Security and Trust Management in Cloud Edge Continuum: AC^3 project approach

Sofiane MESSAOUDI*, Abd-Elghani MELIANI*, Ayoub MOKHTARI*, Adlen KSENTINI*
*Eurecom Institute
*Sophia Antipolis, France
Email: *name.surname@eurecom.fr

Abstract—**In the rapidly evolving world of computing, resource management within the cloud-edge continuum needs to be efficient and secure. The Agile and Cognitive Cloud Edge Continuum (CEC) management (AC^3) project addresses this need by devising a novel architecture to orchestrate and manage micro-service-based applications over the CEC infrastructure. AC^3's architecture features three planes: the *User plane* for Application Developers (ADs), the *Management plane* for application and resource orchestration, and the *Infrastructure plane* for dynamic infrastructure, enabling Cloud Edge Computing Continuum Manager (CECCM) owners with autonomy and emphasizing data excellence through its Data Management (DM) Platform-as-a-Service (PaaS). This paper leverages the AC^3's framework by focusing on security and trust mechanisms within the CECCM. It examines the interactions between ADs, infrastructure providers (InfPs), and CECCM, highlighting AC^3's comprehensive approach to robust security and trust within the CECC ecosystem.**

I. Introduction

In today's digital landscape, the seamless integration of cloud, edge, and far-edge is essential for real-time data processing, low-latency interactions, and optimal user experiences. This integration forms the basis of a federated computing continuum, efficiently harnessing centralized cloud resources while leveraging the immediacy of edge devices.

The Agile and Cognitive Cloud edge Continuum management (AC^3) project addresses the challenges of this integration in the deliverable D2.1 [1] with a layered architectural model consisting of a User, Management, and Infrastructure Planes. Furthermore, the proposed architecture introduces the Cloud Edge Computing Continuum Manager (CECCM), a key component that manages and orchestrates the Cloud Edge Continuum (CEC) infrastructure.

Each plane comprises components facilitating intra-plane and inter-plane communication: *The User Plane*, the user-facing facet, it focuses on enhancing interactions between Application Developers (ADs) and the CECCM. *The Management Plane*, central to the AC^3 framework, it oversees application and resource management, ensuring the efficient lifecycle of applications and the Cloud Edge Computing Continuum (CECC) infrastructure. *The Infrastructure Plane*, hosts essential elements like data sources and computing nodes, guided by federation principles inspired by National Institute of Standards and Technology (NIST) [2] and Gaia-X [3] models.

Indeed, the AC^3 framework manages micro-service-based applications across cloud, edge, and far-edge resources. It introduces CECCM, which interacts with ADs and Infrastructure Providers (InfPs) to manage the application lifecycle (i.e., from their definition up to their deployment) and resource allocation. This architecture also handles data integration as Platform as a Service (PaaS) and selects InfPs to host microservices composing an application through a federation system.

In a complex ecosystem such as AC^3, efficient and secure mechanisms are crucial to protect transactions and communications among actors. However, while robust security mechanisms are crucial for protecting data and interactions within the AC^3 framework, establishing Trust among the various stakeholders is equally important. Trust is essential in a federated environment [4] [5] where multiple independent entities collaborate. It ensures that InfPs adheres to agreed-upon service levels and that application developers can rely on the integrity and reliability of the services offered.

For the security part, the goal is to enforce zero-trust security and ensure authorization, authentication, and encryption for all communications between components, including the internal CECCM components and external entities (i.e., ADs and InfPs).

Additionally, the paper introduces a trust model for Service Level Agreement (SLA) management, utilizing Blockchain and Smart Contracts to verify and guarantee signed SLA. This includes a third-tier entity collecting feedback from the ADs, Key Performance Indicators (KPIs) from CECCM, and InfPs to detect SLA violations and build InfP reputation.

In summary, this paper introduces two main contributions to the cloud edge continuum architecture of the AC^3 project:

- A zero-trust security for the CECCM with robust protocols granting access, communication, resource, and data security within the federation and revoking access if security breaches are observed. It also secures the communication channels and protects

- the CECCM from external attacks through its exposed interfaces to the public (i.e., ADs and InfPs);
- A trust model to verify and guarantee signed SLAs for building InfPs reputations using Blockchain and Smart Contracts.

