101196630 – ESTIMED - STF 685

D6.3

Introductory White paper

Dissemination Level: PUBLIC

Version 1.0
13 May 2025

PROJECT	
Project number:	101196630
Project acronym:	ESTIMED
Project name:	Enabling Standardized IoT deployments in MEC Environments for advanced systems
Project starting date:	01/12/2024
Project duration:	36 months

Deliverable No	Deliverable Name	Work Package No	Lead Beneficiary	R	Dissemination Level	Due Date
D6.3	Introductory White paper	WP6	ETSI	R — Document, Report	Public	31-05-2025

Description

- This white paper explores the synergy between oneM2M and ETSI MEC, providing insights into technical foundations, industry applications, and policy implications. By bridging these standards, the paper highlights the importance of oneM2M and MEC interworking in standardized IoT deployments and its transformative impact on connected systems.
- This is the outcome of T6.1.

Co-funded by the European Union

Co-funded by EFTA

TABLE OF CONTENTS

ABOUT THE AUTHORS

JaeSeung Song *Sejong University, oneM2M Technical Plenary Vice Chair*	**Walter Featherstone** *Apple France, ETSI ISG MEC Vice Chair*
Muhammad Umair Khan *xFlow Research, Research Engineer*	**Marco Picone** *UniMore, Associate Professor*
Dario Sabella *xFlow Research, ETSI ISG MEC Chair*	**Alexandr Tardo** *CNIT, Senior Researcher*
Massimo Vanetti *Digital SME, oneM2M WG1 Chair*	**Andreas Kraft** *JK Consulting, Principal Enterprise Architect*
Joachim Koss *JK Consulting, STF685 Leader*	**Ayesha Ayub** *xFlow Research, Assistant Manager*

EXECUTIVE SUMMARY

Internet-of-Things (IoT) and edge computing are key enablers for the digital economy, with high potential in the industrial sector where 5G networks offer low latency and bandwidth efficiency advantages. The interoperable design of edge IoT systems can allow clusters of devices and agents to interact with each other to achieve common goals, e.g. using swarm computing techniques. These deployments require unifying standards from the edge IoT space: ETSI ISG MEC (Multi-access Edge Computing) and oneM2M.

In June 2023 ETSI White Paper [1], experts from MEC, oneM2M and SmartM2M described how the ETSI MEC and oneM2M architectures complement each other to enable full data access and interoperability. This included initial ideas on deployment options (namely A, B, C and D) that have been considered to trigger the definition of specific use cases and requirements to demonstrate MEC-oneM2M integrations that can be used as the basis for the definition of proof-of-concepts (PoCs).

The interworking among these standard architectures can in fact enable IoT edge services (e.g., mobility in smart city, real-time analytics, etc.) to exploit oneM2M common service layer instantiated on top of MEC hosts and enable the MEC applications to exploit data coming from the IoT generated through the oneM2M standard platforms and supporting MEC system architecture as well as APIs.

Based on this, this white paper aims at identifying potential use cases that might benefit from combining MEC and oneM2M standard architectures and covering several application domains (e.g., automotive, smart cities, smart ports and industrial). As non-functional characteristics and data sharing capabilities are crucial (e.g., reliability, latency, security, etc.), the paper also investigates the existing implementations of MEC and oneM2M that support or are suitable for enhancements required to meet the implementation requirements.

1 INTRODUCTION

The Internet of things is bringing massive changes to the functioning aspect of industries, cities, and daily life. With the smart deployment and utilization of the latest IoT technologies, much optimization has been brought to the factories and autonomous vehicles in the form of super calibrated assembly lines and advanced AI processing units respectively. This has led to massive generation and accumulation of data, and therefore the need to introduce new cloud-centric architectures capable of handling potential challenges that come with these complex systems, such as latency bottlenecks, bandwidth constraints, and fragmented interoperability. Enter Edge Computing, a new cutting-edge technology that aims to bring computation & data storage closer to the source of data generation. With the help of Edge Computing, real-time responsiveness can be unlocked by processing information at the network's edge, thus enabling significant reduction in latency and alleviating the strain on centralized clouding infrastructure. A standardized framework is however needed to fully harness the power of edge computing to ensure seamless interoperability, security, and scalability. This is where the convergence of oneM2M [2] and ETSI MEC [3] emerges as a transformative solution.

The IoT Challenge: Fragmentation Meets Scalability

The process of deploying and integrating IoT systems requires uninterrupted synergy between countless devices, protocols, and vertical industries to function properly. Without proper infrastructure to link these innovative technologies with each other effectively, these systems tend to operate in silos that drastically decrease their overall efficiency, thus leading to reduced scalability and soaring costs. For instance, a smart city might comprise innovative technologies such as traffic sensors, energy grids, and environmental monitors, each requiring its own specified operating platform, but without a unified framework, it becomes significantly complex to manage and integrate such systems.

