D.2.4 Business Analysis of CECC and use case requirements

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	D2.4 Business Analysis of CECC and use case requirements		
Work Package	WP2		
Contractual due date	April 30th, 2024	**Actual submission date**	May 27th, 2024
Type	R- Document, report	**Dissemination Level**	PU – Public
Lead Beneficiary	ARS		
Responsible Author	Gleibis Camejo Castillo (ARS)		
Contributors	Dimitrios Amaxilatis (SPA), Nikolaos Tsironis (SPA), Souvik Sengupta (IONOS), Cristina Catalán (UCM), Abdelhak Kadouma (FIN), Ibrahim Afolabi (FIN), Adlen Ksentini (EUR), Mohamed Mekki (EUR), Akram Boutouchent (EUR), Sara Madariaga (ARS)		
Peer reviewer(s)	Abdelhak Kadouma (FIN), John Beredimas (CSG), Dimitrios Amaxilatis (SPA)		

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	03/05/2023	5%	Initial Deliverable Structure	Gleibis Camejo Castillo (Arsys) Souvik Sengupta (IONOS)
V0.2	31/05/2023	20%	Use Cases information provided	Cristina Catalán (UCM)
V0.3	06/06/2023	40%	First draft for Business Model provided	Gleibis Camejo Castillo (ARS)
V0.4	23/06/2023	45%	Final version for Business Model and stakeholders' interactions provided	Gleibis Camejo Castillo (ARS) Souvik Sengupta (ION)
V0.5	15/09/2023	70%	Completed detailed information on the use cases	Abdelhak Kadouma (FIN) Cristina Catalán (UCM)
V0.6	30/11/2023	90%	Prepared the initial full draft	
V0.7	26/01/2024	95%	Received internal reviewers' feedback Completed Challenges section	Abdelhak Kadouma (FIN) John Beredimas (CSG) Athanasios Kordelas (CSG) Dimitrios Amaxilatis (SPA)
V1.0	31/01/2024	100%	Prepared the final draft	Souvik Sengupta, Ali Nikoukar, Arian Firouzbakhsh (ION) Adlen Ksentini, Mohamed Mekki (EUR) Gleibis Camejo Castillo (ARS) Dimitrios Amaxilatis, Nikolaos Tsironis (SPA) Luis Angel Garrid Platero (IQU) Ibrahim Afolabi, Abdelhak KADOUMA (FIN) Vrettos Moulos (UNIPI) Dimitris Klonidis (UBI)
V1.1	29/03/2024	100%	Prepared the final draft	Souvik Sengupta, Ali Nikoukar, Arian Firouzbakhsh (ION) Adlen Ksentini, Mohamed Mekki (EUR) Gleibis Camejo Castillo, Sara Madariaga (ARS) Dimitrios Amaxilatis, Nikolaos Tsironis (SPA) Luis Angel Garrid Platero (IQU)

				Ibrahim Afolabi, Abdelhak KADOUMA (FIN) Vrettos Moulos (UNIPI) Dimitris Klonidis (UBI)
V1.2	23/04/2024	100%	Reviewed the final draft	Christos Verikoukis (ISI/ATH) Adlen Ksentini (EUR)

Disclaimer

Copyright message

Table of Contents

List of Figures

List of Tables

Glossary of terms and abbreviations used

Abbreviation / Term	Description
AC³	Agile and Cognitive Cloud edge Continuum management
AG	Application Gateway
API	Application Programming Interface
CECC	Cloud Edge Computing Continuum
CECCM	Cloud Edge Computing Continuum Manager
CFN	Compute First Networking
CFRA	Cloud Federation Computing Reference Architecture
CNCF	Cloud Native Computing Foundation
CRUD	Create, Read, Update, and Delete
FHS	Federation hosting service
GUI	Graphical User Interface
IDSA	International Data Spaces Association
IaaS	Infrastructure as a Service
IoT	Internet of Things
KPI	Key Performance Indicator
LCM	Life-Cycle Management
LMS	Local Management System
MEC	Multi-Edge Computing
ML	Machine Learning
NBI	Northbound Interface
OSR	Ontology and Semantic aware Reasoner
OWL	Web Ontology Language
PaaS	Platform as a Service
PoC	Proof of Concept
PLI	Profile Language Interpreter
RDF	Resource Description Framework

SaaS	Software as a Service
SBI	Southbound Interface
SDWAN	Software-Defined Wide Area Network
SIIF	Standard for Intercloud Interoperability and Federation
SLA	Service Level Agreement
TBs	Tight Binding
UAV	Unmanned Aerial Vehicle
WAN	Wide Area Network

1 Executive Summary

The present document D2.4 Business Analysis of CECC and use case requirements is a report on the analysis conducted for the techno-economic aspects of the Cloud Edge Computing Continuum (CECC) resource federation. In the document, we identify the interactions between the stakeholders and the Business Model for the Cloud Edge Computing Continuum Manager (CECCM), as well as the challenges faced by the current federation models. Finally, we report detailed and refined information on the Use Cases (UCs), including challenges, requirements, architecture, solutions proposed, and the Key Performance Indicators (KPIs) to have successful outcomes in the context of AC^3.

Section 2 presents how the continuum represents a flexible and scalable approach to computing aiming to balance the advantages of centralized cloud infrastructure with the benefits of distributed edge, optimizing the computing infrastructure to meet the evolving requirements of modern applications and services. However, the CECC introduces several challenges for organizations, mainly at the technical level, that will be addressed by AC^3 in order to make application deployment at the CECC feasible for any organization.

In Section 3, we analyze how the main Business Model around the CECCM can be seen as a mixture of IaaS, PaaS and SaaS models, and relate them to the layer (plane) structure of the high-level architecture proposed in deliverable "D2.1 1st Release of the CECC framework and CECCM". The main differences between our proposal and the classical computing stacks are: 1) substituting the so-called Runtime layer by AC^3's Application and resource management component, and 2) merging the Operating System, Virtualization, and Servers layers into AC^3's Computing component, the CECC.

We also identify and describe the different CECC stakeholders, both within AC^3 and also extending it outside of the project consortium. There are four main families of stakeholders: resource providers (including data, CECC and telco), resource managers (infrastructure, service catalogue, and data), resource consumers (application developers, end users), and regulatory / standardization bodies.

We also provide a comparative analysis of the two main existing computing federation frameworks: the NIST CFRA and Gaia-X. NIST introduces a 3-plane federation model: trust (so service providers can exchange sensitive information), management (how to join a federation and consume resources), and usage (exposed to users). Gaia-X builds on this 3-plane model and focuses more on enabling sovereign data exchange among organizations within and across data spaces.

Both of them have limitations in terms of security and trust, SLA compliance, dealing with stateful microservices, data management, and lifecycle management in heterogeneous CECC federations. We finish by briefly explaining how AC^3 will overcome all these challenges by means of its novel CECCM framework, which introduces three key elements: a sophisticated application LifeCycle Management (LCM), resource optimization for all the CECC components (including far-edge and networking), and a trusted resource federation leveraging AI, ML capabilities to proactively optimize resources and guarantee application SLA.

Section 4 provides a theoretical approach to the incentives that the CECC and CECCM offer to the different stakeholders. We present a summary of the auction and game theories and conclude that there is a gap in terms of contextual focus on CECC, as they primarily address challenges from the perspective of infrastructure providers. The AC^3 approach integrates other stakeholders, including cloud, service, and far-edge infrastructure providers, and considers the interactions between them described in Section 3.

AC^3 relies on the CECCM to ensure an adequate decision-making. It first receives the price units specified by different infrastructure providers and informs the service providers. Then, it receives the bids of service providers, which allocate resources to their users. By means of a multi-agent reinforcement learning solution,

the CECCM generates the appropriate configurations in order to assign tasks to each infrastructure provider, and also manages the budget received by both the service and the infrastructure providers.

Section 5 goes into the details of the different use cases that will serve as testbeds for AC³'s CECCM, explaining their diverse challenges and requirements. For each use case, we propose a preliminary solution together with a draft architecture based on the AC³ framework. This shows how the project can provide solutions to heterogeneous situations.

Use Case 1 involves an IoT-based smart sensing and monitoring framework for infrastructures. Its main challenges are related to reliability of edge devices, connectivity and network availability, data integration, data management and processing, data security and privacy: system scalability, and maintenance. AC³ proposes a combination of technical solutions, best practices, and organizational strategies provided by the CECCM, including: robust encryption and authentication for data in transit and at rest, redundancy and failover mechanisms and automated software updates to ensure continuous. The Data Management component will guarantee a streamlined data integration process using APIs for data connectors and standard formats and protocols. The testbed network is conceived as a 5G network comprising the following elements: a 5G Core Cloud Domain, a RAN Domain with Multi-Edge Computing nodes, and a selection of 5G-compliant User Equipments.

Use Case 2 will build a powerful and effective video surveillance and streaming system by combining IoT, camera, and UAV technologies. The main challenges are: 1) the computational and hardware heterogeneity and sensing capability of the IoT devices and 2) the limited storage and processing capabilities of some IoT devices, which impedes performing adequate data processing. AC³'s CECC framework will optimize the distribution of computing resources and processing capabilities, resulting in seamless integration of cloud computing and edge computing technologies. This integration will reduce latency and optimize bandwidth, and also significantly improve the reliability, performance, scalability, and flexibility of the surveillance system.

Use Case 3 deals with the distributed processing of 3D datacubes, multidimensional datasets that combine spatial and spectral information. Working with 3D datacubes has the following main challenges: 1) increased data volume, significant storage capacity, additional computational resources and longer processing times, and 2) data exploration and interpretation, parameter estimation, and interdisciplinary collaboration to analyze 3D datacubes. AC³'s modular architecture, combined with containerization, streamlines the management and deployment of analysis tools dedicated to processing datacubes. This architectural design helps with the integration of memory management strategies, software optimizations, and distributed computing techniques. By utilizing container orchestration platforms, we can dynamically allocate computing resources according to workload demands, optimizing the computational resources for processing large volumes of datacubes

Section 6 gives the first conclusions of the analysis, relating them to the project goals. We conclude that most of the objectives of task T2.1 have been covered, leaving for the next version of deliverable D2.1 "1st Release of the CECC framework and CECCM" the incentives for stakeholders and end users.

Finally, Section 7 provides a comprehensive compilation of existing CECC frameworks and research works.

2 Introduction

The Cloud Edge Computing Continuum (CECC) originates from the need to optimize computing resources and data processing capabilities across a spectrum ranging from centralized cloud servers to distributed edge devices. This continuum is driven by the increasing demand for low-latency applications, real-time data processing, and efficient resource utilization in a world where data is increasingly generated at the edge of the network. By integrating cloud computing with edge computing technologies, organizations can leverage the strengths of both approaches to better meet the requirements of modern applications and services. The continuum represents a flexible and scalable approach to computing that aims to balance the advantages of centralized cloud infrastructure with the benefits of distributed edge computing closer to where data is generated and consumed, optimizing their computing infrastructure to meet the evolving requirements of modern applications and services.

The main advantages of the CECC are:

- **Low Latency**: Many applications, such as IoT, autonomous vehicles, real-time analytics, and AR/VR, require low latency for optimal performance. By moving computing resources closer to the edge where data is generated, edge computing reduces latency and improves the responsiveness of these applications.
- **Bandwidth Optimization**: Processing data at the edge reduces the volume of data that needs to be transmitted to centralized cloud servers. This optimization of bandwidth usage not only reduces costs but also avoids network congestion and bottlenecks.
- **Scalability and Flexibility**: The Cloud Edge Computing Continuum provides a scalable architecture that can dynamically allocate computing resources based on the demands of applications. This flexibility ensures efficient resource utilization and enables organizations to scale their infrastructure as needed.
- **Data Privacy and Security**: Edge computing enhances data security and privacy by processing sensitive information locally and transmitting only necessary data to the cloud. This approach reduces the risk of data breaches during transit and helps organizations comply with regulations governing data privacy.
- **Resilience and Reliability**: By distributing computing resources across the cloud and edge, the continuum improves the resilience of applications. Edge computing ensures that critical operations can continue even when connectivity to the cloud is disrupted. The cloud provides the necessary processing and backup capabilities that edge is missing, making the continuum more robust.
- **Real-Time Data Processing**: Edge computing facilitates real-time data processing and analysis, enabling organizations to derive actionable insights from data at the point of origin. This real-time processing enhances decision-making and enables faster response to events. The cloud complements these capabilities by providing storage for big amounts of data and computing resources for training larger AI models with historical data.
- **Offline Functionality**: Edge computing enables devices to perform computational tasks even when disconnected from the cloud. This offline functionality is critical for applications that require continuous operation in environments with unreliable or intermittent connectivity.

However, the CECC introduces several challenges for organizations at technical and business levels, making it hard to elaborate adequate business models and, therefore, hindering its adoption. These include:

- **Complexity**: Managing a hybrid infrastructure that spans both centralized cloud servers and edge devices adds complexity to system administration. Coordinating and monitoring resources across the continuum requires specialized skills and tools.

- **Resource Management**: System administrators need to efficiently allocate resources across the cloud and edge to ensure optimal performance. Balancing workloads, data storage, and processing capabilities between the cloud and edge devices can be challenging.
- **Security Concerns**: Securing a distributed architecture that includes edge devices introduces new security challenges. System administrators must implement robust security measures to protect data and applications at the edge, especially in environments where physical security may be a concern.
- **Data Management**: Application developers need to consider data management strategies that account for the distributed nature of the CECC. Ensuring data consistency, integrity, and availability across cloud and edge locations is essential.
- **Application Deployment and Orchestration**: Deploying and orchestrating applications in a hybrid environment requires careful planning and coordination. Application developers must design applications that can seamlessly operate across the cloud and edge, considering factors such as latency, bandwidth, and resource constraints.
- **Integration Challenges**: Integrating edge devices with existing cloud infrastructure and applications can be complex. Application developers and system administrators need to ensure seamless communication and data flow between edge devices and centralized cloud servers.
- **Monitoring and Maintenance**: Monitoring the performance and health of distributed resources in the Cloud Edge Computing Continuum is crucial for system administrators. Implementing effective monitoring tools and practices to detect and address issues across the continuum is essential for maintaining system reliability.
- **Compliance and Regulations**: Meeting regulatory requirements related to data privacy, residency, and processing presents a challenge in a distributed environment. System administrators and application developers need to ensure compliance with regulations governing data protection and privacy.
- **Costs**: Implementing an infrastructure that spans both the cloud and edge can incur additional costs. Businesses need to invest in edge devices, network infrastructure, and management tools, which can impact the overall cost structure of the organization.
- **Skill Gaps**: Leveraging the CECC requires specialized skills in areas such as edge computing, IoT, networking, and cloud technologies. Businesses may face challenges in finding and retaining talent with the necessary expertise to support this continuum.

The thorough analysis of uses cases performed by the AC³ project partners allowed us to design the architecture of the CECC management system (CECCM). The CECCM will leverage the appropriate tools and technologies to play a critical role in addressing these challenges and provide an adequate framework to ensure the necessary conditions for delivering reliable and efficient services, facilitating the optimal management of IT and network resources and the optimization of energy consumption. All this work led to the definition of the incentives and business models for stakeholders that will be presented in Section 4.

2.1 Mapping AC³ Outputs

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			
D2.4 Business Analysis of CECC and use case requirements Report analyzing the techno-economic aspects of the CECC resource federation and identifying the interactions between the stakeholders. Focusing on business model to be followed and providing detailed requirements of the AC³ use-cases with their own KPIs.			
TASKS			
T2.1 Business analysis and use-case requirements	- Conduct a detailed analysis on the techno-economic aspects of CECC resource federation. - Identify the interactions between stakeholders. - Identify the new challenges that might arise in a federation model. - Reassess the pre-defined set of use cases and perform adjustments as deemed fit to adjust with technological advances and market evolution and ensure feasibility and relevance. - Revisit the determined KPIs and target values for each use case to assess performance accurately.	Section 3 Section 4	Revisiting the defined objectives, investigating the techno-economic aspects of the CECC resource federation, and going deeper into the use-case requirements provide the basis for building the system architecture and understanding the economic incentives of the different CECCM stakeholders.

2.2 Deliverable Overview and Report Structure

In this section we provide a brief description of the structure of the deliverable.

- Section 3 provides a state-of-the-art of the techno-economic aspects of the CECC resource federation that includes the business model selected, together with the interactions among the stakeholders that have been identified, to fulfill the overall objectives of the project, as well as the challenges that might arise in a federation model in the relevant initiatives and existing innovations, which will help to develop the functional architecture for CECC. In this section, the consortium has mainly given their utmost focus on studying the latest Gaia-X/IDSA and NIST specifications for infrastructure and data federation [1].
- Section 4 provides detailed information on each use case including challenges, requirements, architecture and the solution proposed. Additionally, Key Performance Indicators (KPIs) are defined for each use case.