The rest of the paper is structured as follows: Section II provides a background of the AC^3 architecture. Section III offers an overview of the security approach and explores the advanced security architecture of AC^3. Section IV focuses on the trust overview and architecture. Finally, Section V concludes the paper.

II. Background

In this section, we will recall and summarize the AC^3 architecture as introduced in deliverable D2.1 [1].

A. *AC^3 Overall Architecture*

The convergence of cloud, edge, and far-edge technologies is transforming modern computing, driven by the need for real-time data processing, ultra-low latency, and optimized user experiences across various applications. This convergence aims to create a federated computing continuum, combining the strengths of centralized cloud resources with the immediacy of edge devices, facilitating efficient data flow and processing. The innovative European project AC^3 [6] redefines federated computing with a novel architecture that enhance the integration of *edge computing* with *centralized cloud resources* to reduce latency and improve real-time decision-making.

The AC^3 high-level architecture manages the lifecycle of microservice-based applications on a federated CEC infrastructure. At its core is the CECCM, which uses Artificial Intelligence (AI)/Machine Learning (ML) techniques to handle application lifecycles and optimize the CEC infrastructure, considering energy consumption and SLA requirements. This section summarizes the functional architecture, encompassing the User, Management, and Infrastructure Planes, as shown in Figure 1. Each plane has components serving specific functions and roles within the ecosystem. Furthermore, AC^3 also emphasizes Data Management (DM) as a foundational PaaS component, covering data retrieval, storage ,monitoring and semantic reasoning.

1) User Plane: The User Plane serves as the interface for ADs, encapsulating CECCM functionalities through the Application Gateway (AG). It provides a simplified environment for developing, deploying, and managing microservice-based applications, with intuitive interfaces for Create, Read, Update, and Delete (CRUD) operations. Application descriptions and SLAs are expressed as intents, translated into machine-readable formats such as Yet Another Markup Language (YAML) or JavaScript Object Notation (JSON), and then forwarded to the Management Plane.

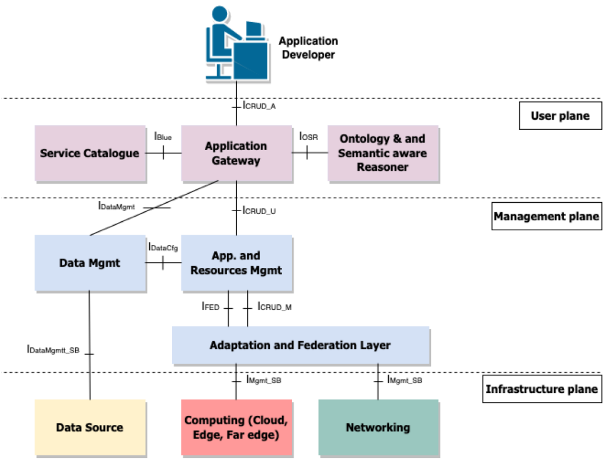

Figure 1: High-Level Architecture of AC^3

2) Management Plane: Central to CECCM, the Management Plane orchestrates applications and manages CEC resources through AI/ML algorithms. It ensures the efficient lifecycle of applications while overseeing the CECC infrastructure. Energy conservation and strategic positioning of the Management Plane are pivotal, highlighting AC^3's commitment to sustainability. DM within AC^3 adheres to Gaia-X [3] and Institute of Electrical and Electronics Engineers (IEEE) Standard for Intercloud Interoperability and Federation (SIIF) [7] specifications, ensuring data security and governance. The abstraction and federation layer streamlines infrastructure layer CRUD Application Programming Interface (API) operations, enhancing efficiency.

3) Federated CECC Infrastructure Plane: The Federated CECC Infrastructure Plane aggregates resources from public/private clouds, edge, and far-edge. It combines Gaia-X elements for interoperability and data sharing with NIST's model for efficient cloud resource management. Trust among AC^3 stakeholders is achieved through linked data representation and verifiable credentials via Gaia-X compliance web portal and trust anchors. The Federation Hosting Service (FHS) (a module from the IEEE SIIF architecture) facilitates interactions between the CECCM and federated resource infrastructures, allowing owners to manage their resources autonomously.