Likewise, contemporary software and emerging applications are becoming more power hungry and require more sophisticated cloud architecture to function properly. Most industries, related to autonomous vehicles and augmented reality, depend on millisecond decision-making, real-time detection, and instantaneous data processing to thrive in a competitive market. Edge computing is perfectly capable of handling and transmitting this massive stream of data to distant servers without causing any delays or bandwidth inefficiencies. Edge computing manages to perform this task through decentralized computation, the success of which depends on properly harmonizing IoT standards with edge infrastructure.

Bridging oneM2M and MEC: A Synergy for the Future

The oneM2M, a global IoT standardization community, provides a common service layer that simplifies complex networks through reusable functions like data routing, device management, and security. Standardized by a global partnership of major telecom bodies, it ensures seamless interoperability across devices and applications enabling smart meters, for example, to communicate effortlessly with city systems regardless of the underlying protocol. This reduces development costs and fosters cross-industry data sharing, making it vital for smart cities, automation, and connected healthcare.

Complementing this, ETSI's ISG MEC brings compute and storage to the network edge, near base stations and access points reducing latency and exposing rich network insights like bandwidth, location, and device context. This empowers real-time applications, from route optimization in logistics to predictive maintenance in manufacturing, through edge-powered intelligence.

The synergetic relationship between oneM2M and MEC also carries major policy implications. In most cases, vendor lock-in is reduced and compliance with data sovereignty rules is ensured though standardization, which in turn enables market stimulation and growth. For many industries, the entry barrier is also reduced due to interoperability thus giving them access to cutting edge local IoT solutions, and enabling innovation in sectors like energy, agriculture, and health. In future, the interplay of 5G, AI and edge computing will bolster the need for such synergies. Immersive metaverse, digital twins, and federated learning applications will require cloud architectures that seamlessly fuse IoT standardization with edge agility.

This white paper serves as a guide for stakeholders navigating the intersection of IoT and edge computing. It explores the technical foundations of oneM2M and ETSI MEC, presents real-world use cases across industries, and outlines deployment strategies for integrated systems. Whether you are a policymaker shaping digital infrastructure standards, an enterprise leader optimizing IoT deployments, or a developer building next-gen applications, this paper provides actionable insights to harness the transformative potential of standardized edge-enabled IoT.

By bridging the gap between standardization and innovation, oneM2M and ETSI MEC pave the way for a connected future where intelligence is distributed, responsive, and universally accessible.

1.1 ONEM2M OVERVIEW

Historically, oneM2M originates as a European standard in ETSI, which subsequently evolved into a global partnership project. Promoting its adoption helps strengthening European thought leadership. oneM2M is a global "de jure" standard, not controlled by any single private company. oneM2M was launched in 2012 as a global partnership initiative between eight of the world's preeminent standards development organizations: ARIB (Japan), ATIS (North America), CCSA (China), ETSI (Europe), TIA (North America), TSDSI (India), TTA (Korea), and TTC (Japan) to develop specifications that ensure the most efficient deployment of Machine-to-Machine (M2M) communications systems and the Internet of Things (IoT). Over the years, the partners have collaborated to develop technical specifications for a universal M2M service layer. This service layer is designed to be easily integrated into a wide range of hardware and software systems, providing a reliable foundation for connecting countless devices in the field to M2M application servers across the globe. By bringing together more than 200 players from many diverse business domains including, oneM2M ensures the global functionality of M2M and prevents the duplication of standardization effort. Current partners are:

- CCSA (China);
- ETSI (Europe);
- TIA (North America);
- TSDSI (India);
- TTA (South Korea).

Each one of the partners transposes oneM2M specifications as their own standards, thus making them valid standards in their respective geographical area. In the case of Europe, oneM2M specifications are transposed as ETSI TSs, which offers them to the public free of charge. Furthermore, the ITU-T approved oneM2M specifications as ITU standard under the Y.4500 series, making the entire suite of oneM2M specifications available for use nationally by ITU-T member

states. From a technical standpoint, the oneM2M specifications define a platform that can be rather simple, e.g., a single instance, or can be articulated over many instances that are interconnected together and cooperate to form a unified system. This is a way to achieve scalability together with separation of concerns.

oneM2M defines a scalable service layer that can be deployed as a single instance or a distributed system, ensuring flexibility across diverse IoT and data-sharing environments. Its architecture supports:

- Universal interoperability across devices, platforms, and emerging Data Spaces;
- Semantic data annotation (e.g., SAREF compatibility) and JSON-based exchange;
- Independence from the underlying network architecture: its IP- and URI-based addressing model ensures alignment with Internet standards, facilitating interoperability.

Adopting oneM2M offers significant advantages for ecosystems requiring seamless data integration:

- *Cost Efficiency*: Eliminates redundant standards, reducing development and operational costs;
- *Market Expansion*: Shifts competition from platform integration to value-added services, accelerating innovation;
- *Cross-Domain Interoperability*: Enables secure data sharing across industries, fostering new business models;
- *Native Interworking*: With legacy systems, proprietary solutions, and modern cloud environments.