3 Analysis of the techno-economic aspects of CECC

3.1 Business Model

The main Business Model around the CECCM can be seen as a mixture of Infrastructure and Platform as a Service (IaaS & PaaS) models, followed by a generalization of Software as a Service (SaaS) due to the layer or plane structure of the high-level architecture proposed in deliverable "D2.1 1st Release of the CECC framework and CECCM", see Figure 1 below:

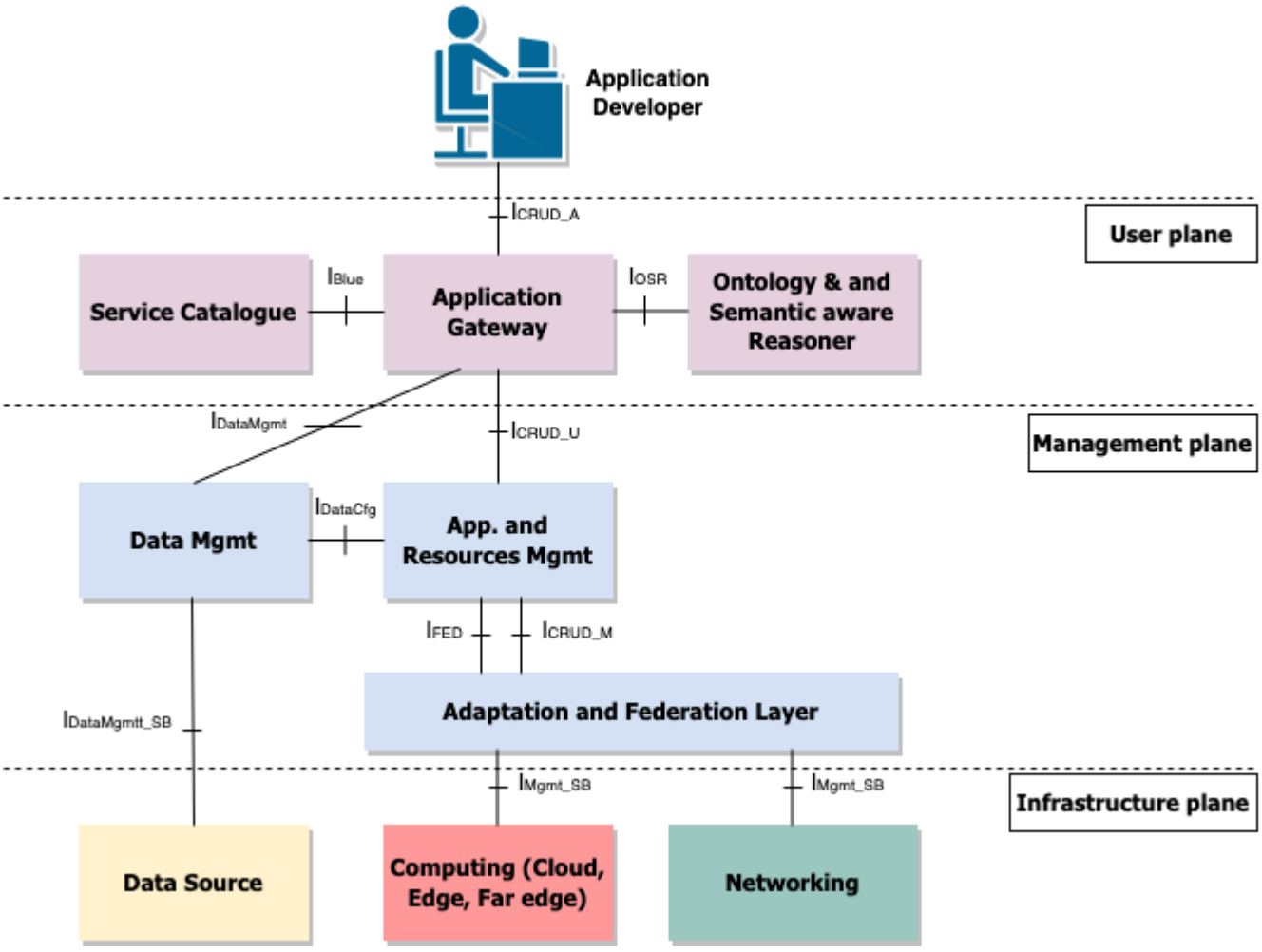

Figure 1. High-level architecture of AC^3.

We briefly recall here the role of each plane and its components.

- The user plane groups all the necessary functionalities to allow an application developer to build data-driven and micro-service-based applications using a common descriptor to define needed resources in terms of data, computing, and networking. It includes:
    - The Application Gateway allows an application developer to interact with the CECCM to develop, deploy, and manage applications' life cycle.
    - The service catalog includes a blueprint of the application's description and information about data sources.

  - The Ontology and Semantic aware Reasoner translates and interprets all policies used by different CECCM actors (e.g., data source and application developers).

- The CECC plane corresponds to the infrastructure constituting the CECC, i.e., a federation of data sources, computing nodes (central cloud, edge, and far edge), and networking belonging to potentially different providers.
- The management plane integrates the necessary management and orchestration functions to handle both the Life Cycle Management (LCM) of applications and their related data, as expressed by the application developer through the user plane, while considering the CECC infrastructure resources. includes three key components:
  - Data management: it manages access to cold and hot data.
  - Application and resource management: it oversees applications and the CECC infrastructure management and orchestration: application instantiation over the CECC infrastructure, handling of the application runtime to guarantee SLA, resource picking from the federation, and monitoring of the application behavior.
  - Abstraction and federation layer: it abstracts the heterogeneity of the infrastructure layer to the application and resource management component, exposing a common API for resource discovery and CRUD operations over the federated CECC and translating everything to the infrastructure-specific APIs.

Table 2 shows the different levels of an application stack and the division between business owners and external vendors in the three different models (IaaS, PaaS, and SaaS). The last column shows the hybrid business model for CECC proposed by the AC^3 consortium.

Table 2. Example of pure IaaS, PaaS and SaaS Business Models and proposal for AC^3.

IaaS	**PaaS**	**SaaS**	**AC^3 proposal**
Application	Application	Application	Application Gateway
Data	Data	Data	Data Management
Runtime	Runtime	Runtime	Application and resource management
Adaptation & Federation	Adaptation & Federation	Adaptation & Federation	Abstraction and federation
Operating System	Operating System	Operating System	Computing (Cloud, Edge, Far edge)
Virtualization	Virtualization	Virtualization	Computing (Cloud, Edge, Far edge)
Servers	Servers	Servers	Computing (Cloud, Edge, Far edge)

Storage	Storage	Storage	Data Source
Networking	Networking	Networking	Networking

Legend

Handled by the business itself

Handled by external vendors

Layers to be modified within the AC³ scope

3.2 Stakeholders within AC³ environment and beyond

To have a better understanding of the role of each stakeholder within the scope of AC³, we provide below a list of roles with their associated main functions. Figure 2 shows an indicative diagram of the interactions between the identified families of stakeholders.

- **Computing Continuum Provider**: provider of computing resources across the CECC. It should allow the integration across distributed and multicloud infrastructures, while supporting IoT and far-edge applications. Within the framework of the AC³ project, such a provider will interact with:
    - **Cloud Service Providers**: to ensure on-demand access to shared resources over the internet, offering scalability and flexibility to its users (FIN, ION, ARS).
    - **Edge Computing Providers**: to integrate systems and services from data centres in the cloud and devices in the edge and far edge, a function that is key in the performance for the CECCM due to the possibility of processing data closer to its source, reducing latency (EUR).
    - **Far-edge Providers**: to interact with devices which capture and process data locally, allowing the synchronization with the rest of the continuum for an optimal application performance (EUR).
- **Infrastructure Resource Manager:** it will assume the operation of the CECCM which will be developed during the project. This implementation will include resource discovery and brokering functionalities, as well as adaptation agents and gateway, which will communicate directly with the monitoring and lifecycle management modules. See further details in the deliverable D2.1.
- **Telecom Provider(s):** responsible for providing the connectivity needed to enable the CECC, out of the scope of the AC³ consortium tasks; we will rely on external provider(s).
- **Device Manufacturers/Owners:** this role will be played either by 1) partners that produce data as a result of running the use cases in the scope of AC³, or 2) partners that will exploit these data to deliver the project results for business, educational, or scientific purposes. Specifically, we have:
    - **Data Owners**: FIN, IQU, SPA, UCM, RHT, EUR.
    - **Data Consumer**: UPR.
- **AC³ Service Catalogue & Data Management Engine**: orchestrate the application access and data availability from a single access point through APIs to all the data compiled by the different stakeholders.
- **Application/Software developers**: group of partners that will develop or own the applications and/or software to be deployed in the project.

- **Vertical ecosystems**: related to the clusters of partners that work together with the responsibility for developing and executing the proposed use cases (more detail in Section 5).
- **Regulatory and Standardization bodies**: define regulations and standards related to the different CECC components and also to data management. Although they are not part of the consortium, we will closely follow up on their activities and publications which are relevant to the project. Examples of these organizations are: ITU, International Data Spaces Association, Gaia-X, NIST, or Cloud Security Alliance.
- **End User**: any stakeholder that would consume the solution within the proposed CECC framework.

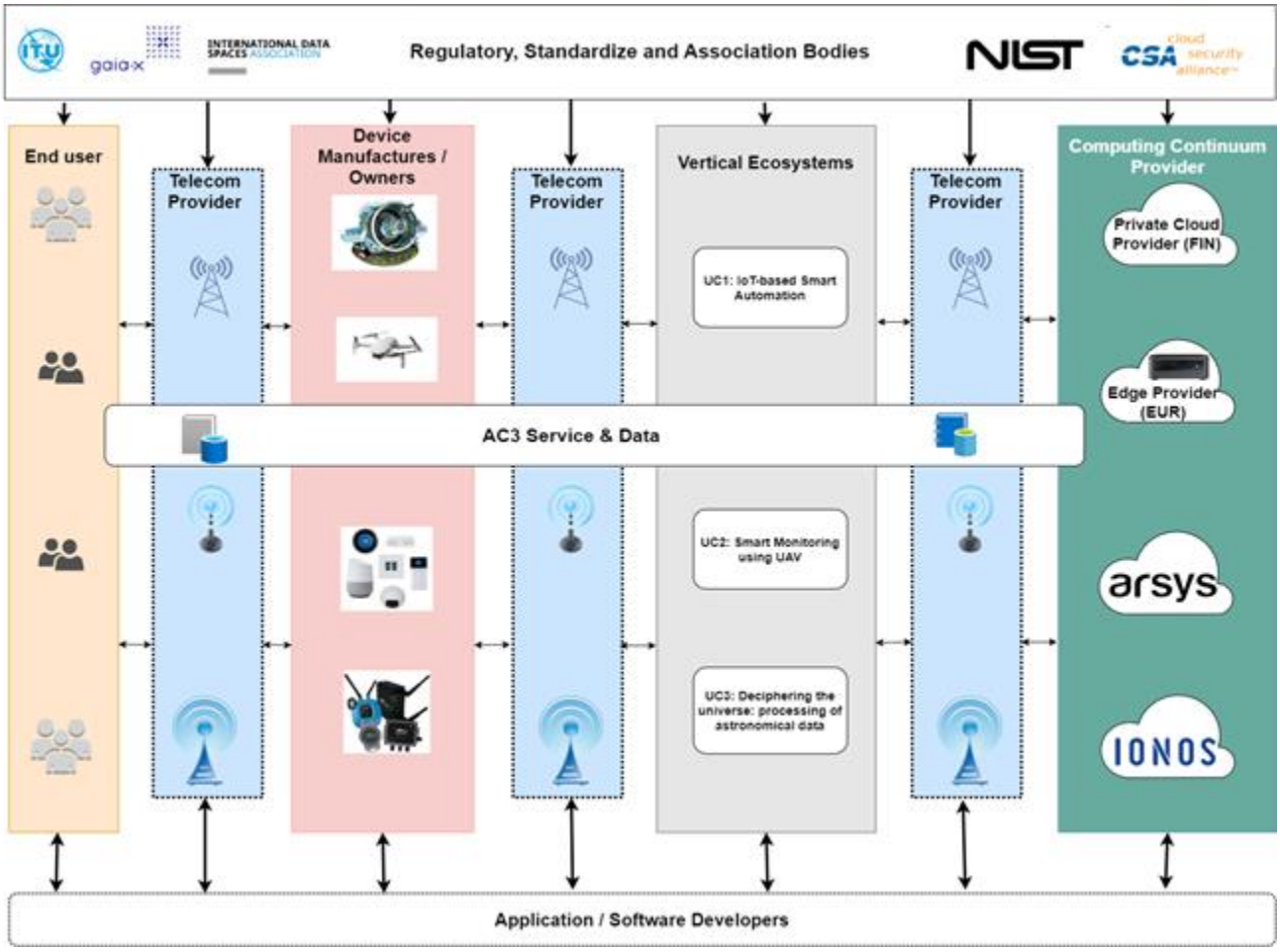

Figure 2: Stakeholder Interactions foreseen in the project.

Table 3 explains the expertise and roles of the consortium partners within the AC³ project. Then, Figure 3 below shows the interactions among them.

Table 3. AC³ partners' expertise and role in the project.

Partner	Expertise and role in AC³
ISI/ATH	ISI/ATH brings expertise in resource allocation, energy efficiency and cloud native applications. ISI/ATH will contribute to AI/ML techniques for resource allocation and will offer service migration solutions.

Partner	Expertise and role in AC³
EUR	EUR will contribute to XAI/AI/ML algorithms, LMS for far-edge, and programmable networking. EUR will host and participate in UC3, and co-lead UC2.
IBM	IBM is the largest middleware company in the world. Ireland Software Group will lead IBM's participation in the project via the Ireland Lab Innovation Exchange. They will apply intent-based and AI-assisted resource management for AC³ and a range of problems/use cases addressed in the project.
CSG	Service Provider Centre-of-Excellence, with expertise in L4-L7 traffic optimization and ADC solutions, integrated with NFV MANO stack, SDN controllers and CNCF stack. ML/AI & Cybersecurity Centre of Excellence, with expertise in cloud-based, multi-offering, multi-tenant, secure-by-design, real-time data processing and multi-modal big data storage, ML, and analytics platform, enabling a security analytics (UEBA) solution.
ARS	Cloud infrastructure and service provider, a pioneer of the public cloud in Europe and specialized in designing fully customized cloud projects. It will integrate computing resources from its portfolio, as needed in the use cases, and provide expertise in Gaia-X and data spaces connectors.
ION	One of the largest cloud providers in Europe, it brings the required expertise to support the design of the CECC architecture. Additionally, IONOS will use their hands-on cloud service expertise in the implementation and integration of the demonstrative testbed, ensuring compliance with standards; e.g., Gaia-X.
RHT	RHT is the world's leading provider of enterprise open-source solutions, including high-performing Linux, cloud, container, and Kubernetes technologies. Red Hat will extend the state-of-the-art of features in the open-source projects that will support this project's infrastructure, among them, the interconnectivity between clusters, CI/CD deployments, low-footprint Kubernetes in the Edge, standard Kubernetes deployments, and the cluster's resources management.
SPA	SPA is a nascent technology company delivering advanced hardware and software products in the areas of IoT and smart devices, data analytics, ambient intelligence and cloud and edge computing as a member of the AWS partner network.
UBI	Innovative software house and technology provider for leading edge intelligent technical solutions Its major contributions relate to application profile modelling and LCM and microservices-based application deployment.
FIN	Finnish SME with solid expertise in cloud computing, data management, security and trust management, and the application of disruptive technologies such as AI for resource management. It will contribute to AI for resource allocation and management and play a leading role in hosting and demonstrating UC2.
IQU	IQU has expertise in software development, consultancy in wireless network planning, and R&D for beyond-5G networks, with extensive know-how and state-of-the-art contributions in AI/ML, IoT Networks, SDN, and Cloud Computing. IQU will lead UC1.
UCM	Pioneer in the design and build of advanced astronomy instruments such as MEGARA, it participates in the development of world-wide instruments (MOSAIC) thanks to their Advanced Scientific Instrumentation Laboratory (LICA). UCM will lead UC3.

Partner	Expertise and role in AC^3
UPR	UPR brings expertise in AIOps, services and applications dimensioning (in terms of the required resources based on various service parameters), deployment patterns optimization, data management within and across DCs, and AI-driven orchestration. Additionally, several AI/ML research efforts have led to models utilized in various application domains (e.g., finance, transportation, health, logistics, manufacturing environments, etc.).

Figure 3 below showcases two types of interactions:

- Double lines represent two-way interactions.
- Single-line arrows indicate how some components control the operation and enforce actions on other parts of the CECC.