B. *CECCM Key Components*

Application Gateway (AG): Provides Graphical User Interface (GUI) and API interfaces for CRUD operations, translating human intents into machine formats.

Ontology and Semantic-aware Reasoner (OSR): Utilizes semantic web technologies to interpret and adapt policies, optimizing the application lifecycle.

Service Catalogue (SC): Central repository for application blueprints and metadata, supporting CRUD operations and tracking ownership.

Application and Resource Management (ARM): Uses AI to manage the application lifecycle and optimize resources. It includes modules for AI-based Life-Cycle Management (LCM), monitoring, application/resource profiling, and decision enforcement.

Data Management (DM): Manages access to hot and cold data, following Gaia-X procedures for data lakes and Internet of Things (IoT) data sources.

Adaptation and Federation Layer (AFL): Serves as an intermediary between CECCM management functions and the Local Management System (LMS) on the federated infrastructure, facilitating resource discovery and management.

This section has provided an overview of the AC^3 architecture, detailing the core components and their roles in managing microservice-based applications across a federated CEC infrastructure.

III. Security in AC^3

In this section, we provide an overview of the security proposal in AC^3, detailing its architecture and workflows.

A. Overview

Securing a multi-service architecture like AC^3 is challenging due to its three planes: User, Management, and CECC Planes. Each plane's components and interfaces increase the attack surface. Furthermore, internal components communicate via remote calls, each with entry points, increasing susceptibility to attacks. Consequently, security must be robust across all components.

Since distributed security screenings can degrade performance due to repetitive checks and remote connections, we propose adopting zero-trust principles to ensure secure access, minimize privileges, and enforce stringent authentication and authorization processes. Zero-trust networking, introduced by John Kindervag [8], exemplified by Google's BeyondCorp [9] and guided by NIST SP 800-207 [10], mandates secure communication irrespective of network location. It shifts access controls from the network perimeter to individual users, enabling secure operations without traditional Virtual Private Networks (VPNs).

Additionally, sharing user context across services in a multi-service environment requires explicit context passing, unlike monolithic applications that use common sessions. To facilitate this, JSON Web Tokenss (JWTs) [11] are used for secure cryptography transmission of user attributes between microservices.

1) Requirements and Proposed Approach: In what follows, we will recall the key security principles:

Authentication: Identify the requesting party to prevent spoofing of a system (e.g., a service) or a user.

Integrity: Ensure data has not been altered during transmission through signing and secure communication channels like Transport Layer Security (TLS), HTTP Secure (HTTPS), Message Queuing Telemetry Transport (MQTT), and Simple Object Access Protocol (SOAP) depending on the specific requirements of the microservices application.

Non-repudiation: Provide proof of data origin and integrity, often using digital signatures.

Confidentiality: Protect sensitive information using encryption, access controls, and secure storage.

Availability: Ensure system accessibility to legitimate users, defending against Denial-of-Service (DoS)/Distributed Denial-of-Service (DDoS) attacks.

Authorization: Determine actions an authenticated user can perform, enforced at the overall application entry point (i.e., AG) and within individual microservices.

By adhering to these principles, we aim to provide a secure environment for managing microservice-based applications across a federated CEC infrastructure.

B. Architecture

In this section, we propose several improvements to the AC^3 initial architecture, shown in Figure 1, following key criteria to create an efficient and secure system aimed at achieving a zero-trust security model. The CECCM has external and internal communications that need to be secured. External north/south communications involve interactions between the CECCM, the ADs and the Northbound Interfaces (NBIs) of the federated infrastructures (i.e., LMSs), while internal (east/west) communications consist of interactions between CECCM components. This is achieved bu relying on the security mechanisms specified by the NIST model. It's noteworthy that the security of communication among federation actors is out of the scope of AC^3.

A leveraged version of the AC^3 architecture featuring security is illustrated in Figure 2, which includes novel components for a zero-trust security architecture. These components are the Secure Gateway, Identity Provider/Authentication Server, and Security Policies Administration, whose functions and roles will be detailed in what follows.