Furthermore, the oneM2M key functional features include:

Data Management:

- Unified storage, historization, and real-time/contextual data sharing;
- Semantic enrichment for improved discoverability.

Security & Access Control:

- Granular permissions (role-, token-, and attribute-based);
- Dynamic policies (time-, location-, and identity-aware).

Deployment Flexibility:

- Cloud- and edge-compatible, with native device management;
- Adaptable to sector-specific requirements (e.g., industrial, smart cities).

Finally, oneM2M is designed to integrate with:

- Legacy systems and proprietary solutions, ensuring backward compatibility;
- Heterogeneous networks (any IP network, fixed/mobile, LPWAN, cellular);
- Semantic web technologies, enabling cross-domain data harmonisation.

While oneM2M is widely applied in IoT, its agnostic architecture makes it equally suited for generalised data exchange, including Data Spaces and cross-platform interoperability. By providing a standardised yet adaptable foundation, it addresses critical challenges in scalability, security, and fragmentation — positioning it as a key enabler for the connected economy.

1.2 ETSI ISG MEC OVERVIEW

The Multi-access Edge Computing (MEC) initiative is an Industry Specification Group (ISG) within ETSI (European Telecommunications Standards Institute). The purpose of the ISG is to create a standardized, open environment which allows an efficient and seamless integration of applications from vendors, service providers, and third-parties across multi-vendor Multi-access Edge Computing platforms.

The group has published a set of specifications focusing on MEC architecture (ETSI GS MEC 003 [4]), management and orchestration (MANO) of MEC applications (ETSI GS MEC 010-2 [5], application enablement and related service Application Programming Interfaces (APIs) (ETSI GS MEC 011 [6]) and the User Equipment (UE) application API (ETSI GS MEC 016 [7]).

MEC offers application developers and content providers cloud computing capabilities and service exposure at the edge of the network, in an environment characterized by ultra-low latency and high bandwidth together with real-time access to radio network information that can be leveraged by applications. MEC fosters the rapid deployment of innovative applications and services for mobile subscribers, enterprises and vertical segments. As a key supporting technology for many 5G use cases MEC can be also deployed in Wi-Fi and fixed networks, and is a key enabler for interworking with cloud providers, edge data centres and neutral hosts.

Crucial to this is the creation of a consistent set of Application Programming Interfaces (APIs) for edge developers to build services and applications. Examples of the MEC specified service APIs include, radio network information (ETSI GS MEC 012 [8]), location service (ETSI GS MEC 013 [9]), traffic and bandwidth management (ETSI GS MEC 015 [10]) and application mobility (ETSI GS MEC 021 [11]). The full set of MEC APIs are freely available from ETSI Forge (forge.etsi.org), which exploits an automated generation process to create OpenAPI compliant descriptions directly from the MEC group specifications.

In 2019, the Working Group DECODE (Deployment and ECOsystem Development) was established. Reflecting rapid growth of the MEC ecosystem, this group manages all MEC STFs, Proofs of Concept (PoCs), Deployment Trials (MDTs), MEC APIs, testing/compliance and Hackathons as well as the MEC ecosystem wiki. A number of DECODE Work Items are currently in progress, within the scope of the EC-funded ESTIMED (Enabling Standard IoT & MEC Deployments) initiative. These variously address use cases, integration solutions, MEC & oneM2M interworking, test scenarios and conformance testing.

DECODE also curates the MEC Sandbox (try-mec.etsi.org), an interactive environment where developers can experiment with their own applications based on ETSI MEC service APIs from anywhere in the world.

During 2025, there are several Work Items in ETSI MEC that can be potentially relevant for edge IoT deployments: i) a Group Report (GR) on MEC in resource constrained terminals, ii) a GR on abstracted network information exposure for vertical industries, iii) a GR on distributed edge networks, iv) a GR on exploiting edge computing resources and AI/ML in MEC v) a Group Specification (GS) describing an API gateway for client applications, vi) a GS updating the Sensor-sharing API and vii) a new GS addressing MEC support for security monitoring and management (SMM).

2 USE CASES FOR MEC AND ONEM2M INTERWORKING

The identification and detailed analysis of use cases is a strategic activity in the effort to bridge the capabilities of ETSI MEC and oneM2M. Use cases serve not only as illustrative scenarios for the application of these technologies but also as essential instruments for guiding integration efforts and shaping future standardization work. Within the scope of the ongoing project, a dedicated focus has been placed on selecting and analysing use cases that clearly demonstrate the value of combining edge computing with standardized IoT frameworks.

This activity involves a comprehensive examination of existing oneM2M and MEC implementations, including their architectures, APIs, and functional components. By investigating how these elements can be aligned and integrated to address the operational requirements of selected use cases, the project seeks to explore the technical feasibility and practical benefits of interworking between the two frameworks. Such integration is not approached as a purely theoretical exercise but is rooted in real-world application contexts, ensuring that the proposed solutions remain grounded, scalable, and actionable.