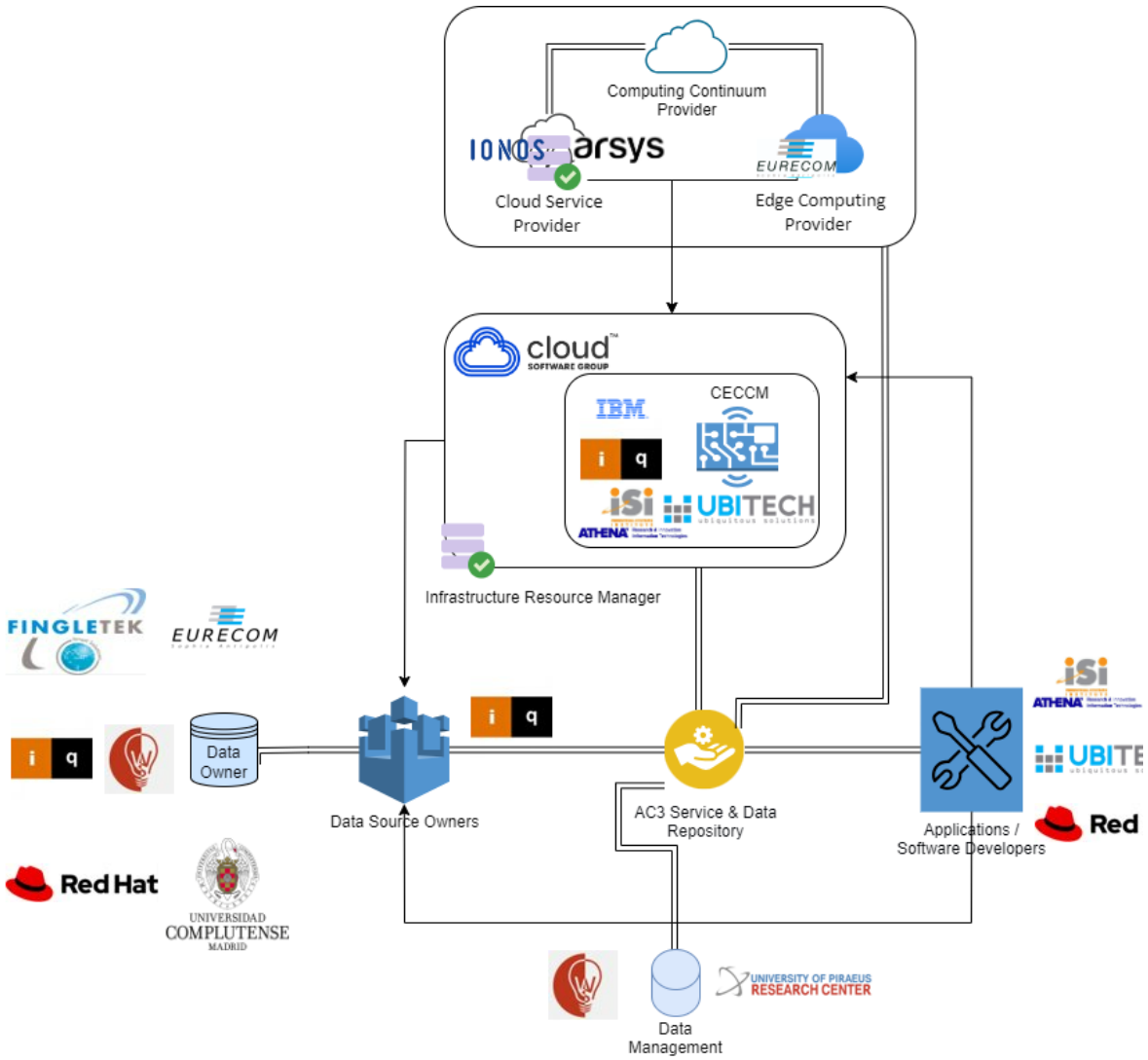

Figure 3: Indicative interactions among the AC3 consortium´s partners.

3.3 Challenges faced by current federation models

This section provides an overview of the existing federation models and frameworks as well as the challenges the AC³'s design and implementation will need to overcome.

3.3.1 Overview of existing frameworks

In the context of this deliverable as federation models we consider Gaia-X [2] and NIST Cloud Federation Computing Reference Architecture (CFRA) [3]. Note that we only provide a summary of these models to document their limitations and the challenges posed relevant to a Cloud Edge Computing Continuum (CECC) architecture and the AC³ project, rather than an exhausting presentation that extends beyond the scope of this document. For further details, please refer to the project deliverable D2.1, "1st Release of the CECC framework and CECCM".

3.3.1.1 NIST overview

The NIST CFRA [3] starts with a somewhat abstract and simplistic view of a federation (Figure 4) wherein:

- Users in Organization A can discover and invoke services in Organization B.
- Service Providers in Organization B can validate credentials from Organization A and make the proper access decisions.

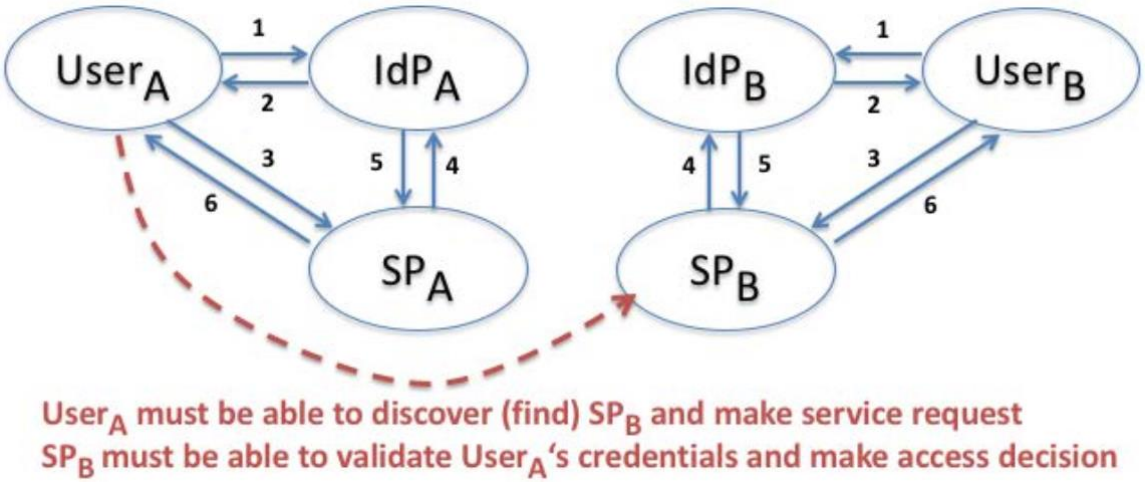

Figure 4. Federated authentication and authorization. Source: NIST CFRA.

Starting from this simple abstraction, NIST defines a Cloud Federation as a Virtual Administrative Domain where Cloud Consumers can connect and consume services across multiple service providers. However, this virtual domain only corresponds to the usage plane, which is exposed to end users and cloud consumers. Firstly, to enable this administrative domain, the Service Providers involved need to establish some sort of trust to exchange sensitive information, including but not limited to available services, APIs, user credentials, and/or tokens. Secondly, they need to jointly define federation membership details, namely users that join it and consume resources from their non-home service provider, their permission levels, the type of services and resources they can consume, etc. Thus, NIST introduces a 3-plane Federation model, which is illustrated in Figure 5.

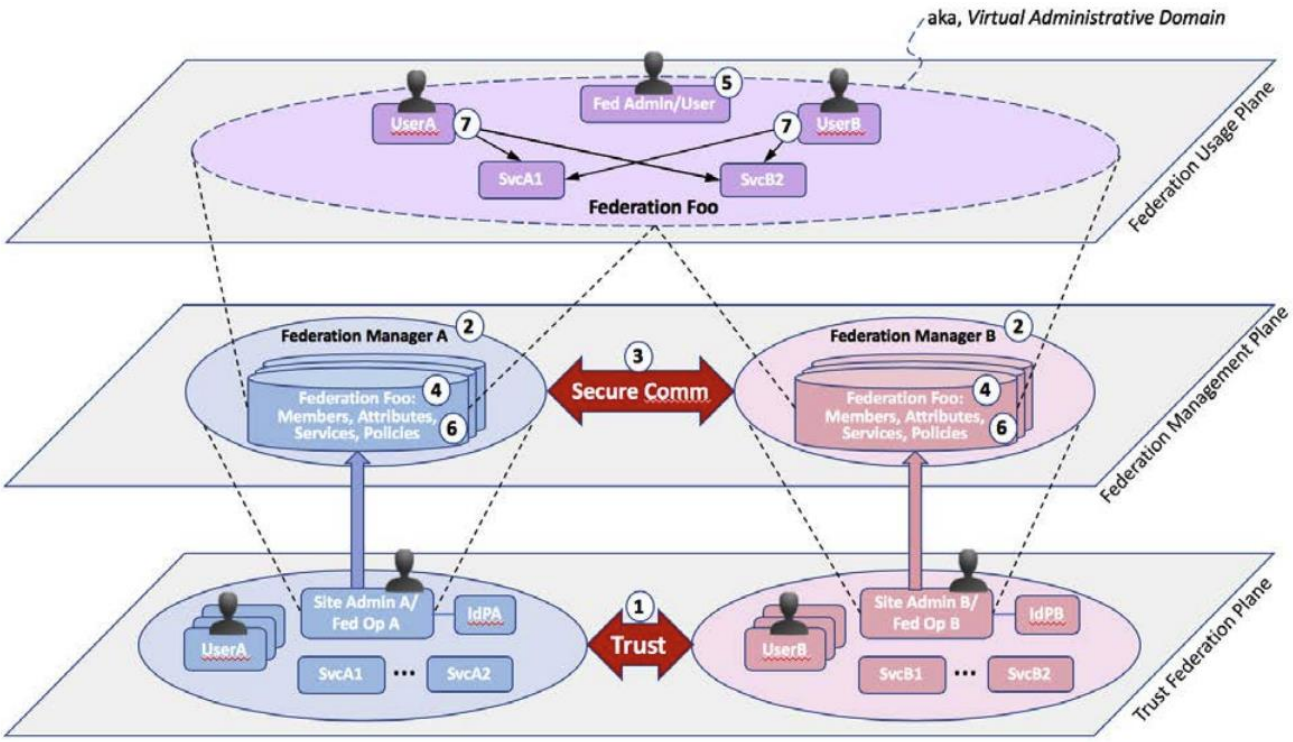

Figure 5: A Three-Plane Illustration of the CFRA. Source: NIST-CFRA [3].

The remainder of the CFRA is an actor/role-based model with five distinct roles somewhat essential to the formation of a federation (see Figure 6): the Federation Manager(s), Operator(s), Carrier(s), Broker(s), and Auditor(s). It defines in detail each of these actors, together with their responsibilities and interactions, not only among themselves but also with other actors who are also part of the standard Cloud Reference Architecture, such as Identity Providers, Cloud Service Consumers, Cloud Service Providers, and more.

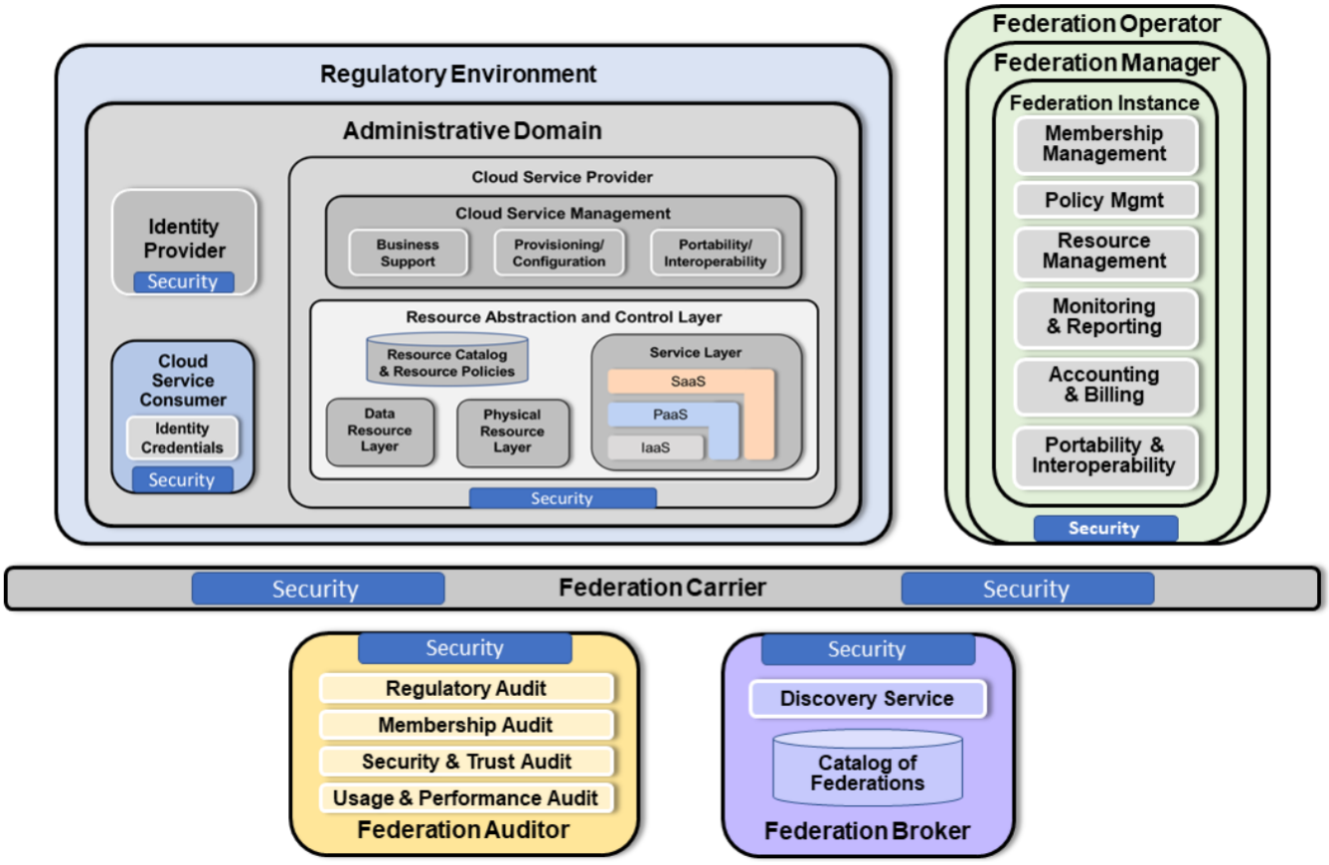

Figure 6: The NIST Cloud Federation Reference Architecture Actors. Source: NIST-CFRA [3].

Having established the essential actors and roles in creating a federation, the NIST model moves away from the initial simplistic 2-peer federation view and considers more complex federation implementations, such as peer-to-peer, centralized, hierarchical, and various variations thereof. For example, a sample hierarchical deployment with a per-site Federation Manager is illustrated in Figure 7 below.

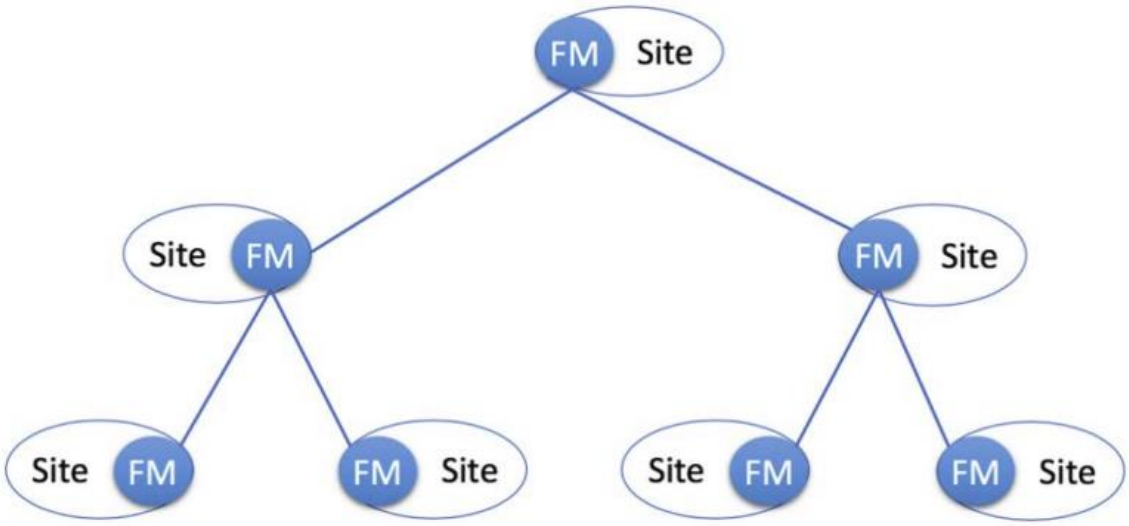

Figure 7: Large Hierarchical Internal FM Deployments. Source: NIST-CFRA [3].