1) Secure User-CECCM Communications: The Secure Gateway, depicted in Figure 2, acts as a Policy Enforcement Point (PEP), authenticating, authorizing, and intercepting all end-user requests to enforce access control policies. It ensures that only trusted users can access the AG API. For end-user authentication, approaches include certificate-based authentication [12] and Open Authorization (OAuth) 2.0-based access delegation [13]. The Secure Gateway authenticates OAuth 2.0 security tokens accompanying each API request, representing both the application and the user who granted access.

Additionally, the Secure Gateway enforces authorization by applying access control policies at the service level. The Identity Provider/Auth Server produces tokens or certificates for a user. After verifying the connection integrity, the Secure Gateway forwards requests with

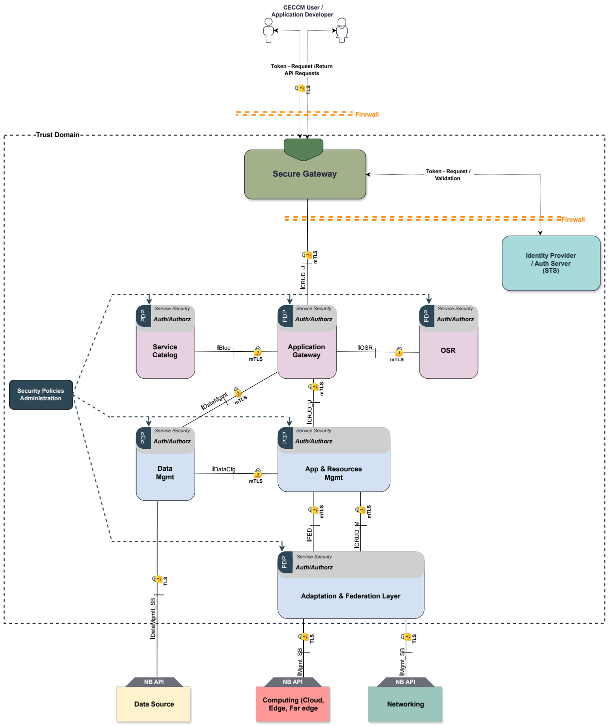

Figure 2: Security Architecture Overview in AC^3

user context to the AG API, then redirects them to the appropriate CECCM service. To do so, options include passing the user context in an HyperText Transfer Protocol (HTTP) header or creating a JWT containing the user data. While the HTTP header is straightforward, it raises trust concerns. Using JWT provides confidence that the content remains unaltered, as the issuer of the JWT signs it. Additionally, communication between the Secure Gateway and the AG API requires mutual TLS (mTLS) authentication to ensure channel security.

Figure 3 illustrates the secure communication workflow between the AD and the CECCM. When a developer deploys an application within the CECCM-managed infrastructure, they provide the application's definition to the CECCM and submit their credentials through the secure gateway. The secure gateway interacts with the identity provider to generate a user certificate and obtain a JWT token. It then forwards the request, including the user context, to the user plane for enforcement.

2) Secure CECCM Service-Service Communications: Ensuring authentication and authorization between services is essential to minimize vulnerabilities and security threats. For authenticating inter-CECCM components communications, approaches include trusting the network, mTLS, and JWTs. Trusting the network is less suitable for our case. Instead, we adopt a zero-trust network approach, assuming a hostile network environment where every request must pass authentication and authorization.

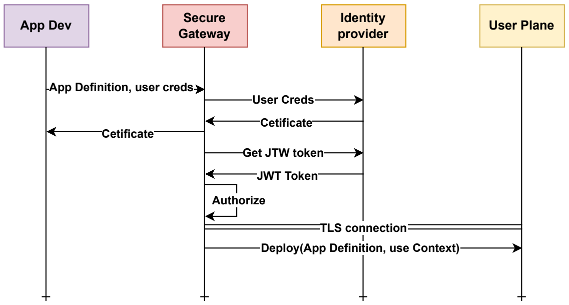

Figure 3: Secure Communication Workflow between the AD and the CECCM

In fact, mTLS is widely adopted for securing service-to-service interactions. Each service uses a public/private key pair for authentication. Alternatively, JWT operates at the application layer, transporting a set of claims between services. In our scenario, we adopt a hybrid approach, combining mTLS for encryption and authentication with JWT for transmitting essential information like user details or authorization levels. This ensures confidentiality, integrity, and mutual authentication.