Moreover, the process of analysing use cases is expected to reveal new functionalities together with required extension in the current specifications and solutions. This, in turn, enables the identification of new functional and architectural requirements that are directly driven by the needs observed in the field. These requirements are not only valuable for improving the current state of MEC and oneM2M technologies, but they also play a crucial role in informing the broader evolution of IoT and edge computing standards. As a result, the outcomes of this use case-driven analysis will contribute to enhancing interoperability, supporting system-wide collaboration, and promoting the adoption of standardized solutions across industries.

One of the key work package of the project together with its activity tasks is focused on these aspects aiming to a set of well-defined, selected use cases along with a detailed breakdown of how each use case maps onto the capabilities of MEC and oneM2M. This activity will include a thorough analysis of any architectural adaptations or extensions that may be necessary, covering aspects such as the addition of new functional components, the need for new or updated APIs, and potential refinements to the existing frameworks. Through this analysis, the project aims to deliver a coherent view of how MEC and oneM2M can be aligned in practice to support emerging applications.

Particular emphasis will be placed on use cases that highlight the synergy between IoT and edge computing technologies, especially those that benefit from the localized processing capabilities of MEC in conjunction with the device and data management strengths of oneM2M. Dedicated focus will be given to standard-based use cases that offer clear advantages in terms of interoperability, scalability, and alignment with current regulatory and industry trends. Examples of such use cases include industrial automation systems that require real-time data processing and control, smart city deployments integrating heterogeneous IoT devices, remote monitoring solutions with strict latency and privacy requirements, and logistics applications relying on context-aware routing and tracking.

In the following sections, a number of selected use cases are presented as representative examples of how MEC and oneM2M interworking can be applied to real-world scenarios. These use cases serve as key reference points, offering tangible insights into the integration potential of the two frameworks. However, they do not represent an exhaustive list. The dynamic nature of this ongoing activity means that additional use cases may be identified over the course of the project, reflecting new discoveries, evolving requirements, and emerging technological contexts. The examples

provided thus serve as a starting point for broader exploration and are expected to be complemented and expanded as the project progresses.

2.1 SMART CITY & INTELLIGENT MOBILITY

The integration of Edge computing and IoT technologies within Smart City and Intelligent Mobility infrastructures is redefining the landscape of urban mobility and intelligent transportation systems. By combining the MEC and oneM2M frameworks, cities can deploy scalable and interoperable solutions that address the low-latency demands of communication, while also ensuring secure and efficient data exchange across heterogeneous environments. This section explores how these complementary technologies come together to enhance traffic flow, ensure the safety of road users, and support the deployment of autonomous and connected vehicles.

In the domain of connected and autonomous vehicles, MEC plays a pivotal role by delivering ultra-low-latency data processing at the network edge. This capability is essential for enabling dynamic service discovery and supporting seamless, localized communication across heterogeneous access networks. MEC facilitates real-time interactions among various urban mobility actors — including vehicles, Road Side Equipment (RSE), sensors, and actuators — by processing data closer to where it is generated and consumed. When combined with the oneM2M standard, MEC nodes are further empowered by a unified service layer that ensures interoperability and standardized communication between vehicles, infrastructure components, and other road users. This integration not only allows vehicles and connected nodes to make rapid, context-aware decisions at the edge but also enables them to access and contribute to broader IoT data ecosystems. As a result, connected vehicles benefit from enhanced situational awareness, improved coordination, and uninterrupted service continuity across diverse and dynamic mobility scenarios.