3.3.1.2 Gaia-X

Gaia-X [2] provides a federated and secure data infrastructure that enables hybrid solutions by providing links between cloud service providers. It aims to create a federated open data infrastructure by designing and implementing a data-sharing architecture. More specifically, it provides the components to address compliance, federation, and interoperable data exchange. An overview of the main components and their basic interactions is depicted in Figure 8 below [4].

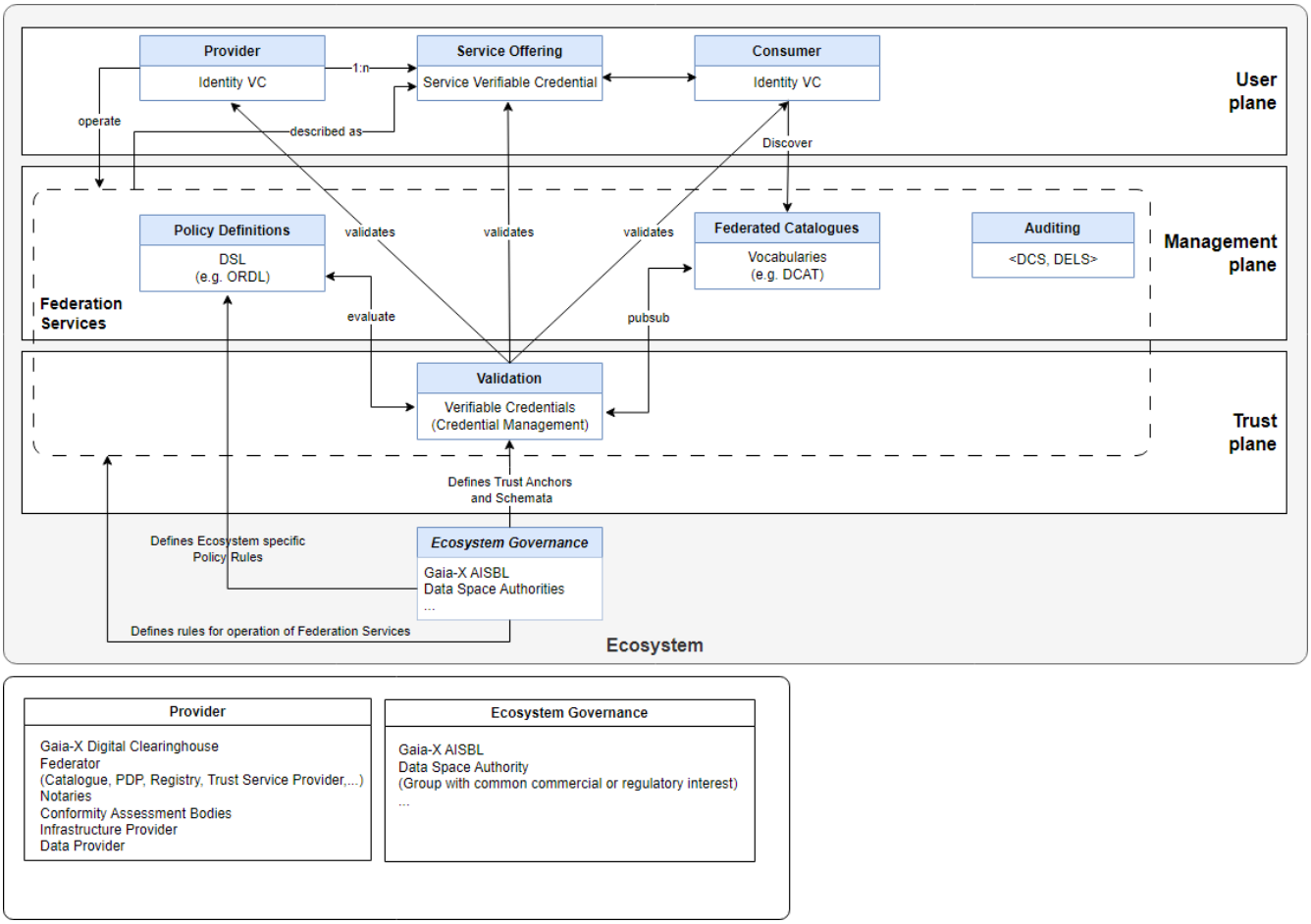

Figure 8. Gaia-X conceptual model overview. Source: Gaia-X Architecture Document [4].

To some extent, Gaia-X shares common goals with the NIST CFRA, namely promoting interoperability across federated cloud infrastructure providers. However, NIST is broader in scope in that it strives to provide a generalized framework and guidelines that are not specific to any particular region. On the other hand, Gaia-X has a more specific focus:

- It has a strong emphasis on fostering a European federated infrastructure.
- It places a significant focus on data sovereignty, aiming to provide a framework where organizations not only have control over their data but can also ensure that data sharing, processing, and storage are in accordance with EU laws and regulations (i.e., GDPR).
- It encourages vertical collaboration across various industries (Aerospace, Finance, Energy, Agriculture, and others) to create a common data infrastructure, also known as “data space”.
- It aims to establish common standards to ensure interoperability and trust within the federation rather than leveraging existing ones.

Adoption of the Gaia-X framework is mostly relevant to projects or groups that need to join an existing federated infrastructure implemented on top of it, such as data/cloud infrastructure providers or members of an industry vertical. However, it would introduce a significant complexity for AC^3 since the project would need to consider several requirements that are not necessarily relevant to the use cases at hand. In addition, while Gaia-X can be used to implement federated cloud infrastructure, its focus is mainly on data processing rather than computing or otherwise a generalized federated infrastructure. Last but not least, Gaia-X does not make any special considerations for the federation of disparate infrastructure, such as far-edge resources that a CECC needs to consider.

Note that the AC^3 project will take advantage of the knowledge gained and difficulties already encountered in the Gaia-X project in order to propose a less prescriptive solution that allows for:

- Cloud Management and federation support, in addition to Data Management
    - Security and Trust
    - Service Level Agreement
- Easier deployment, configuration, and integration of the framework
    - Stateful Microservices
    - Lifecycle management across non-uniform infrastructure
- Support of far-edge devices

3.3.2 Limitations of existing federation frameworks

The current federation frameworks present several challenges in the context of the Edge and Far Edge Cloud continuum in the domains of security and trust, data management and federation, and microservice lifecycle management. The remainder of this section presents a rough summary of the relevant challenges that our AC^3 evolved architecture addresses.

3.3.2.1 Security and Trust

Security and trust management are critical for AC^3 since they aim to federate resources and data on the fly from various domains and devices. Of particular importance are Far Edge devices, such as drones, IoT gateways, vehicles, and others.

Depending on requirements, the central and edge cloud components of the CECC can fit nicely with the NIST model using either a centralized or hierarchical view, similar to what is depicted above. However, Far Edge resources introduce unique challenges to the trust model of a federation. The NIST model does take into consideration that its rigid approach is not suitable for all purposes, noting that some federations may have very lenient membership requirements, i.e., any user or site could self-identify and join the federation. On the other hand, it only considers a somewhat controlled membership approach, calling for some process for vetting and on-boarding new members. The existence of a federated identity credential could possibly be derived from a member's home institution credentials, calling for each site to have a site admin and possibly run its own federation manager. Its onboarding process calls for either simple self-identification or non-trivial steps, such as in-person interviews, recommendations from current members, or known reputation.

Overall, the NIST CFRA trust model hinges on the assumption that federation membership is a somewhat long-lived attribute of a participating site. It may leave room for lenient membership and revocation but fails to recommend specific actors/roles, processes, and interactions that would accommodate quick and on-the-fly onboarding of new sites to the federation in a manner that still allows for:

- a minimum yet acceptable level of trust that can establish guarantees that sites will adhere to the federation's operational, regulatory, and other requirements.

- a quick revocation process to expunge from the federation misbehaving far-edge resources (due to security or SLA breaches).
- minimal computing and other resources available on the participating site, which do not allow for running a federated manager or similar components.
- short, on-the-fly membership timeframes, during which the far-edge resource may share available services with the federation or consume federation resources.
- wildly diverging APIs and capabilities since said far-edge sites are typically constrained not only in terms of available resources but also cloud software and services that they can run.

3.3.2.2 Service Level Agreement

A critical mechanism to be revisited is the verification and guarantee of signed SLA for microservice-based applications that can be deployed in different cloud/edge and far edge devices. This includes a new architecture/platform for the CECC SLA [5] [6].

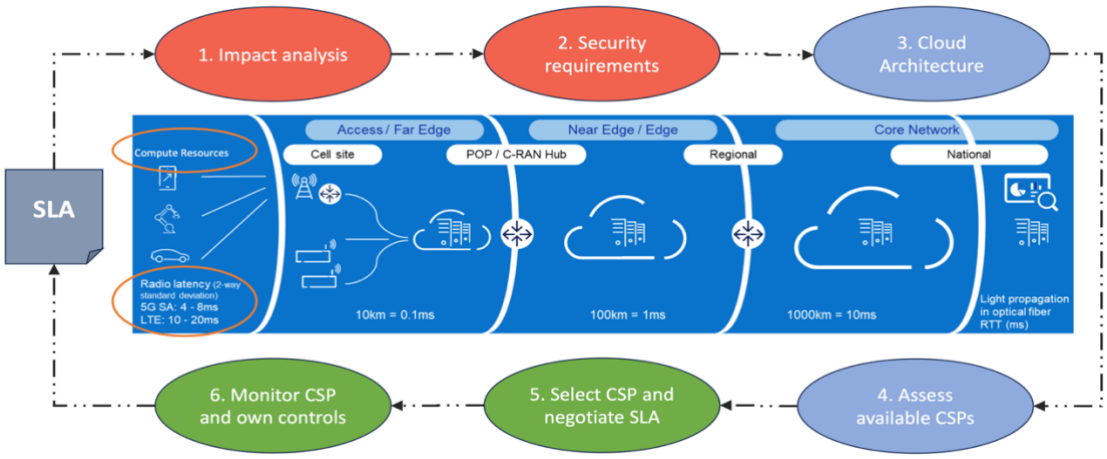

Figure 9: Far/Near Edge and Core Network SLA. Source: Dell Technologies [6].

Meeting SLA targets in a federated environment similar to the one depicted above presents two main challenges.

Firstly, NIST [7] has limited provisions for guaranteed SLA when implementing a federated environment. The authors mention in Section 3.7.2 that some services may need low latency support but do not propose a structured and thorough architecture to cover this requirement. Also, they provide a few examples in Section 3.7.2 about the selection of a specific region/location, but there is no proposal for how the deployment process could be implemented. Finally, Section 3.7.3 refers to an optional need for resource awareness; however, it does not describe which metrics could be crucial and how they can be used.

Secondly, the introduction of Far Edge resources poses additional challenges, namely mobility, temporary membership, varying network performance, and diverse hardware resources. These further increase the complexity of the SLA mechanism.

In general, the following list of SLA-related challenges needs to be considered and addressed by AC^3:

- Identification of SLA metrics relevant to the project and its applications. These include but are not limited to, uptime/availability, network latency (both between services and between microservice components), environmental restrictions (i.e., % of energy consumed from green sources), and throughput to far edge devices that generate data.
- Regional constraints introduced by the CECC applications. For example, use cases like agriculture, fire detection, and area security may explicitly require microservices deployment on far-edge devices within

the area of interest. In addition, data-intensive use cases like astronomy data, further analyzed in Section 4.3, may significantly benefit from edge deployment and meeting an SLA, since edge processing allows for reliable data processing in spite of varying network conditions as well as limits network utilization, since only processed data are sent over the (potentially unreliable) far-edge to near-edge network.

- Availability constraints are introduced by the unpredictable nature of the far-edge. These include, but are not limited to, varying network availability and performance, limited computing resources, and limited abstractions and features compared to generic servers available in the near edge and core network.
- Past/recent SLA trends of both far-edge and near-edge components. For instance, a drone flying over a somewhat small area may be preferable to a moving car unless the latter has been known to provide reliable network connectivity in the past. A brown datacenter may be preferred despite a stated preference for green energy if during subsequent cloudy days, the green datacenter has exhibited outages or failures to accept new deployment requests.
- Application SLA requirements. Ultimately, the application stakeholder determines the requirements, both at an application level as well as connected microservices. The AC³ lifecycle manager will need to take these into account and ensure that the SLA level is met while considering the challenges above.

3.3.2.3 Stateful Microservices

The deployment shift to container-based microservice model hosting poses additional challenges, particularly when the pods have storage needs. While it could be argued that the NIST framework [7] is broad enough to consider a federation of arbitrary resources, it only makes limited considerations for microservices. NIST is mostly oriented towards either VMs or higher-level abstractions like DBaaS or PaaS but does not overly consider network and data-intensive microservices. Additionally, data management in microservices adds new challenges: although microservices are supposed to work autonomously, they often end up exhibiting functionality and private state dependencies amongst each other [8].

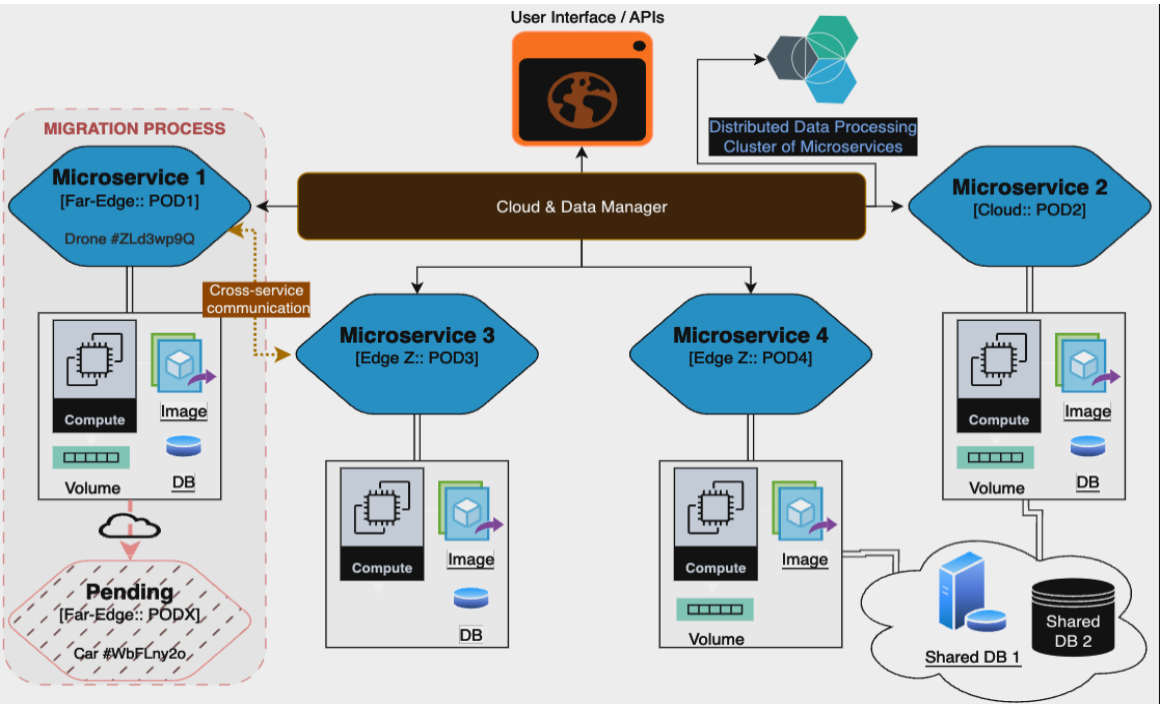

Figure 10: Examples of microservices with different dependencies.

Consider, for example, the hybrid deployment (cloud, edge, and far-edge) scenario depicted in Figure 10. Four

microservices are deployed in different resources of a CECC, and the Cloud & Data (Lifecycle) manager chooses to migrate Microservice-1 from one far-edge device (e.g., car) to another (e.g., drone) and allow faster processes of locally generated data. As shown, Microservice-1 has cross-service communication (e.g., API offering) with data dependencies (e.g., volume, DB, or shared DBs). Such challenges should be properly covered to allow the smooth transition/migration of a microservice without affecting service offerings.