For service-level authorization, the Policy Decision Point (PDP) within each CECCM component stores policies centrally defined at the Policy Administration Point (PAP) and evaluated locally. Services receive policy updates from the centralized PAP via events, ensuring up-to-date access-control policies.

The Secure Communication Workflow between CECCM Services, depicted in Figure 4, emphasizes the importance of mutual authentication. In this workflow, the application gateway requests blueprints from the service catalog by presenting its certificate. The service catalog, in turn, validates the sender's identity and authorization level before fulfilling the request. This process ensures that all interactions between internal services are secure and authenticated, maintaining the integrity and confidentiality of the communications.

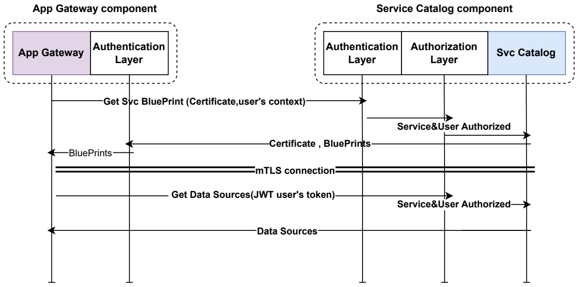

Figure 4: Secure Communication Workflow between CECCM Services

3) Secure CECCM-Federated Infrastructure Communication: In a typical deployment of multi-service software, multiple trust domains are common. When the CECCM needs to establish connections with North Bound (NB) APIs of LMS within the federated infrastructure, LMSs needs certificates from a trusted public certificate authority to ensure authenticity. The CECCM Adaptation and Federation component can encrypt a secret key using the LMS public key for further communication tunnel encryption, as illustrated in Figure 5.

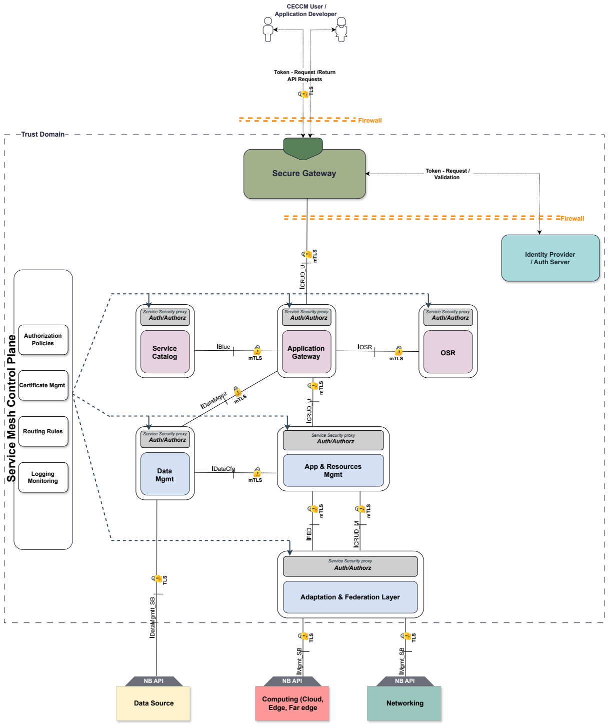

Figure 5: Security Architecture Instantiation for CECCM Microservices based Implementation in AC^3

To secure communications between the CECCM and various LMSs, the adaptation agent initiates a secure connection with an LMS for application on-boarding. The LMS provides a certificate from a trusted public certificate authority, ensuring authenticity. A secure session is established for interaction with the LMS's NBI endpoint (see Figure 6).

IV. Trust in AC^3

This section is dedicated to the trust overview and model in AC^3.