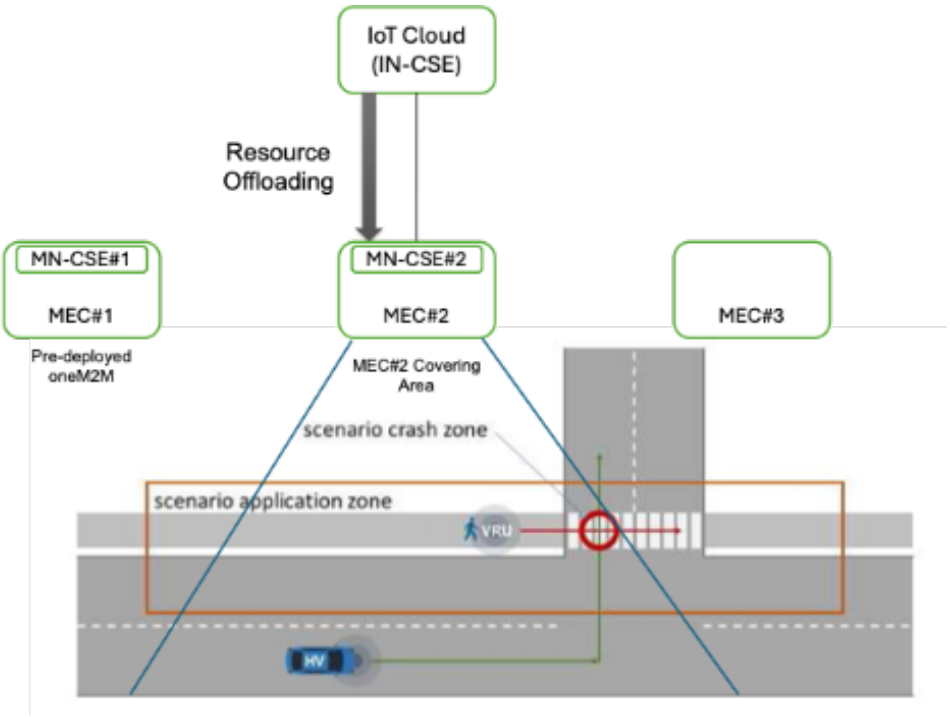

Figure 2: High-level overview of the VRU use case with MEC areas, oneM2M deployment together with tasks and resource offloading.

A particularly impactful application of MEC and oneM2M interworking can be found in the protection of Vulnerable Road Users (VRUs), including pedestrians, cyclists, and individuals with limited mobility. This use case illustrates how the integration of ETSI MEC and the oneM2M can support a responsive and intelligent VRUs Detection Service designed to enhance road safety. In scenarios involving connected vehicles operating in urban environments, the rapid detection and communication of the presence of pedestrians, cyclists, or other at-risk road users is critical. The proposed service leverages the distributed computing capabilities of MEC, positioned close to the edge of the mobile network, together with the interoperable data management model of oneM2M, to enable real-time risk prediction and alerting.

In the described scenario (illustrated in Figure 2), the oneM2M system consists of a central Infrastructure Node Common Services Entity (IN-CSE) deployed in the cloud, which acts as a global coordinator for VRU data collection and management. Data is gathered from various sources including roadside sensors, mobile applications running on pedestrians' smartphones, and other city infrastructure. When a Host Vehicle (HV) enters a specific geographical area served by a MEC node, the IN-CSE initiates a handover of relevant processing tasks and data to an edge-resident oneM2M MN-CSE (Middle Node CSE) instance. This MEC-hosted instance is dynamically deployed to reduce latency and handle localized decision-making, such as real-time collision risk assessments.

The MN-CSE, deployed as a MEC application, executes lightweight VRU detection logic using data streams from local sensors and connected devices. This edge node can interact with MEC platform services such as the Radio Network Information Service (RNIS) and Location Service to refine its contextual understanding of the area. When a VRU, such as a pedestrian, unexpectedly enters a crosswalk, the MEC node can detect this movement immediately through video analysis, LiDAR input, or GPS data from nearby mobile devices. The MN-CSE processes this event locally and sends an immediate alert to the approaching vehicle using low-latency V2X communication protocols.

This approach enables a layered architecture where the cloud-based IN-CSE handles broader coordination and historical data storage, while edge-based MN-CSE instances execute time-critical tasks close to where data is generated and where reactions are needed most. Vehicles equipped with V2X communication systems dynamically connect to the closest MN-CSE instance as they move through the network. These vehicles may initially communicate with the IN-CSE but are redirected to the edge node when latency constraints or specific context-sensitive requirements demand faster response times.

The service begins when the vehicle's onboard application submits an offloading request to the IN-CSE. This request includes parameters such as the desired latency (e.g., below 100 milliseconds), the geolocation of interest, and the logic that should be executed. The IN-CSE evaluates the request and works with the MEC orchestrator to identify a suitable edge node. Once selected, the system provisions resources, deploys the required application components, and transfers the necessary context data.

After deployment, the MN-CSE starts processing real-time inputs and builds a local oneM2M resource tree to manage subscriptions, VRU updates, and alert logic. It also communicates directly with the vehicle using MEC's local break-out and data plane capabilities. As the vehicle continues along its path, the edge-hosted MN-CSE remains active and synchronized with the cloud-based IN-CSE to maintain continuity. When the vehicle moves out of the coverage area, the MN-CSE instance

is either gracefully handed over to the next edge node or decommissioned after syncing the final state back to the cloud.

Various triggers can activate the VRU alert service. For example, if a VRU suddenly crosses into a vehicle's path, the system recognizes the potential collision and alerts the driver. Similarly, VRUs that are partially occluded or hidden behind parked vehicles can be detected by combining data from multiple sensors. The system also uses vehicle behaviour indicators, such as the activation of turn signals or proximity to intersections, to anticipate situations where vulnerable users might be at increased risk.

This use case offers several benefits. It ensures that alerts and decisions are made within strict latency bounds, improving the effectiveness of collision avoidance systems. It enables a more efficient use of network resources by offloading only the necessary logic to the edge, while leveraging the cloud for long-term data storage and service continuity. It also provides a standards-based interface between IoT data producers and consumers, facilitating interoperability between different vendors' systems.