In what follows, a detailed list of the challenges (such as a shared database as shown on the right side of Figure 10 raised by seamless service migration of stateful microservices:

- When microservices are migrated to a different pod, the registered volumes and databases must also be migrated along with the containers. In situations where large volumes of data are being migrated to an edge device or in long-distance migrations, additional expected hardware and network bottlenecks, such as available storage resources and network bandwidth, need to be considered prior to making a decision to migrate. Complex dependencies, similar to the ones illustrated in Figure 10 above between internal volumes, databases, and shared databases, need to be considered in a holistic manner since the end-to-end service can only be resumed after the complete migration of both the microservice, as well as its dependencies and linked dependencies is completed managements.
- Network stability, both in terms of availability as well as latency and throughput, is an additional challenge that needs to be considered for any environment, including but not limited to the CECC that includes Far Edge components.
- Energy management of far-edge devices, since any additional edge processing needs to be offset against the available energy capacity, which may be limited in the case of lightweight devices such as drones.
- Service Level Availability, since any migration constraint related to the unpredictable nature of far edge components (network stability, energy capacity, available compute resources) needs to be weighed against not only the relatively short migration window but also the longer period during which the service, or parts thereof, will be running at the far edge.
- Environmental considerations should weigh into any migration decision since optimizing energy consumption to favor “green” data centers is a well-defined goal of AC^3.
- Data replication, semantics and data consistency across a variety of storage technologies need to be guaranteed [8]. Different databases or distributed file systems support different metadata, configurations and APIs. The challenge is to design and structure mapping files in such a way that they can be used by the cloud manager as an intermediate layer agnostic of vendor-specific implementation.
- Online ad-hoc queries (manually applied by a user), real-time processing of data streams, event-driven computing (e.g., data pipelines), and cross-microservice data validations need conduction of cloud manager capabilities with data manager ones. Moreover, the cloud manager should continue to provide the same management features (e.g., microservice support, monitoring, logging, namespace, permission, etc.) and it should also provide all the distributed data resource management features (e.g., query execution DAG [9] [10], SQL support [11], optimization [10] [12], data streams, etc.). Due to the many requirements of increased complexity, robust design is a huge challenge.

3.3.2.4 Lifecycle management across non-uniform infrastructure

Another challenge not covered by NIST and many of the so-called “hyperscalers” such as Google Cloud Platform, AWS, and Azure, which already provide federated secure data infrastructures, is the integration of edge and far-edge resources belonging to end-users or (very) small organizations. This is a problem of such complexity that even the established hyperscalers require end-users and customers to run well-defined software platforms [13] [14], hardware platforms [15] or both, in order to accommodate a common lifecycle management experience across both their public cloud resources and on-premise infrastructure.

A solid lifecycle management experience is a key factor for the success of the AC^3 project, which will mitigate big

challenges like cloud costs, energy consumption, and communications latencies by moving the computation closer to the end-users and energy optimization. A number of these challenges are outlined in the previous sections and are related to the unpredictable short/mid-term future state of far-edge components, which is not currently described by NIST.

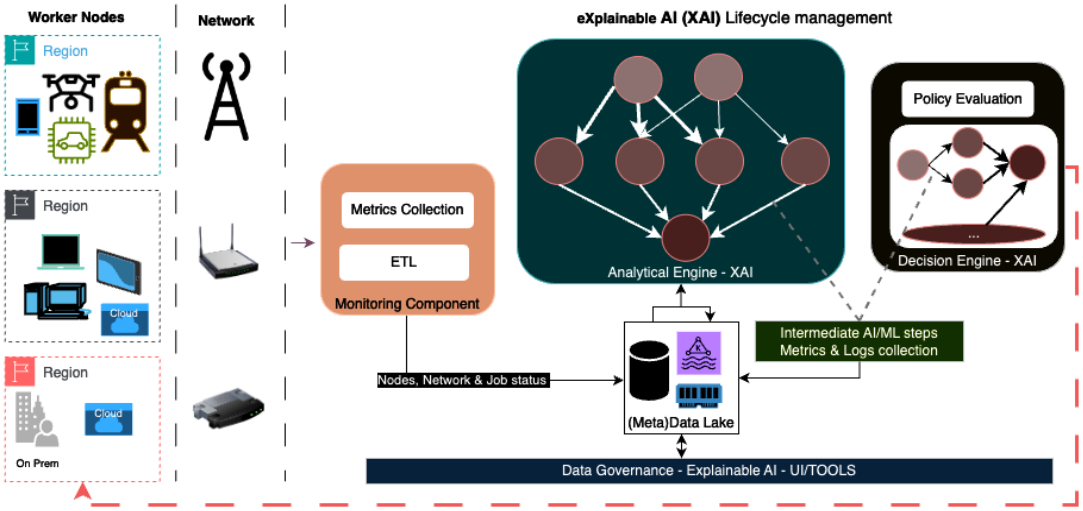

Figure 11: Proactive eXplainable AI (XAI) lifecycle management.

AC³ will be adopting forecasting capabilities and algorithms (AI/ML) so as to address the challenges posed by far edge devices. However, the Proactive eXplainable AI (XAI) Lifecycle management proposed by AC³ (see Figure 11) is a crucial process with many challenges by itself. Over the last few years, the academic community has been attracted to research the use of forecasting for better scaling/allocation of resources/microservices, but there is no standard/protocol that clearly covers its challenges:

- Increase model explainability
  - Reduce the ML algorithm complexity can make it explainable, transparent and understandable, but it also reduces the prediction accuracy. Finding a proper balance between the tradeoffs and/or proposing new techniques (e.g., intermediate steps) is not trivial, so additional research is needed.
- Automatically find which algorithm/model better serves each use-case. This need has received the attention of the research community over the last few years, but there is no generic solution available due to the many possible available combinations of the following parameters/features:
  - Data characteristics (e.g., traffic spikes, seasonality, etc.).
  - Cluster characteristics. (e.g., number of microservices, number of regions, edge and far-edge, etc.)
  - Network characteristics and bottlenecks.
- Identify new features derived from the integration with far-edge devices
  - As an example, the estimated remaining battery based on the expected CPU usage needs to be taken into consideration. The challenge is to identify and examine such kind of parameters.
- Advanced error handling
  - New corner cases derived from the integration with the far-edge devices need to be properly covered. As an example, if the AI resource manager decides to spawn a task in a far-edge device based on the estimated remaining battery, but the CPU usage has suddenly increased during the initialization of the task, the AI resource manager must be able to quickly identify if the task should be moved into another node, or if the remaining battery continues to be sufficient for the fulfilment of the task.

3.3.2.5 Data management and Federation

Data management and federation have received significant popularity in recent years. Apart from the enormous

research work, many well-established companies like Google, but also start-ups, base their products on data collection, processing, labelling, energy-efficient management, and federation. Apache [16] is consistently working on developing all the frameworks needed to compose a complete data platform. Also, Gaia-X is a federated and secure data infrastructure, which is supported by representatives from business, science, and administration in many EU member states.

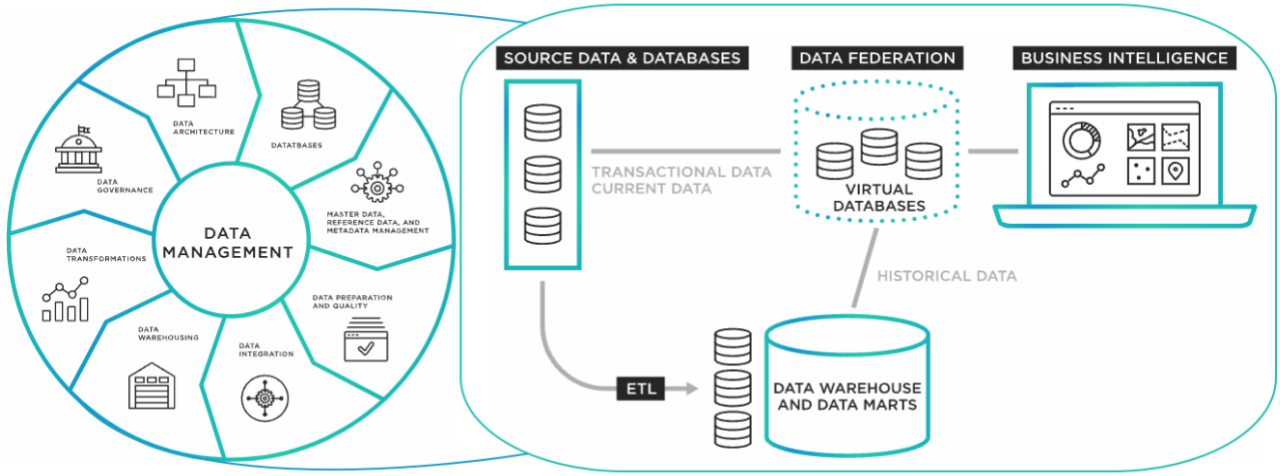

Figure 12: Architecture and offerings of Data Management and Federation [1].

The challenges of data management are huge since multiple heavy operations/jobs, data repositories, and live data streams (e.g., IoT sensors, monitor functions, etc.) need to be handled with robustness. Open-source or enterprise data projects like Apache Spark, Apache Delta-Lake, Apache Dremio, Airbyte, or Gaia-X need to be carefully selected or used as guides, and unlike these frameworks, the proposed data management platform should be integrated into CECCM and be the first management system which handles both cloud and data resources. Additionally, the integration challenges are many due to complex configurations with high regression risks and increased resource complexity (many microservices, networks, storage volumes, managements/processing services, etc.), which can easily lead to Out-Of-Service (OOS) state or low performance due to bottlenecks. A more detailed description is provided in the following list:

- Secure and authenticated data collection, data governance and sharing mechanisms across business domains. This allows for data use in operations, analytics and governance. The existence of far-edge devices raises extra challenges since it is expected to often connect/disconnect from the CECC and, thus are not stable members of the federation. This requires optimized authentication mechanisms, which can quickly register/unregister the devices without affecting the federation quality.
- Web interfaces and API servers, which provide easy access to cloud resources, and data, including streaming, transactional, structured and unstructured data as well as online queries [17]. Special handling is needed for the additional varying latency and connection drops introduced by the far-edge devices. For instance, when a data query is running in a distributed manner, and one of the nodes (e.g., a far-edge device) has a connection drop, then this microservice should be migrated to another node, which will process the specific partition of data, whereas, in the case of connection degradation (abrupt increase in latency or decrease in bandwidth), it should be decided whether it is better to wait for the response or migrate to another node and start the processing from the beginning.
- A scalable infrastructure that can evolve as business changes. Especially in cases of far-edge devices, quick and stable (un)registration is a high-priority item that needs to be covered by the data manager.
- The ability to work with existing, legacy, distributed, and monolithic technologies of different operating systems and programming languages. It is expected to have different hardware resources as different edge providers and different types of far-edge devices (e.g., cars, drones, etc.) will be connected to the platform.

Especially for the far-edge devices, they may have fewer hardware resources, run different architectures (ARM instead of x86) and have different or more restricted runtimes and frameworks.

- Cost and Energy reduction is always an open item, especially in cases of big-data where the cloud costs are huge and for far-edge devices where autonomy is critical.
    - Propose a management system that can handle both cloud and data services. Most of the data management [18] platforms are independent of the cloud management system, which slows down the deployment of data-driven applications over the cloud and the edge.
    - Find innovative solutions like smart selection between monolithic and distributed approaches per use-case.
    - Proper tuning of frameworks for optimal resource allocation and use of smart load balancers like Apache's Spark dynamic load-balancing.
    - Enable battery-saver capabilities for far-edge devices where the processing and energy will be wisely used.
- Automated data discovery, data evaluation, metadata, and feature extraction with big-data support. The design and proposal of generalized (auto)configurable distributed solutions of different inhomogeneous data sources (e.g., cloud, far-edge, etc.) raises even higher complexity, which has not been addressed in the past. Such mechanisms increase the quality of a data federation [1] system, as they enable rapid search for new data across multiple systems and devices as well as data evaluation and determinization of whether the collected data suit the use case without the need for manual steps. The development challenge is to implement multiple clients for the integration of all different cloud and data solutions available, as well as to automatically identify the schema of the data source before proceeding with the evaluation, metadata, and feature extraction. The complete system should also be well optimized to handle multiple different data sources smoothly.

3.4 How AC^3 addresses the challenges faced by current federation models

The AC^3 project addresses (among others) the challenges described in Section 3.3.2 above by **designing a novel CECC management (CECCM) framework**; for further details, please refer to deliverable "D2.1 1st Release of the CECC framework and CECCM".

This CECCM will play a critical role in supporting scalability and agility when managing the IT resources (i.e., computing, memory, and storage), as well as the emerging applications throughout the coverage scope of its system and infrastructure resources. The CECCM would consider three key dimensions to innovate and ensure the necessary agility for application execution and deployment, while optimally managing IT and network resources to ensure resource efficiency and energy consumption optimization.

1. A sophisticated application LifeCycle Management (LCM), which considers not only the application and its components lifecycle, but also the data source(s) that it relies on, and the SLA agreed with the application developer.
2. CECC IT components, including far-edge, and networking resource optimization.
3. A trusted resource federation to handle the lack of resources particularly at the edge, which is necessary to control any data deluge optimally. The CECCM would leverage state-of-the-art AI, ML, and semantic and context awareness algorithms when managing the application LCM, as well as cloud/edge/far edge and networking resources. It is critical for the CECCM to predict events and be proactive in order to optimize the IT and networking resources and guarantee application SLA. AI/ML algorithms would be used in combination with a scalable monitoring system considering the current load to serve and the amount of data needed by the application, enabling a proper prediction of the application resource needs.

4 Incentives and business interaction: a theoretical perspective

4.1 Introduction

The AC³ project builds on the emerging CECC paradigm, which advocates the integration of federated cloud and edge resources under a common management platform. Such a paradigm is enabled by the integration of diverse stakeholders. These stakeholders provide different types of resources or/and services and leverage different incentive mechanisms. Consequently, it becomes crucial to address the challenges associated with incentive dynamics among these stakeholders, particularly considering that some may operate within specific administrative domains delineated by geographical boundaries. Resource federation, in this context, offers benefits by facilitating the sharing of resources and services among stakeholders. For instance, when there is resource scarcity within a particular administrative domain, a stakeholder can request a federation of resources from another entity while agreeing on the service's SLA and pricing. Such incentive interaction needs to be well-defined and studied in order to ensure profit for both parties.

In the context of AC³, the global architecture encompasses a set of parties interacting with each other. The main stakeholders are identified as service providers, infrastructure providers (edge, far-edge, and cloud) and end-devices. Two primary types of business interactions emerge from this framework. Firstly, resource federation, which may be dependent on various criteria like the hardware/software capabilities of available servers, location/mobility, pricing, past collaborations, etc. Secondly, infrastructure providers also perform a decision-making process in selecting the service providers, which is based on performances, prices, location, etc. To model these interactions, research contributions have employed diverse techniques, including game theory, auction theory, and machine learning. This document begins by presenting a comprehensive analysis of current research contributions addressing incentive interactions among stakeholders in the context of CECC. Subsequently, the resource allocation problem is formulated from the perspective of multiple stakeholders, followed by the presentation of a solution to the formulated problem.

4.2 Related work

The integration of the edge computing paradigm enabled several use cases and applications and an important evolution of network performances. On the other hand, it created some novel challenges to be addressed by the research and industrial community. Challenges related to pricing, incentives, and economics in the field of Cloud and Edge Computing Continuum still lack more focus. The research community has addressed such challenges by leveraging several techniques, like Auction theory, Game theory, and Machine Learning.

Auction theory [19] is a popular economic approach that has been investigated to address the economic challenges of edge computing. Specifically, auction-based mechanisms are promising since they can fairly and efficiently allocate sellers' limited resources to buyers in a trading form at competitive prices. An ideal auction-based mechanism should ensure several desirable properties, e.g., truthfulness, budget balance, individual rationality, and economic efficiency [20]. In an auction theory model applied to edge computing, the sellers are the infrastructure providers, and the buyers (bidders) are service providers or user equipment (end devices). These entities interact in a bidding process to federate a certain commodity (resources, services, software, etc.) at a given price. In their work [21], authors addressed the challenge of maximizing social welfare in MEC systems. They formulate a problem of virtual machine allocation, where the buyers are end devices, and the sellers are the edge platforms. They leveraged the use of a combinatorial auction mechanism, called G-ERAP which is integrated with the combinatorial auction and the greedy algorithm.

Another technique leveraged in the context of multi-stakeholders is a double auction which is a popular method applied for typical many-to-many stakeholder scenarios. It is a technique investigated in [22], where a single-

round double auction mechanism is proposed. In the model, the end devices (buyers) with individual awareness and preferences compete for edge servers (sellers) with limited computation resources. The authors include a trustworthy third party that manages the whole auction process. The proposed solution considers the resource's location, resource allocation, and network economics in the decision process.