A. Overview

To enhance trust among stakeholders providing CECC infrastructure, AC^3 incorporates trust management functionalities into its architecture. This involves creating trust profiles for InfPs within the federation. These profiles assess the providers' reliability in supporting

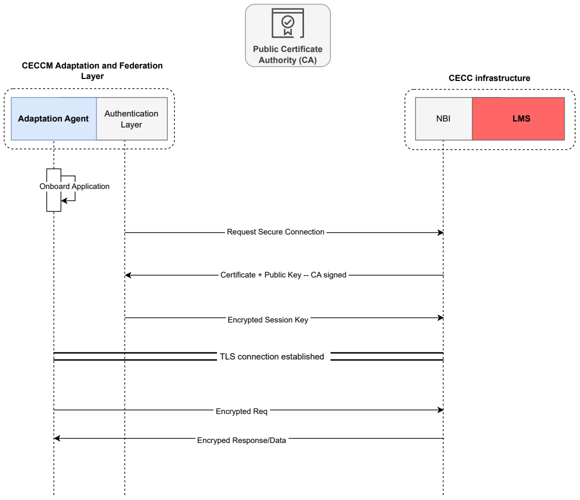
Public Certificate Authority (CA)
CECCM Adaptation and Federation Layer
Adaptation Agent
Authentication Layer
CECC infrastructure
NBI
LMS
Onboard Application
Request Secure Connection
Certificate + Public Key -- CA signed
Encrypted Session Key
TLS connection established
Encrypted Req
Encryped Response/Data

Figure 6: CECCM – Infrastructure LMS Secure Communication

and validating the SLAs between the CECCM, InfPs, and ADs.

As outlined in D2.1, ADs define the application components, including images, configurations, and SLAs, which specify expected performance parameters such as service availability, link capacity, and latency. Indeed, SLAs meticulously delineates parameters governing the expected performance of microservices. These parameters cover critical aspects such as throughput requirements, scalability, security provisions, and communication channels. SLAs serves as a comprehensive framework, fostering collaboration between service components and InfPs.

The structure of an SLA includes elements such as the period of validity, involved parties, template, service types, parameters, guarantees, billing, and termination conditions. By defining clear expectations, responsibilities, and terms, SLAs establishes a robust foundation for collaboration and accountability among stakeholders. Figure 7 shows the structure of an SLA.

B. Architecture

To ensure a trustworthy infrastructure, a comprehensive trust architecture is proposed in Figure 8. This architecture introduces the Trust Manager, alongside existing CECCM components, aimed at deriving the reputation of InfPs. The Trust Manager comprises several key modules:

KPI Monitoring: The KPI Monitoring module collects monitoring data regarding SLA performances from involved actors, including the CECCM, InfPs, and ADs. This data is crucial for detecting violations and ensuring compliance with SLAs.

SLA Monitoring: The SLA Monitoring module utilizes collected data to detect SLA violations auto-

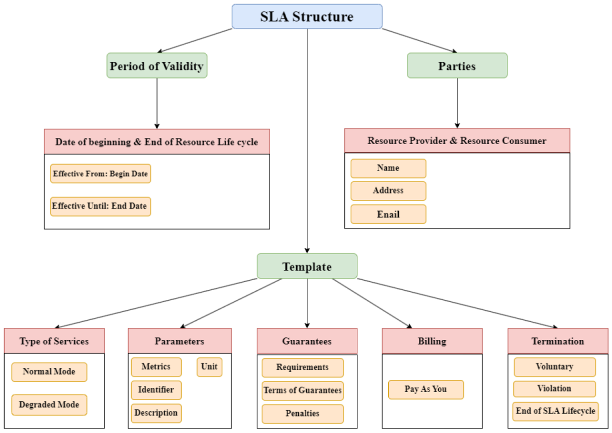

Figure 7: Service Level Agreement Structure

matically. *Smart contracts* are central to this process, establishing and enforcing SLAs with efficiency and precision. These self-executing contracts encode predefined terms directly into code, automating the enforcement of SLA conditions. By operating on principles of transparency, security, and automation, smart contracts ensure compliance with SLAs and enable real-time monitoring and enforcement.

Feedbacks: The Feedback module collects end-user feedback about service experiences. This feedback gathered periodically or after application use, provides valuable insights into user satisfaction and helps improve service quality.

Trust Management Module: The Trust Management module plays a vital role in deriving the reputation values of InfPs. It integrates data from the SLA Monitoring and the Feedback modules to assess the performance and reliability of each provider. These reputation values are then published on a *blockchain* for dynamic updating and transparency.