To fully realize this service, several technical requirements must be satisfied. The oneM2M and MEC platforms must support real-time data synchronization between IN-CSE and MN-CSE instances, enforce secure access control and authentication for connected devices, and enable edge-side caching and context-aware deployment. Additionally, the orchestration of MN-CSE instances must be dynamic and scalable, responding to changes in vehicle movement patterns and network load.

In conclusion, this use case exemplifies how the integration of ETSI MEC and oneM2M can enable an intelligent, distributed, and latency-aware IoT service for road safety. By coordinating cloud-based and edge-based components, it is possible to deliver immediate situational awareness and actionable information to connected vehicles, thereby protecting vulnerable road users and enhancing the safety of intelligent transportation systems.

2.2 INDUSTRIAL & SMART FACTORY

In the rapidly advancing landscape of Industry 4.0, the convergence of edge computing with standardized IoT frameworks—such as ETSI MEC and oneM2M—represents a pivotal enabler for the next generation of smart factories. This interworking architecture supports the seamless integration of intelligent functionalities directly into the industrial environment, enabling real-time machine monitoring, advanced process automation, and data-driven decision-making.

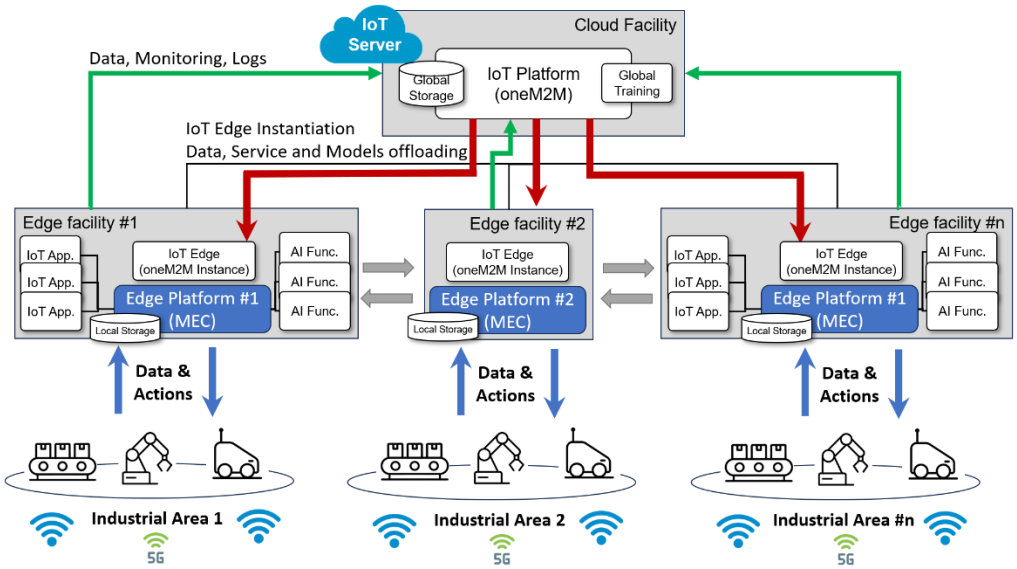

Figure 3: High-level overview of the Industrial use case with Integration of MEC and oneM2M for real-time industrial automation with local control and centralized coordination.

As shown in Figure 3, deploying computation near machines and production lines through MEC enables ultra-low latency, localized data processing, and rapid system responsiveness—crucial in environments where timing and reliability are paramount. In parallel, oneM2M offers a standardized and scalable framework for device abstraction, service discovery, and interoperability, allowing diverse industrial equipment to interact seamlessly within complex manufacturing systems. The envisioned approach spans multiple industrial areas, each equipped with an edge facility hosting a MEC platform and a local oneM2M instance. These Edge Platforms process data and execute time-sensitive tasks locally, such as interacting with industrial and monitoring production lines, ensuring low latency and high responsiveness. Each edge facility manages local IoT applications and AI functions with direct access to industrial equipment multiple connectivity according to the different requirements and characteristics of the deployments (e.g., 5G, 4G, Wi-Fi, Ethernet or LoRaWAN). Data, services, and AI models can be offloaded or synchronized with a centralized cloud facility that hosts a global oneM2M IoT platform, enabling system-wide coordination, historical data storage, and global AI model training. The cloud also collects monitoring logs and contextual data from all edge facilities, ensuring visibility and control across the distributed system. Additionally, MEC platforms support the transfer of data and control logic across edge facilities, allowing seamless operations as assets or processes shift between industrial areas.

This synergy unlocks new levels of operational efficiency through capabilities such as predictive maintenance, where sensor data from machinery is analysed at the edge to detect early signs of wear or failure. This minimizes downtime, optimizes maintenance schedules, and extends equipment lifespan. Moreover, the combination of MEC and oneM2M ensures that data is managed securely and efficiently, even in environments with stringent privacy, safety, and availability requirements. Together, MEC and oneM2M form a robust digital foundation for Industry 4.0, allowing manufacturers to build agile, intelligent, and interconnected production systems that are not only scalable and resilient but also future-proof against the evolving demands of industrial automation.