Alternatively, game theory and especially Stackleberg games has been heavily leveraged to model the incentive interactions between stakeholders. For instance, in their work [23], the authors addressed the challenge of designing a pricing schema in the context of IoT applications. The work considers the collaboration between edge, cloud and IoT devices. They model the problem as a dual Stackelberg game, where the market is composed of a cloud provider, a multi-edge infrastructure provider, and a set of end devices. In the proposed game, the cloud provider takes action first, and then the edge infrastructure providers decide on their strategies; based on these, the users define their strategies as well. The strategies of edge and cloud entities are the price of task execution in each domain. The decision strategy of end-devices is the actual payment as well as the dissatisfaction with service quality. To solve the problem, the authors rely on the use of a double-label radius k-nearest neighbors' algorithm (KNN) to filter out pricing schemes with a high success rate and boost system pricing efficiency. In the same context, the authors in [24] model the resource allocation in edge computing with a Stackleberg game. The end devices request resources to maximize their utility; on the other hand, edge providers provide resources to maximize their profit. The authors decompose into subproblems, wherein each subproblem, a type of resource, is considered. They solve the problem by using an iterative algorithm. Alternatively, the authors of [25] consider three dynamic pricing mechanisms in the context of an IoT-enabled edge computing environment. Namely, they consider a BID-proportional allocation mechanism, a uniform pricing mechanism, and a fairness-seeking differentiated pricing mechanism. They analyze these mechanisms in order to give edge computing service providers guidance on various kinds of pricing schemes. The authors of [26] introduced a novel sharing economy-inspired business model, following the trend towards a decentralized provision and sharing of digital resources. In their model, a platform facilitates the sharing of excess resource quota among users, leading to a more efficient usage of resources. Finally, [27] proposes a new market-based framework for efficiently allocating resources of heterogeneous capacity-limited edge nodes (EN) to multiple competing services at the network edge. By properly pricing the geographically distributed ENs, the proposed framework generates a market equilibrium (ME) solution that allocates optimal resource bundles to the services given their budget constraints.

In another context, machine learning, especially reinforcement learning (RL), proved to be very efficient in tackling problems in complex environments. While each RL agent can represent a stakeholder in the game, a multi-agent scenario might be suitable to maximize the fairness and utility of each actor. This has been experimented by the authors of [24]. The work considers a context of 5G networks, where virtual network functions are allocated in a multi-domain scenario. The authors propose an auction-based approach to allow inter-domain resource allocation. The proposed model incorporates a market composed of an infrastructure provider and several service providers. To solve this problem, they leverage a distributed multi-agent reinforcement learning solution.

4.3 Approach model for the Cloud Edge Computing Continuum

Following the literature review conducted in the preceding section, a noticeable gap is identified in terms of contextual focus on CECC. The majority of the previously mentioned works primarily address challenges from the perspective of infrastructure providers. However, within the AC^3 framework, our approach involves the integration of various stakeholders, including cloud providers, service providers, and far-edge infrastructure providers. In this section, we articulate a resource allocation problem that encompasses different stakeholder types. To model our problem, we employ an auction mechanism. In Figure 13, we present our envisioned system

model, which is composed of a set of cloud providers, edge infrastructure providers, and a set of users that access services through an access layer that can be heterogeneous. First of all, the infrastructure providers decide on the price of their resources and inform the CECCM as a trustworthy entity. Then, the service providers make bids on the proposed resources to the CECCM in order to satisfy their users' tasks and maximize their budget. In what follows, we describe in more detail our envisioned system.

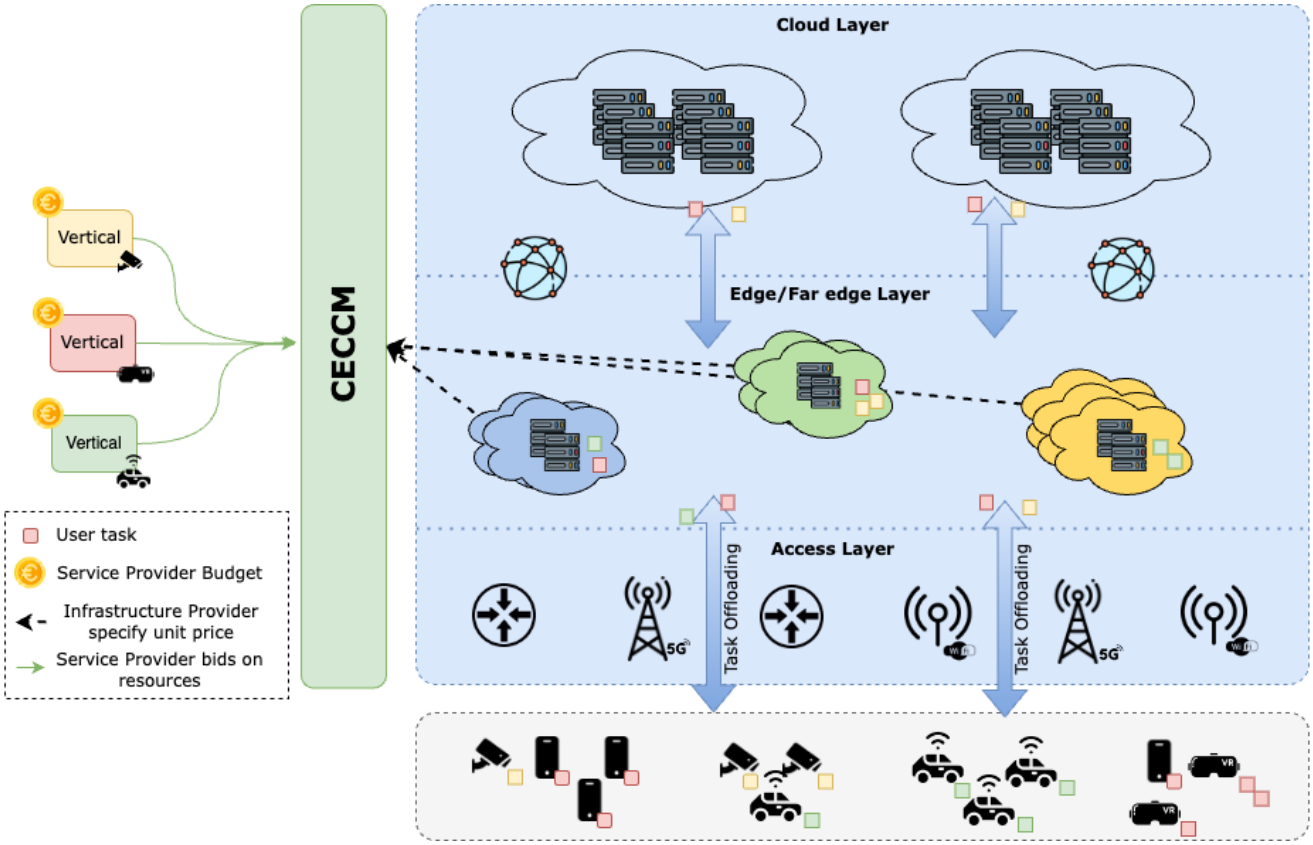

Figure 13. Business interaction between stakeholders, verticals and CECCM.

Let us envision a scenario with a collection of service providers, denoted as $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$, and a group of infrastructure providers, labeled as $\mathcal{F} = \{F_1, F_2, \ldots, F_n\}$ where N_s represents the number of service providers and N_F represents the number of infrastructure providers. In our scenario, a service provider is a vertical that provides services to its users. These services range from AR/VR applications to IoT data processing and autonomous Vehicles. Conversely, an infrastructure provider, whether cloud-based or edge/far-edge, possesses a pool of computing resources denoted as $compute_F$ and $memory_F$, which it allocates to service providers for executing tasks of their users. Infrastructure providers offer their resources in exchange for a specified price, with the pricing schema defined on a per-unit basis. For instance, P_c^F represents the price for a compute unit, while P_m^F denotes the price for a memory unit, both specified by infrastructure provider *F*. These unit prices are determined based on various factors such as resource availability and historical demand patterns.

We employ a temporal decomposition via a set of discrete time steps denoted as $\sqcup$, where each time step represents an iteration. Let us define a set of users as $\mathcal{U}$, assuming that each user can utilize only one service at any given time. The users of a specific service provided by service provider *s* are represented as $\mathcal{U}_s$, where $\mathcal{U} = \{U_s | \forall s \in \mathcal{S}\}$.

At the start of each iteration, the users of different service providers request the execution of certain tasks. These tasks can be executed in different places, mentioning a local execution (which is battery-consuming), edge/far-

edge, and cloud. The users specify their tasks in terms of the required amount of computing, memory, and latency to be executed for each task $T_i^u = (compute_i, memory_i, latency_i)$.

Each service provider enters the market with an initial budget *B*, aiming to maximize this budget and maintain market presence. To achieve this, service providers receive user requests for various tasks. They engage in a bidding process with different infrastructure providers to allocate resources for these tasks based on the prices proposed by the infrastructure providers. We assume that the service provider receives a certain amount of money after each iteration depending on the user's satisfaction; this is inspired by subscription models utilized by certain service providers. Similarly, infrastructure providers seek to maximize their budgets by offering resources at competitive prices.

To ensure trustworthy and reliable decision-making, our proposed solution relies on the Cloud and Edge Computing Continuum Manager (CECCM). The CECCM first receives the price units specified by different infrastructure providers and then informs the service providers. Then, it receives the bids of service providers, which allocate resources to their users. Based on these, the CECCM makes decisions and generates the appropriate configurations in order to assign tasks to each infrastructure provider. Additionally, the CECCM manages the budget received by both the service provider and the infrastructure provider.

To tackle the complexity of the environment, we propose employing a multi-agent reinforcement learning (RL) solution. Initially, we can model the service providers as RL agents whose goal is to maximize their budgets and consider that the infrastructure provider follows a greedy approach. Service providers must compete for resources by proposing higher bids while ensuring they remain within budget. Furthermore, we can extend this approach to include infrastructure providers, which also make intelligent decisions by leveraging RL agents.

In the field of reinforcement learning, defining an agent involves specifying its state, action, and reward functions. For service providers, the state comprises information about users' tasks, pricing from various infrastructure providers, and the current budget. The action space encompasses bidding information for selected infrastructure providers. The reward function for service providers aims to maximize their budgets and prolong their presence in the market.

Alternatively, infrastructure providers can initially be modeled using a simple price formulation to facilitate convergence of the service provider agents. The state space for infrastructure provider agents includes the current budget, past experiences (previous transactions), and resource availability. The action space involves setting price units for compute and memory. The reward function for infrastructure providers aims to maximize their budgets.

5 Use cases

5.1 Use-case 1: IoT and Data

Use Case 1 entails an IoT-based smart sensing and monitoring framework for infrastructures, harnessing the advantages of edge AI, to enhance the performance, reliability, and efficiency of various types of infrastructure, such as buildings, transportation systems, and industrial facilities. This framework combines IoT devices, edge computing, and AI algorithms to collect, process, and analyze data locally, reducing latency, improving real-time decision-making, and minimizing the need for constant cloud connectivity. Moreover, this functionality can seamlessly expand from basic prototypes to extensive deployments, incorporating autonomous agents like drones as crucial components for accessing remote installations and making onsite decisions.

5.1.1 Challenges

While an IoT-based smart sensing and monitoring framework with edge AI capabilities offers significant advantages to infrastructure managers, it also presents several challenges that need to be addressed for successful implementation and operation. The most important ones, relevant to the AC^3 project, include:

- **Sensor Integration and Calibration**: Integrating multiple sensor types and ensuring their accurate calibration can be complex. Different sensors may have varying measurement accuracies, calibration procedures, or communication protocols, requiring careful coordination and calibration procedures to ensure reliable data acquisition.
- **Data Integration**: Many infrastructures have legacy systems with existing data sources. Integrating data from these diverse sources into a unified monitoring framework can be challenging but is often necessary for comprehensive insights.
- **Edge Device Reliability**: Edge devices, such as gateways or servers, need to operate reliably regardless of their deployment conditions. Ensuring their robustness and fault tolerance is essential to maintain the system's uninterrupted operation.
- **Connectivity and Network Issues**: Ensuring reliable connectivity between sensors, edge devices, and the central system can be challenging, especially in remote or geographically dispersed locations. Network failures, signal interference, or limited coverage may affect data transmission and system responsiveness.
- **Data Management and Processing**: Dealing with 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 analysis can be demanding, particularly when dealing with diverse data formats and sources.
- **Data Security and Privacy**: Collecting and transmitting sensitive data from IoT sensors can raise security and privacy concerns. Protecting data from unauthorized access and ensuring compliance with privacy regulations (e.g., GDPR) is critical.
- **System Scalability**: As the infrastructure expands, ensuring scalability becomes essential. The system must be able to accommodate a growing number of sensors, handle increased data volumes, and support the addition of new functionalities without significant performance degradation.
- **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.

Addressing these challenges requires careful planning, robust engineering, and ongoing monitoring and management. However, with the right strategies and technologies in place, an IoT-based smart sensing and

monitoring framework can significantly improve the efficiency and reliability of infrastructure management.

5.1.2 AC³ proposed solution

Addressing the challenges associated with this use case requires a combination of technical solutions, best practices, and organizational strategies that will be provided by the AC³ CECCM. Regarding **Data Security and Privacy** AC³ will implement robust encryption and authentication mechanisms for data in transit and at rest, using secure communication protocols like HTTPS and MQTT with proper authentication and compliance with data protection regulations and anonymizing or pseudonymizing data as necessary. The AC³ CECCM will also provide mechanisms for redundancy and failover mechanisms (**reliability**) to ensure continuous operation using monitoring tools to maintain edge computing capabilities and cloud failover solutions. Also, the modular architecture of AC³, with containerization and microservices, will help us expand our system's capabilities to handle increased loads (scalability) and local analytics, deployed at the deep edge domain. The AC³ Data Management PaaS will act as a guarantee for streamlining the process of **data integration** using APIs for developing data connectors and standardized formats and protocols. Finally, the AC³ CECCM will help us achieve easy **maintenance** of our system, with automated software updates at all levels of our system.

5.1.3 Architecture

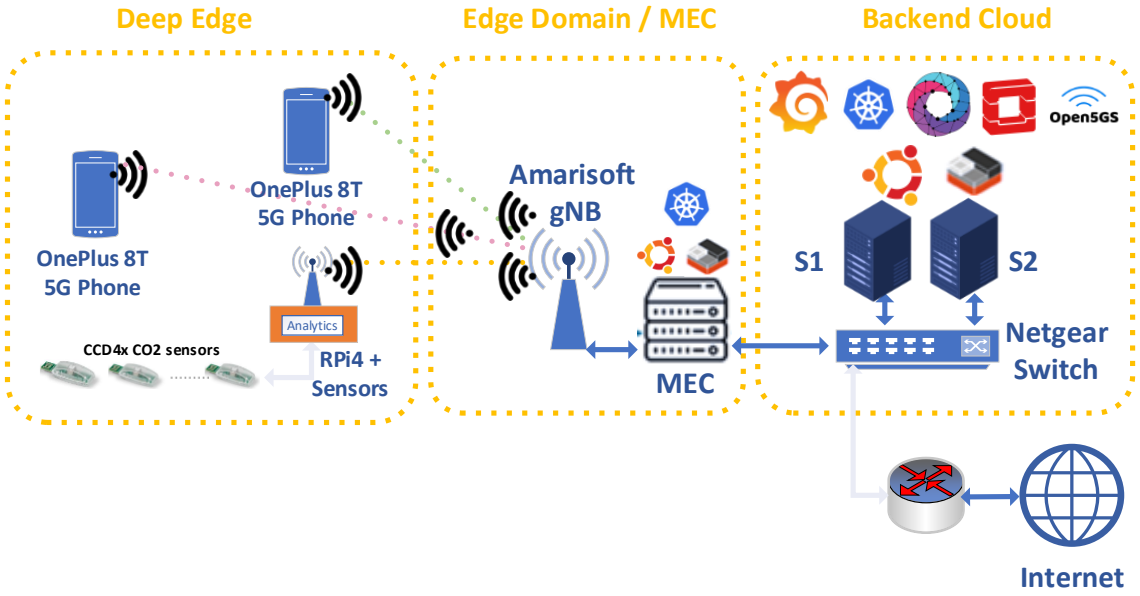

Figure 14: Architecture for UC1: IQU 5G / IoT testbed with Deep Edge deployment and local analytics.

The testbed architecture for this use case consists of a 5G network, as shown in Figure 14. The testbed for this use case consists of a 5G / IoT testbed with the following elements:

- Backend Cloud Domain: Two high-performance servers are deployed running Apache Kafka and Zookeeper services, as well as Open5GS, which is an open source 5G Mobile Core. The first server (S1) has an Intel Xeon Gold processor, with 26/52 cores running at 2.10GHz with 192GB of RAM memory and an ASUS ATX motherboard. While the second server (S2) has an Intel I9-10900L processor with 10/20 processor cores. The storage devices in S1 are: 1) one SATA SSD with 2TB of storage, and 2) a NVMe (Non-Volatile Memory Express) SSD disk with also 2TB of storage. The former is used to host the Ubuntu 20.04 filesystem and user-level data in two separate partitions. The latter is managed by ZFS [28], and it is used by LXD as a storage pool to allocate storage for all the VMs and containers that the server handles. S2, on the other hand, only has one NVMe SSD.