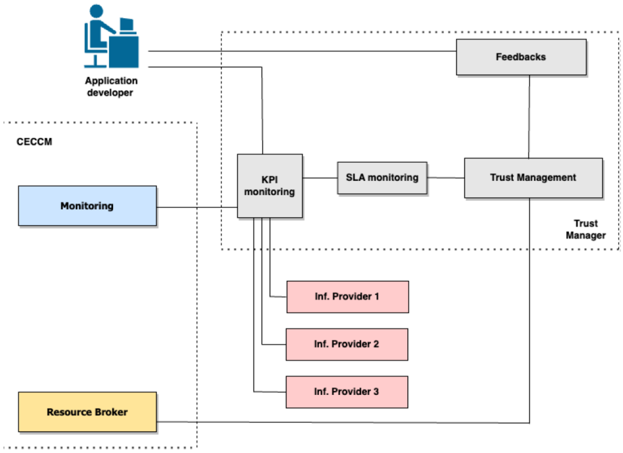

Figure 8: Trust Model Overview in AC^3

The CECCM deploys micro-service components by selecting InfPs based on resources, cost, and provider reputation, considering SLAs. The Resource Broker, a key CECCM component, handles the selection of InfPs from the federation. A notable innovation of AC^3 is the separation of resources from application management, allowing the CECCM owner to deploy applications on CECC infrastructure without ownership. This federated resource model enhances flexibility and trust in the deployment process.

V. Conclusion

In summary, AC^3 project redefines federated computing by integrating robust security and trust mechanisms across cloud, edge, and far-edge. The CECC framework and the CECCM combine centralized cloud resources with edge device immediacy, enhancing resource availability and system resilience. This PaaS for DM supports seamless application development and deployment. Our paper explores security and trust management within the CECC ecosystem, presenting methodologies to safeguard data and interactions. The AC^3 project lays the groundwork for future innovations, ensuring a secure and resilient CECC.

Acknowledgment

This work was partially supported by the European Union's Horizon Research and Innovation program under AC^3 project and grant agreement No 101093129.

References

[1] AC^3 partners, "D2.1: 1st release of the cecc framework and ceccm," 2024. [Online]. Available: https://ac3-project.eu/wp-content/uploads/2024/02/AC3_D2.1.pdf

[2] C. A. Lee, R. B. Bohn, and M. Michel, "The nist cloud federation reference architecture 5," *NIST Special Publication*, vol. 500, p. 332, 2020.

[3] A. Braud, G. Fromentoux, B. Radier, and O. Le Grand, "The road to european digital sovereignty with gaia-x and idsa," *IEEE network*, vol. 35, no. 2, pp. 4–5, 2021.

[4] S. B. Saad, A. Ksentini, and B. Brik, "An end-to-end trusted architecture for network slicing in 5g and beyond networks," *Secur. Priv.*, vol. 5, no. 1, 2022.

[5] ——, "A trust architecture for the SLA management in 5g networks," in *ICC 2021 - IEEE International Conference on Communications, Montreal, QC, Canada, June 14-23, 2021*. IEEE, 2021, pp. 1–6.

[6] Accessed: 2024-05-30. [Online]. Available: https://ac3-project.eu/

[7] Y. Demchenko, M. X. Makkes, R. Strijkers, and C. De Laat, "Intercloud architecture for interoperability and integration," in *4th IEEE International Conference on Cloud Computing Technology and Science Proceedings*. IEEE, 2012, pp. 666–674.

[8] J. Kindervag, S. Balaouras, K. Mak, and J. Blackborow, "No more chewy centers: The zero trust model of information security," *Forrester, March*, vol. 23, p. 18, 2016.

[9] R. Ward and B. Beyer, "Beyondcorp: A new approach to enterprise security," *; login:: the magazine of USENIX & SAGE*, vol. 39, no. 6, pp. 6–11, 2014.

[10] S. D. Young, "Moving the us government toward zero trust cybersecurity principles," *Memorandum M-22-09*, 2022.

[11] W. K. A. N. Dias and P. Siriwardena, *Microservices security in action*. Simon and Schuster, 2020.

[12] Accessed: 2024-06-04. [Online]. Available: https://www.ibm.com/docs/en/mas-cd/maximo-monitor/continuous-delivery?topic=devices-certificate-based-authentication

[13] Accessed: 2024-06-04. [Online]. Available: https://oauth.net/2/