Modern industrial environments are characterized by a high degree of heterogeneity and fragmentation at the physical layer. They host a wide spectrum of devices—from fixed machines with wired connectivity to mobile collaborative robots and wireless sensor networks—each with

varying communication protocols, capabilities, and levels of mobility. Managing these diverse assets in a cohesive and scalable manner is a complex challenge. This is where the oneM2M framework offers decisive advantages. By providing a standardized and interoperable abstraction layer, oneM2M simplifies device discovery, service description, and seamless interaction across diverse equipment, enabling a more unified and manageable industrial IoT ecosystem.

The integration of oneM2M with MEC further enhances this setup by introducing the power of local computation and low-latency responsiveness at the network edge. In industrial scenarios, where low-latency and real-time operations such as fault detection, anomaly response, or adaptive process control are essential, MEC provides the required proximity and compute capabilities to support immediate data-driven decisions. The result is not only improved efficiency and safety but also the decoupling of the complexity of the underlying hardware infrastructure from the intelligent digital services that operate above it.

By embedding intelligence at every layer of the industrial process—while ensuring interoperability through oneM2M and ultra-fast compute availability through MEC—smart factories can achieve a fine balance between decentralized autonomy and centralized oversight. Additionally, MEC's role in managing multiple access networks—wired, wireless, and hybrid—becomes crucial in maintaining seamless communication flows, especially in dynamic environments with mobile assets and time-sensitive operations.

An interesting use case with respect to the industrial domain and application scenario is related to the coordination of *Autonomous Guided Vehicles (AGVs)*, continuous environmental monitoring, and precise asset tracking—all of which require both robust data handling and ultra-low latency response capabilities. By combining the global orchestration capabilities of oneM2M's Infrastructure Node (IN-CSE) with the local responsiveness enabled by MEC-hosted Middle Nodes (MN-CSE), the system achieves scalable, resilient, and efficient warehouse operations.

In a typical deployment, the oneM2M IN-CSE acts as a centralized IoT management platform that aggregates sensor data from across the warehouse. This includes temperature and humidity readings, gas leak detectors, RFID-based asset identification, and telemetry data from AGVs. These resources are structured into oneM2M's hierarchical data model and enriched with historical records and operational policies, such as acceptable climate thresholds and AGV behavioural profiles. The IN-CSE maintains oversight over the warehouse environment, continuously evaluating whether certain conditions necessitate time-critical intervention. When a situation demands a rapid or localized response—for example, rerouting AGVs away from a hazard or rebalancing asset locations in real time—the IN-CSE delegates processing responsibilities to a MEC node hosting a MN-CSE instance.

The MN-CSE is instantiated as a MEC application, operating within the edge computing environment in proximity to the physical warehouse floor. It subscribes to relevant oneM2M resources to receive real-time sensor updates and AGV telemetry. Through local processing, it executes context-sensitive logic with minimal delay. For example, if a gas sensor detects a hazardous leak in a specific zone, the MN-CSE immediately halts nearby AGVs, triggers alerts to on-site personnel via mobile or wearable devices, and interfaces with air control systems to initiate emergency ventilation. Similarly, when AGVs must navigate dynamically changing routes due to congestion or temporary obstructions, the MEC-hosted application updates their paths in real time, avoiding unnecessary delays.

Warehouse environments are typically large and segmented, often requiring AGVs and assets to move across different physical and logical zones. To maintain uninterrupted automation across these

zones, the MEC platform must support seamless mobility and service continuity. As an AGV exits one MEC coverage area and enters another, the MEC orchestrator coordinates the transfer of the relevant MN-CSE instance or the migration of task context to a neighbouring edge node. This ensures that decisions and control logic are continuously informed by the most recent local data and conditions, with no loss of state or responsiveness.

The oneM2M system supports this distributed architecture through well-defined protocols and data synchronization mechanisms. Communication between IN-CSE and MN-CSE instances may rely on HTTP or MQTT bindings, and all instances are capable of subscribing to, publishing, and caching relevant resource states. During MEC transitions, the IN-CSE acts as a stable anchor point for context synchronization and policy enforcement, ensuring that edge nodes always operate within globally defined constraints while reacting locally to real-time changes.

From the MEC perspective, this use case demonstrates the value of edge computing in supporting industrial automation. The MEC infrastructure not only provides the necessary compute and storage resources to host MN-CSE applications but also enables sub-second response times for critical operations. By localizing data processing and control execution, MEC avoids the latencies and bandwidth limitations associated with cloud-only solutions, making it ideal for high-frequency event streams such as AGV telemetry or environmental sensing. MEC's service APIs further allow applications to access local network conditions, compute availability, and location services, enabling the MN-CSE to adapt its behaviour based on real-time operational constraints.

