also enables us to start testing components, and provide feedback and corrections thereof, in a shorter timeframe. Table 8 below shows that most of the components depicted in the AC^3 architecture (see Figure 1), except for the Adaptation and Federation Layer ones, will be validated by one or more UCs, validating our UC based proposed approach.

Table 8. CECCM components exercised by Use Cases

Architecture Component	Sub-component	Validated in UC1	Validated in UC2	Validated in UC3
Application gateway (GUI)		X	X	X
Ontology & Semantic Reasoner		X	X	X
LMS Edge		X	X	
LMS Far Edge			X	
LMS Cloud		X	X	X
LMS Networking			X	X
Service Catalogue		X		
Application and Resource Management	AI-based LCM and Decision Enforcement	X	X	X
	AI-Based Resource Profile	X	X	X
	AI-Based Application Profile	X	X	X
	Monitoring	X	X	X
Data Management	Data Catalogue	X		X
	Data Manipulator	X		X

	Data Mapper	X		
	Data Provider Connector	X		X
	Data Consumer Connector	X		
	Message Broker	X		
Adaptation and Federation	Adaptation Gateway			
	Adaptation Agents			
	Resource Broker			
	Resource Discovery			

Note that the Adaptation and Federation Layer cannot be fully validated within the project UCs and their respective testbeds. Towards this end, a centralized cloud at either the ION or ARS infrastructure will be leveraged to compliment the UC testbeds. Field trials and simulations can leverage this centralized cloud to demonstrate the correctness and proper functioning of federated components.

The first findings of the integration work will be presented in the upcoming deliverable "D5.2 Report on Integration the CECCM". This document will provide additional details on the progress of AC3 components integration with each UC, updates on the status of each UC, associated risks, and any initial validation results that may be available.

8 References

[1] AC3 Project, "D2.1 "1st Release of the CECC framework and CECCM"," 31 August 2023. [Online]. Available: https://ac3-project.eu/wp-content/uploads/2024/02/AC3_D2.1.pdf

[2] AC3 Project, "D2.4 "Business Analysis of CECC and Use Case Requirements"," 27 May 2024. [Online]. Available: https://ac3-project.eu/wp-content/uploads/2024/07/Final-draft_AC3_Business-analysis-of-CECC-and-use-case-requirements_22.05.2024_FINAL.pdf

[3] AC3 Project, "D3.3 "Initial Report on Data Management for Applications in CECC"," 28 June 2024. [Online]. Available: https://ac3-project.eu/wp-content/uploads/2024/11/AC3_D3.3_20240222_v1.0.pdf

[4] AC3 Project, "D4.1 "Initial Report on Mechanisms that Enable Green-Oriented Zero Touch Management of CECC Resources"," 30 June 2024. [Online]. Available: https://ac3-project.eu/wp-content/uploads/2024/11/AC3_D4.1_to_be_submitted_annexes.pdf

[5] AC3 Project, "D3.1 "Report on the application LCM in the CEC - Initial"," 2023 December 2023. [Online]. Available: https://ac3-project.eu/wp-content/uploads/2024/11/AC3_D3.1_v7.pdf

[6] AC3 Project, "D2.3 "Report on technological tools for CECC"," 31 December 2023. [Online]. Available: https://ac3-project.eu/wp-content/uploads/2024/07/AC3_D2.3_v1.6_version_to_be_submitted.pdf

[7] ETSI, "Network Functions Virtualization," [Online]. Available: https://www.etsi.org/technologies/nfv

[8] ETSI, "Multi-access Edge Computing," [Online]. Available: https://www.etsi.org/technologies/multi-access-edge-computing