- An Edge Domain with Multi-Edge Computing (MEC) nodes: the Edge Domain consists of a 5G-compliant gNB and Shuttle PCs, each with a mobile Core i7 processor and 16 GB of RAM, acting as MEC nodes. The gNB currently in use is an Amarisoft Callbox Mini, which is suitable for experimental and testing environments.
- A Deep Edge domain, that consists of Raspberry Pi RPI4s enabled with SIM8200EA-M2 5G HATs providing 5G connectivity; WiFi can also be used as an alternative. The RPIs currently have Sensirion CO2 sensors attached to them via a Bluetooth Link.

The testbed, as described above, supports the following features:

- Orchestration capabilities: materialized through OpenStack and Opensource MANO (OSM) deployed in a virtualized environment.
- Monitoring System: deployment of multiple instances of the Monitoring System (MS) for data gathering and for deployment of the different enablers.
- Cloud-native Support: this is necessary to allow the flexible deployment of the enablers and to provide a federation of resources across the 5G Core Cloud and the 5G Edge Cloud Domain. A Kubernetes cluster is deployed to provide for this functionality.
- Deep Edge deployment of services; in the context of UC1 these are leveraged for performing analytics on the received sensor values, and detecting anomalies (e.g., CO2 spikes) while avoiding latencies inherent with cloud deployments.

5.1.4 Requirements

5.1.4.1 Functional Requirements

- **Scalability**: The application should be able to scale based on input and output volume of data and requests
- **API-Based**: The application should be designed to operate using well-defined APIs that allow the independent development of additional services.
- **Data Management**: The application should be able to store and retrieve data in a reliable and efficient manner from the deployed sensing infrastructure.
- **Calculations and Processing**: the software should be able to perform the required analysis on the collected data and generate the expected calculated outputs.
- **Security**: The application should incorporate appropriate security measures to protect sensitive data and ensure that users or other applications have the necessary permissions to access or modify data according to their roles and responsibilities.
- **Error Handling and Logging**: The application should handle errors and exceptions gracefully by providing informative error messages and logging relevant information for troubleshooting and debugging purposes.
- **Deep Edge deployment**: Lightweight k8s deployments (e.g., k3s) at the Deep Edge domain, that will support the deployment of local analytics functions (e.g., anomaly detectors).

5.1.4.2 Non-Functional Requirements

- **Performance**: The application should operate within the desired operational thresholds (response times, throughput, and resource utilization levels) under various conditions, such as normal usage, peak loads, or large data volumes.
- **Performance Efficiency**: The application should use system resources, such as memory, CPU, or network bandwidth, in an efficient manner, reducing latency and or minimizing energy consumption.
- **Availability**: The application should have minimal downtime.

- **Compatibility**: the application should be able to operate effectively with different hardware, software, and network environments.
- **Extensibility**: The application should be open to be easily updated with new functionality and hardware compatibility (e.g., new sensors).
- **Maintainability**: The application should be easily updated with minimal downtime and without the need for local interventions.
- **Cost**: The application should operate in a cost- and energy-effective way.
- **Modularity**: The application should be designed in a modular way to enable easy integration of new components or swapping of existing ones.

5.1.5 Key Performance Indicators (KPIs)

5.1.5.1 Application related KPIs

- Number of Integrated Sensor Types: 10 sensors - [Extensibility, Modularity].
- Edge to Cloud Data Volume Reduction of 50% [Performance Efficiency, Cost].
- Data Processing Latency < 50 milliseconds [Performance].
- ML Model Execution Time (Cloud vs Edge) < 20 % [Performance].

5.1.5.2 System Related KPIs

- Service Update Time (Cloud) < 180 seconds [Maintainability].
- Service Update Time (Edge) < 300 seconds [Maintainability].
- Service Migration Time (Edge to Cloud) < 120 seconds [Availability, Performance].
- Service Migration Time (Cloud to Edge) < 120 seconds [Availability, Performance].
- Application Availability > 99% [Availability, Maintainability, Performance].

5.2 Use-case 2: Smart Monitoring System using UAV

Use Case 2 creates a powerful and effective video surveillance and streaming system by combining IoT, camera, and UAV (Unmanned Aerial Vehicle) technologies. This use case supports both live streaming and video-on-demand (VoD) deployments. The video content would have been uploaded and proceeded previously in the on-demand deployments, whereby the video content and metadata would be stored for further use. The camera and IoT devices can be deployed on the ground or onboard the UAV to monitor various sensing data, such as CO, CO2, and passive infrared sensors (PIR). IoT devices and UAVs have been designed to enhance video surveillance capabilities by omitting blind spots in video surveillance applications. Cameras, UAVs, and IoT devices have distinct and heterogeneous computational capabilities that should be treated accordingly. While some cameras, IoT devices, and UAVs have higher processing power and are capable of on-board processing for cutting-edge functions, others have limited computing power to execute only limited processing while offloading the heaviest operations to the cloud and edge layers of the platform.

5.2.1 Challenges

Use Case 2 aims to provide an efficient IoT sensor data gathering system, live streaming, and video-on-demand (VoD) processing with customized data analytics. The main challenge of this use case is the computational and hardware heterogeneity and sensing capability of the IoT device. Due to the limited resources of some IoT devices, performing data processing is challenging, such as video transcoding and running deep learning techniques on sensors and video content. Providing network connectivity with a high data rate for sparse IoT devices and cameras is a challenging issue that hinders the ability to deploy the proposed use case on edge and far edge locations. The streamed video content should deliver a high-quality experience to the end users.

Moreover, managing the heterogeneity for performing various video content and IoT sensor processes is challenging. Efficient mechanisms for detecting the IoT device's resources and real-time capability are needed to decide whether to process the locally or offload to more powerful servers (Regional servers). Moreover, it is expected that the amount of generated video content and sensor data will be overwhelming, creating an extra overhead for the distributed devices and storage system. Finally, designing a customizable platform that can adapt and cohabit according to the end user's needs is another challenge. The platform should offer the end users the ability to specify their needs using a generic query language, and accordingly, different IoT devices collaborate to respond to the formulated request. Thus, the envisioned system should be designed to be a microservice-based platform with the ability and flexibility to use the resources as needed while reducing the final cost.

5.2.2 AC^3 proposed solution

To deal with the before mentioned challenges, we have suggested using an elastic architecture with high flexibility, whereby each component should be implemented as one or multiple microservices, each running a container on top of CECC. The following are the suggested components that should run as microservices:

- **Central Server:** This component's primary responsibility is to ensure the interaction between the clients and different components of this use case. This component consists of the following main sub-components:
    - **The front-end API:** This is the main component that is responsible for the interaction between the users and admins with all components of the systems. Thanks to this component, the users and admins can retrieve and manipulate information related to the users, video contents, sensor data, and regions. The users would be able to launch new live streams and upload or watch already recorded video content.
    - **The central server management system:** This component is opted with CRUD (Create, Read, Update, and Delete) functionality to ensure the user, camera, IoT, region, and notification lifecycle management. This system will be dotted with an SQL database (e.g., MariaDB) for storing all needed information. This component enables admins and users to manipulate, register or smoothly remove devices, users, regions, and notifications.
    - **The query processing system:** This component interacts with and empowers the front-end API, enabling users and admins to formulate their requests abstractly. This component allows users to launch live video streaming, stream VOD, and gather sensor data from different regions. Moreover, according to the specified inquiry, live video streaming and stream VOD can be joined with metadata and analytical data, such as object detection and tracking. Likewise, the system can provide performance analysis with the request, such as CPU and RAM utilization.
    - **The authorization and authentication service:** This component is planned to implement different user roles smoothly and efficiently. It will ensure the security and integrity of this use case by authenticating the users and admins and protecting the resources. Based on the user roles will have different authorization access. For instance, while some users can upload video content, others can only view those content. Not all users are allowed to launch data analytics on video content or sensor data. Finally, not all users are allowed to register new regions, IoT devices, cameras, or UAVs.
- **Regional Server:** This component is responsible for IoT devices and cameras in a specific area. This component mainly plays the role of reverse proxy or gateway for exposing all services provided by IoT devices and cameras located in a specific region. This strategy will enforce security and optimize public IP address utilization by allocating private IPs to IoT devices and cameras within a specific region. This component leverages Gstreamer for video content streaming and NVIDIA DeepStream for data analytics. It will stream the IoT sensor data and processed video content from IoT devices with high capability to

the end user via the frontend API. Meanwhile, this component offers local video content processing for constrained IoT devices and cameras. This component will process (e.g., data analytics and transcoding) the video contents on behalf of constrained IoT devices. For instance, a raw HLS (HTTP Live Stream) will be sent from a constrained IoT device to a regional server. The latter will process the video by transcoding and running a deep-learning model on the received stream by extracting and sending the requested metadata (e.g., Object detection and Object tracking). Direct communication between the regional server and the frontend API would alleviate the overhead on the central server. Finally, it will play a middle role in enabling federated learning of deployed deep learning models at different region servers and IoT devices. While the global model will be hosted at the central server, the local models will be deployed on regional servers and unconstrained IoT devices.

- **Camera and IoT devices:** This component will play the role of data provider by generating sensor data and video content. The latter could be streamed as a live stream or recorded for further use as VoD. We distinguish between two types of devices: constrained and unconstrained. To mitigate the network overhead, the unconstrained IoT devices will process the video content and sensing information locally. Thus, the video content and its metadata (e.g., data analytics) would be forwarded to the regional servers for those IoT devices. In contrast, only raw video content will be forwarded for constrained IoT devices. Similar to the regional server, unconstrained IoT devices will use Gstreamer for video content streaming and NVIDIA DeepStream for data analytics.

The CECC framework provided by the project AC^3 and its CECCM component will empower the second use case UC2 with different capabilities for ensuring the desired QoE. Thanks to the AI-based orchestration system provided by CECCM, the qualitative and quantitative KPIs of UC2 will be strengthened by enhancing various aspects of the smart monitoring system. These aspects include but are not limited to reducing the latency, improving reliability, enhancing the availability, and thus enabling more efficient management of the surveillance infrastructure. Through its advanced capabilities, the CECC framework will optimize the distribution of computing resources and processing capabilities, resulting in seamless integration of cloud computing and edge computing technologies. This integration will not only reduce latency and optimize bandwidth but also significantly improve the reliability, performance, scalability, and flexibility of the surveillance system. Moreover, thanks to service relocation and offloading capabilities, the CECC framework will empower UC2 with high resiliency and better resource computation.

5.2.3 Architecture

The picture below depicts the general architecture of the use case. Different users (e.g., ordinary users and admins) will interact with the system via the central server frontend API. A new microservice will be launched in different regions for registering new cameras or IoT devices, as well as launching new video streams or uploading new content. Different streams and content processing will be launched at different regional servers and IoT devices when users formulate their requests via the query processing system. Direct communication will be established between various regional servers and the user to avoid the overhead on the central server. An authentication and authorization system that ensures the security and integrity of the content will be established between different components in the use case. Users only get content or services with dedicated permission to ensure data privacy. The content would be processed locally at IoT devices and cameras or offloaded to the regional servers according to their processing capability.

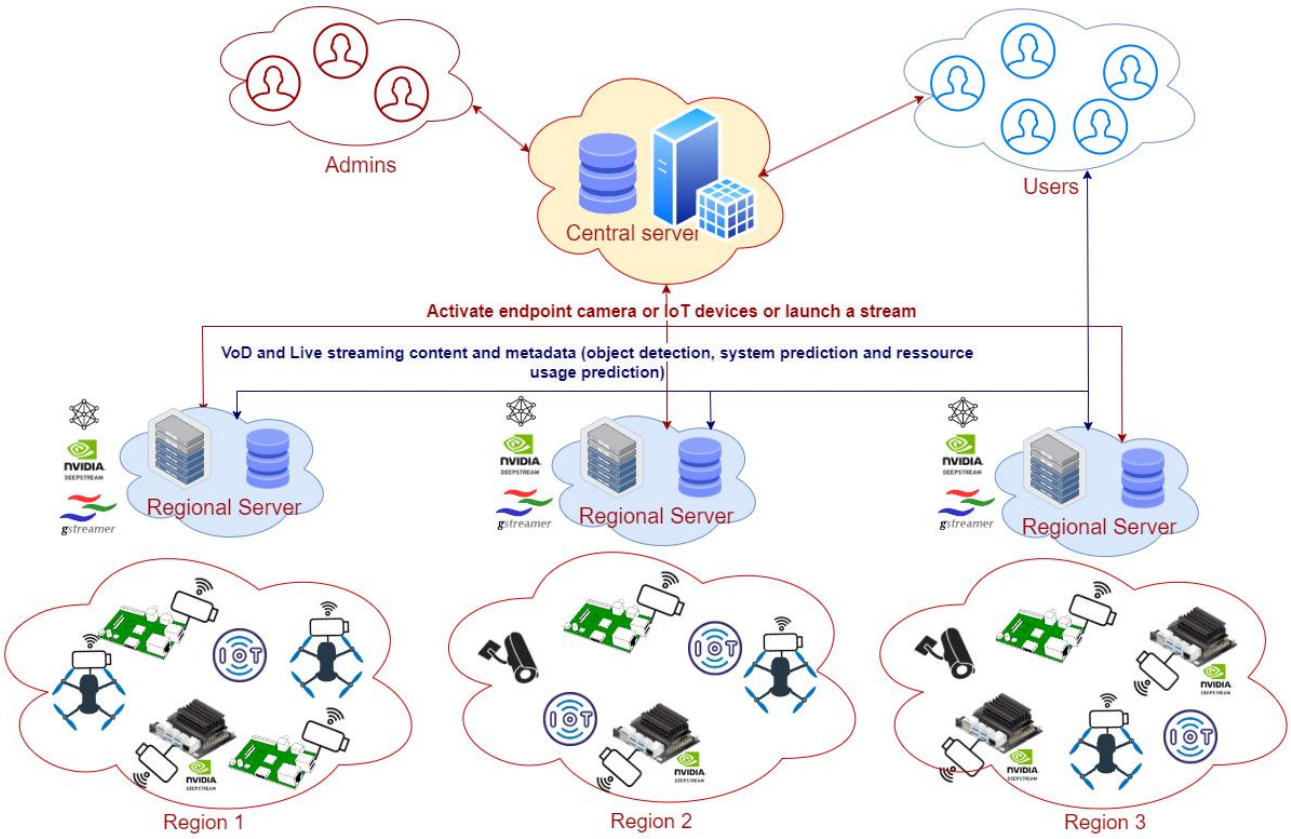

Figure 15: Testbed for UC2: A smart monitoring system using UAVs.

5.2.4 Requirements

This use case will leverage the CECCM to run and deploy the central and regional servers. Moreover, it is expected to get information on the CECC infrastructure regarding used and available resources. The monitoring information will help the use case for providing a high-quality experience for the end users by deploying or duplicating services as needed. The main requirements of this use case can be summarized as follows:

5.2.4.1 Functional Requirements

- **Customizability:** The services should be designed with high customization, allowing the users to specify their needs smoothly and efficiently.
- **Network connectivity:** Network connectivity should be ensured smoothly between different components to ensure the excellent functionality of the use case.
- **Computation resources:** The CECC should provide enough computation resources for RAMs, CPUs, and GPUs to allow the execution of heavy tasks handled by regional servers.
- **Scalability:** The CECCM should provide all the required APIs for registering, launching, or deleting microservices to enable elasticity when managing regional and central servers.
- **Network bandwidth:** The network bandwidth between cameras, IoT devices, regional servers, and end users should be enough for streaming video content and sending metadata and sensor data.
- **Storage:** The storage in the system should be enough for keeping all the generated contents in terms of VoD, sensors data, and metadata.

5.2.4.2 Non-Functional Requirements

- **Availability:** The system should be able to run even if there is some deviance in some IoT devices or cameras.
- **Time latency and jitter:** The end-to-end network between different components should be designed to provide network jitter and low latency.
- **Secure network connection:** The connection between different equipment is encrypted to ensure confidentiality.
- **Data replication and backup:** The File system should be designed to ensure the data replication and backup to ensure that the generated contents can be retrieved in case of storage defiance.

5.2.5 Key Performance Indicators

5.2.5.1 Application-related KPIs

- A number of Integrated IoT Devices and Cameras: Target a total of 100 devices for initial deployment, with the scalability to add more as needed.
- Data Processing Latency: Aim to keep data processing latency under 50 milliseconds for real-time responsiveness.
- Quality of Service in Video Streaming: Strive for 98% uptime for live and VoD streams, with a resolution of at least 720p and buffering time under 2 seconds.
- Accuracy of Analytics and Detection Algorithms: Target an accuracy rate of 95% for object detection and other analytics, with a recall rate of at least 85%.

5.2.5.2 System-related KPIs

- Service Update and Migration Times: Aim for service updates on cloud servers to be completed within 5 minutes and migrations on edge servers within 10 minutes.
- Network Bandwidth Utilization: Maintain a network utilization rate of 70-80%, ensuring efficient data transfer without overloading the network.
- System Availability and Uptime: Target an overall system availability of 99.5% or higher.
- Data Throughput Rate: Set a benchmark for a data throughput rate of at least 1 Gbps, ensuring efficient handling of high-definition video and sensor data.

5.3 Use-case 3: Deciphering the universe: processing hundreds of TBs of astronomy data

5.3.1 Challenges

Use Case 3 deals with the distributed processing of 3D datacubes. A 3D datacube is a multidimensional dataset that combines spatial and spectral information, and it is saved in a FITS format. This three-dimensional array contains two spatial dimensions that represent a specific region of the celestial sphere containing the targeted galaxy while the third dimension represents the wavelength or frequency range. 3D datacubes allow for spatially resolved analysis which provides information about the physical processes occurring at different locations within an object. Despite all the benefits, working with 3D datacubes introduces additional dimensions and complexity in comparison with traditional 1D spectra. There are several main challenges that are associated with both (1) the increased complexity and volume of the data, as well as (2) the specific requirements associated with the analysis. The main challenges associated with point (1) include: data volume, significant storage capacity, additional computational resources and longer processing times. Regarding point (2), the key challenges are the following: data exploration and interpretation, parameter estimation, and interdisciplinary collaboration to

analyze 3D datacubes (i.e., astronomers, computer scientists, etc.) to advance in the treatment and analysis of these data.

5.3.2 AC³ proposed solution

The AC³ project plays a vital role in supporting UC3 through its specialized architecture, tailored to handle intricate data processing tasks, particularly those involving datacubes. AC³'s modular architecture, combined with containerization, streamlines the management and deployment of analysis tools dedicated to processing datacubes. Each tool can be encapsulated within containers, ensuring scalability. This architectural design helps with the integration of memory management strategies, software optimizations, and distributed computing techniques. By utilizing container orchestration platforms, AC³ can dynamically allocate computing resources according to workload demands, optimizing the computational resources for processing large volumes of datacubes. To deal with the above-mentioned challenges, the proposed solutions are as follows:

- Memory management: the large number of spectral elements and spatial pixels that datacubes contain require efficient management of the memory. The strategies that will be developed in the context of the AC³ project to overcome memory limitations consist of data compression, data binning, and the use of techniques that include parallel processing to accelerate the computations and optimize the processing time.
- Computational resources and data volume storage that are needed due to the size of the datacubes, or more specifically, when dealing with a large number of datacubes (i.e., working on a particular survey or a specific observational program). We propose to use high-performance computing (HPC) systems or distributed computing techniques to handle the computational load. Parallel processing (as mentioned in the previous point) and distributed computing frameworks, such as the use of Kubernetes or GPU acceleration, would help to speed up the analysis.
- Software optimizations that include both efficient algorithms and software tools are proposed to deal with data exploration, interpretation, and parameter estimation. 3D datacubes have higher dimensionality compared to 1D spectra which makes the number of free parameters increase, this leads to complex optimization problems in terms of both parameter estimation and model fitting.
- Scalability of existing software. The spectral synthesis software that will be used within the AC³ project to help in the analysis and modeling of the datacubes (such as the pPXF - Penalized PiXel-Fitting method - program) needs to find the optimal parameter values while handling degeneracies and ensuring the convergence of the fitting process. We will optimize the use of various spectral software programs to apply them specifically to the datacubes used within the context of the AC³ project. We propose to decompose the original 3D datacubes into small cubes. Then, we would apply programs such as pPXF, STECKMAP, and/or STARLIGHT on each of them to perform the analysis. Finally, we reconstruct a new 3D datacube (the same dimensions as the original one). The use of these software assets tends to be computationally expensive and requires efficient memory management. We will apply practices such as data chunking or parallelization and complex optimization solutions.

5.3.3 Architecture

The Use Case 3 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 which their 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 AC³ project to perform this task are the following:

- STECKMAP (STEllar Content and Kinematics via Maximum A Posteriori [29]), 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 [30][31] uses an optimization algorithm to adjust the weights applied to each spectral template to minimize the difference between the modeled spectrum and the observed spectrum. Common optimization algorithms include techniques based on χ^2 minimization or maximum likelihood estimation.
- pPXF (penalized PiXel-Fitting method, [32]) a code developed in Python to extract the stellar kinematics and stellar population from absorption-line spectra of galaxies, using a maximum penalized likelihood approach.

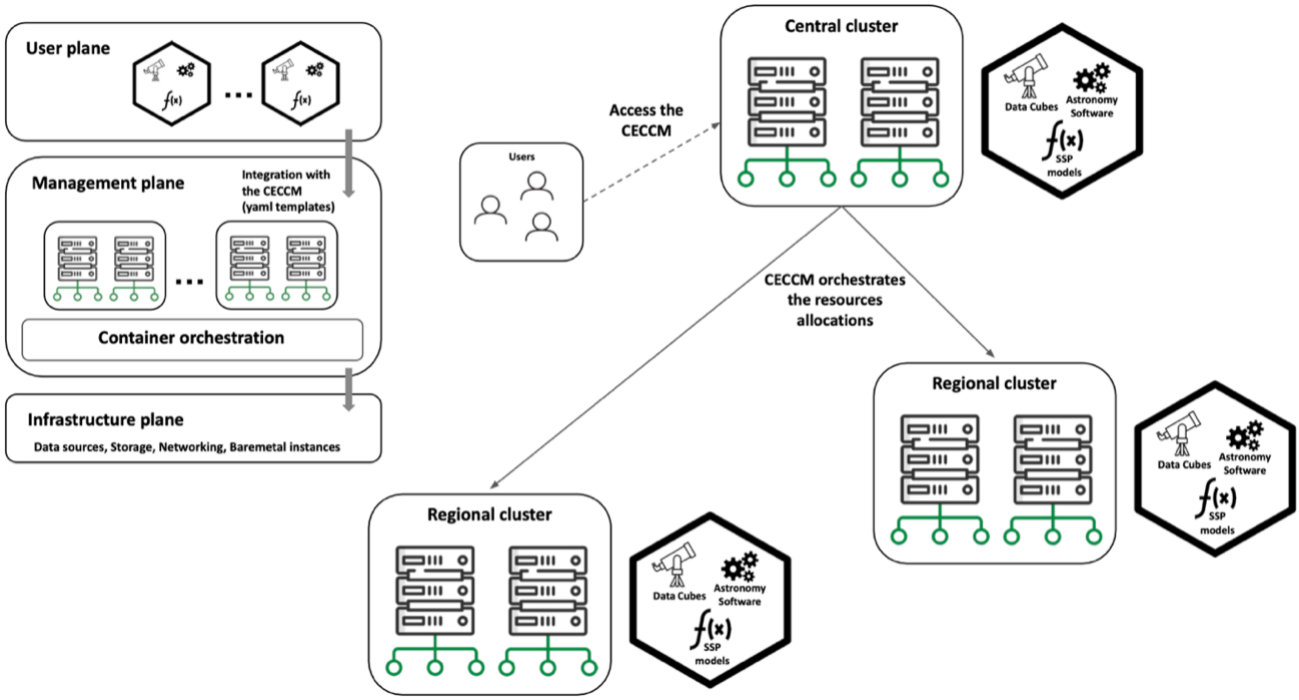

Figure 16: Architecture of UC3.

Figure 16 below describes the overall architecture of Use Case 3 summarizing the main architectural components and how they interact. The astronomy software is modeled as a microservice user application based in Linux containers. The users will interact with the CECM to request access to these applications. The CECCM schedules the execution of the applications and the availability of the required resources (datasources, networking, storage, computing). The execution of the astronomy software might be federated into different regions depending on the overall system load and the requirements. The UCM domain will have a local view of the data

and will allow to deploy the local placement of the software in charge of processing the data. Once the software finishes its execution, the result can be stored in the provisioned storage resources. In general, use case 3 will be composed of one or more domains, where there must be availability for both the astronomical software assets and the datasets to be submitted as CECCM jobs. Figure 16 also shows a set of API endpoints that will be advertised to interact with all the CECCM services, components and layers, allowing to process the astronomical datacubes. An API will be exposed to the CECCM to deploy additional microservices across the cloud or edge infrastructure. This API will be part of the use-case Local Management System so it can be consumed by the CECCM.

5.3.4 Requirements

The main requirements needed for use case 3 could be described as follows:

- Consume the infrastructure management components to allow optimizing data-intensive applications.
- Reduce the end-to-end execution time and maximize the use of local bandwidth.
- Consume the CECC capabilities to orchestrate the data sources and applications execution.
- Develop a microservice-oriented application to distribute software and data across the federated infrastructure.

5.3.4.1 Functional Requirements

- Scalability: The application should be able to scale up or down as per the needs of the requirements without any performance degradation.
- Elasticity: The application should be able to handle the fluctuation in traffic and adapt accordingly.
- Resiliency: The application should be able to recover from failures, and it should be highly available.
- Microservices Architecture: The application should be designed in a modular way, with each module performing a specific function.
- Continuous Delivery: The application should support continuous delivery, with the ability to automatically deploy new features and updates.
- Multi-tenancy: The application should support multiple tenants or users, with each user having access to their own data and resources.

5.3.4.2 Non-Functional Requirements

- Performance: The application should be highly performant, with low latency and fast response times.
- Availability: The application should be highly available, with minimal downtime or disruption to users.
- Reliability: The application should be reliable, with a low rate of failure and data loss.
- Maintainability: The application should be easy to maintain and update, with minimal downtime or disruption to users.
- Flexibility: The application should be flexible and able to support changes in business requirements or new features.
- Cost: The application should be cost-effective, with the ability to scale up or down based on the needs of the business.

5.3.5 Key Performance Indicators

5.3.5.1 Application-Related KPIs

- Computational Efficiency: Minimize the processing time needed to execute the spectral synthesis code per galaxy in terms of the spatial and spectral dimensions (same configuration). Expected achievement: reduced by at least 50% in comparison with a standalone computing node. We assume that the pre-processing and the post-processing times of the computed spectral data is not considered.

5.3.5.2 System-Related KPIs

- CPU usage: Minimize the Percentage of the CPU and memory usage relative to the system´s capacity when processing galaxies of different sizes during the spectral fitting. Expected achievement: utilize less than 80% of available CPU and memory resources.
- Guarantee high availability of the application => KPI: 99% reliability.

5.3.5.3 Additional measurement metrics

- Minimize the whole computing time given a specific galaxy with different configurations: number of models to be loaded and different signal noise. Increase 10 times the SSP models used. Expected achievement: reduced by at least 10% in comparison with a standalone computing node.

6 Conclusions

This document provides the first basis for the work to be done in AC³, following the decision to have a mixed business model (IaaS, PaaS, SaaS), including changes in most of the layers (Figure 2). The main stakeholders have been identified, clarifying the approach to be followed as some services/products will rely on third-party providers, and interactions with main functions among the consortium are described.

As part as well of the techno-economic analysis, challenges facing current federation models were analysed, making emphasis on NIST and Gaia-X understanding which give us the best practices to be followed, choosing NIST for the general part of the federation but considering the directrices of Gaia-X for Data handling. More detailed information on the reasons is covered in deliverable D2.1 "1st Release of the CECC framework and CECCM".

The pre-defined use case requirements went through a reassessment, and adjustments were performed as deemed fit to adjust with technological advances and market evolution, ensuring feasibility and relevance. Each of the use cases reviewed provides its KPIs that will be checked during PoCs to be developed in the testbed covered in WP5 to ensure the objectives of the project are reached.

Most of the objectives of task T2.1 have been covered, leaving for the next version of deliverable D2.1 "1st Release of the CECC framework and CECCM" the incentives for stakeholders and end users. With this part, all the objectives will be met.

7 References

[1] «Tibco,». Available: https://www.tibco.com/reference-center/what-is-a-data-federation.

[2] P. C. N. a. A. R. Kraemer, «Gaia-x and business models,» 2023.

[3] R. Bohr, C. Lee y M. Michel, "The NIST Cloud Federation Reference Architecture," *Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD,* nº 500-332, 13 February 2020.

[4] Gaia-X European Association for Data and Cloud AISBL, "Gaia-X Architecture Document - 23.10 Release", 31 October 2023. Available: https://docs.gaia-x.eu/technical-committee/architecture-document/latest/.

[5] J. e. a. Luna, de *"Leveraging the potential of cloud security service-level agreements through standards." IEEE Cloud Computing 2.3* , 2015, pp. 32-40.

[6] DELL Technologies, "What is Happening in the Network Edge", 26 June 2023. Available: https://infohub.delltechnologies.com/p/what-is-happening-in-the-network-edge/.

[7] C. A. R. B. B. a. M. M. Lee, "The NIST cloud federation reference architecture 5." NIST Special Publication 500 (2020): 332,» 2020.

[8] R. e. a. Laigner, "Data management in microservices: State of the practice, challenges, and research directions." arXiv preprint arXiv:2103.00170, 2021.

[9] "Hadoop apache", Available: https://hadoop.apache.org/docs/stable/hadoop-yarn/hadoop-yarn-site/YARN.html .

[10] "Spark by examples", Available: https://sparkbyexamples.com/spark/what-is-dag-in-spark/ .

[11] "Data Bricks", Available: https://www.databricks.com/glossary/catalyst-optimizer..

[12] "Flink Apache" Available: https://flink.apache.org/..

[13] "Microsoft Azure Stack solutions," Available: https://azure.microsoft.com/en-us/products/azure-stack.

[14] "Google Kubernetes Engine (GKE) Anthos Technical Overview" Available: https://cloud.google.com/anthos/docs/concepts/overview.

[15] "AWS Outposts Family", Available: https://aws.amazon.com/outposts/.

[16] "Projects Apache", Available: https://projects.apache.org/projects.html?category#big-data.

[17] P. S. D. M. a. G. P. Junior, "Stateful container migration in geo-distributed environments." IEEE International Conference on Cloud Computing Technology and Science (CloudCom),2020.

[18] "Tibco", Available: https://www.tibco.com/reference-center/what-is-data-management.

[19] V. Krishna, "Auction Theory", Academic Press, 2002. .

[20] H. Q. e. al., "Applications of Auction and Mechanism Design in Edge Computing: A Survey", *IEEE Transactions on Cognitive Communications and Networking,* p. 1034–1058, 2021.

[21] Y. C. e. al., "A Stackelberg game approach to multiple resources allocation and pricing in mobile edge computing", *Future Generation Computer Systems,* p. 273–287, 2020.

[22] W. S. a. J. L. Yanlin Yue, "A Double Auction-Based Ap- proach for Multi-User Resource Allocation in Mobile Edge Computing", *14th International Wireless Communications Mobile Computing Conference (IWCMC).*, 2018.

[23] T. W. e. al., "EIHDP: Edge-Intelligent Hierarchical Dynamic Pricing Based on Cloud-Edge-Client Collaboration for IoT Systems" *IEEE Transactions on Computers ,* p. 1285–1298, 2021.

[24] M. D. e. al., "Market Driven Multidomain Network Service Orchestration in 5G Networks", *IEEE Journal on Selected Areas in Communications,* p. 1417–1431, 2020.

[25] B. B. e. al., "Three Dynamic Pricing Schemes for Resource Allocation of Edge Computing for IoT Environment", *IEEE Internet of Things Journal* , p. 4292–4303, 2020.

[26] M. S. e. al., "Dynamic Pricing for Resource-Quota Sharing in Multi-Access Edge Computing," IEEE Transactions on Network Science and Engineering, p. 2901–2912, 2020.

[27] L. B. L. a. V. B. Duong Tung Nguyen, "Price-Based Resource Allocation for Edge Computing: A Market Equilibrium Approach", *IEEE Transactions on Cloud Computing ,* pp. 302-317, 201.

[28] "Cloud Google", Available: https://cloud.google.com/bigquery.

[29] C. Pappalardo, "Galaxy Evolution through spectral fitting tools: A comparative study between STECKMAP and FADO. Uncovering Early Galaxy Evolution in the ALMA and JWST Era", Proceedings of the International Astronomical Union 2020.

[30] R. Cid Fernandes, “Resolving galaxies in time and space. I. Applying STARLIGHT to CALIFA datacubes”. Astronomy & Astrophysics, Volume 557, id.A86, 2014.

[31] J. Ge, “Recovering stellar population parameters via two full-spectrum fitting algorithms in the absence of model uncertainties”. Monthly Notices of the Royal Astronomical Society, Volume 478, pp. 2633-2649, Issue 2, 2019.

[32] M. Capellari, “Improving the full spectrum fitting method: accurate convolution with Gauss-Hermite functions”, Monthly Notices of the Royal Astronomical Society, Volume 466, Issue 1, pp. 798-811, 2023.