Altogether, this integrated architecture allows warehouse operators to implement a robust automation strategy that combines the centralized governance of a oneM2M IoT platform with the responsiveness and reliability of ETSI MEC. It ensures that safety mechanisms, routing intelligence, and asset management operations are not only synchronized across the system but also executed as close as possible to the source of data and action. This synergy reduces downtime, improves resource utilization, and enhances the overall agility of warehouse operations in dynamic and potentially hazardous environments.

Ultimately, the collaboration between oneM2M and ETSI MEC represents a transformative enabler for industrial digitalization. It not only simplifies the integration of intelligent capabilities across complex factory systems but also ensures that the growing computational demands of AI-driven automation can be met locally, securely, and efficiently. This layered architecture paves the way for factories that are not only smarter and faster but also inherently more resilient and adaptable to future technological shifts.

2.3 MARITIME & SMART PORTS

This clause examines the application of ETSI MEC and oneM2M technologies within the maritime domain, with a particular focus on smart ports scenarios. The effective integration and interworking of these two standards enables significant advancements in operational efficiency, safety, and interoperability. By combining MEC's edge computing capabilities with oneM2M's standardized framework for IoT device management, ports can support low-latency and effective processing across heterogeneous access networks while ensuring seamless coordination among connected devices from different manufacturers. This approach facilitates reliable, scalable, and secure communication among systems with diverse roles, constraints, and connectivity requirements, laying the groundwork for smarter, more automated and resilient port infrastructures.

For example, surveillance cameras installed in ports for monitoring purposes can process video streams in real time using AI algorithms deployed on the MEC infrastructure. This allows the automatic detection of suspicious activities, intrusions, or hazards without the need to transfer large volumes of data to the cloud. Similarly, if a container is accessed in a restricted area, security sensors can generate an alert that is processed directly at the edge, triggering an immediate response such as notifying personnel, locking access gates, or alerting authorities, therefore drastically reducing response time compared to traditional cloud-based solutions.

Furthermore, in an increasingly interconnected and globalized environment, the logistics and supply chain industry faces complex challenges such as the need to monitor the movement of goods in real time, reduce downtime, ensure shipment security, and maintain continuous operations despite the presence of heterogeneous technologies and infrastructures. ETSI MEC and oneM2M provide a powerful framework to address these challenges, enabling low-latency communication, reliable data handling, and coordination across diverse devices and networks.

The MEC architecture plays a key role by allowing data processing to happen directly at the edge, within the port terminals or logistic centers, rather than relying on remote cloud servers. MEC also enables the deployment of local services that, by collecting and analysing data from distributed IoT sensors, can enable the detection of anomalies like unusual vibrations, critical temperature variations, or signs of structural damage on containers and transport vehicles. By acting on this information locally, interventions can be initiated before faults escalate, improving the reliability of operations. A crucial strength of MEC lies in its independence from the underlying network technology. Whether the system is connected via 5G, 4G, Wi-Fi, LoRaWAN, or other wireless-based standards, local processing continues uninterrupted, ensuring consistent operational capabilities even during network outages or congestion. While MEC enhances performance and responsiveness, modern logistics also demands interoperability among a wide variety of IoT devices and communication technologies. This is where oneM2M adds value by offering a standardized approach for data exchange and device integration, regardless of manufacturer or connectivity method. The integration of MEC and oneM2M provides a robust and scalable framework for managing a high density of connected IoT devices and services within localized environments. This approach streamlines logistics operations by enabling consistent data handling and coordination across diverse infrastructures and geographic regions, significantly lowering integration complexity and associated costs.

Distributed monitoring IoT networks for enhanced situational awareness, digitalization and the automation of logistics operations are good examples of maritime domain scenarios where a joint application of both oneM2M and MEC technologies can be exploited. In this regard, we propose a scenario focused on the Maritime Autonomous Surface Ships (MASS). This acronym is used by the International Maritime Organization (IMO) to define commercial vessels that operate with little or no human intervention. Unmanned ships should be able to navigate without colliding with other vessels, taking into account the naval traffic and unexpected situations. These ships are usually equipped with advanced technologies like integrated automation systems based on AI-models and sensing capabilities which allow them to navigate, make decisions and perform tasks autonomously. They usually rely on the communication with the Remote Operation Centers (ROCs) serving as command hubs for monitoring, controlling and managing ships from the shore. ROCs serve as pivotal hubs for overseeing and controlling unmanned vessels remotely from onshore locations so that operators can utilize advanced sensor technologies and data telemetry to monitor vessel position, speed, course, environmental conditions, and other operational parameters in real-time. The ROC is implemented following a cloud-based architecture, which is able to receive data coming from the on-board equipment (e.g., LIDARs, Radars, Cameras, IoT sensors, AIS, etc.), store and process it using

AI-based models so that to provide early warnings and prevent risks in advance. The communication between the vessel and the ROC is performed exploiting existing low-latency communication technologies such as 5G NR and beyond.

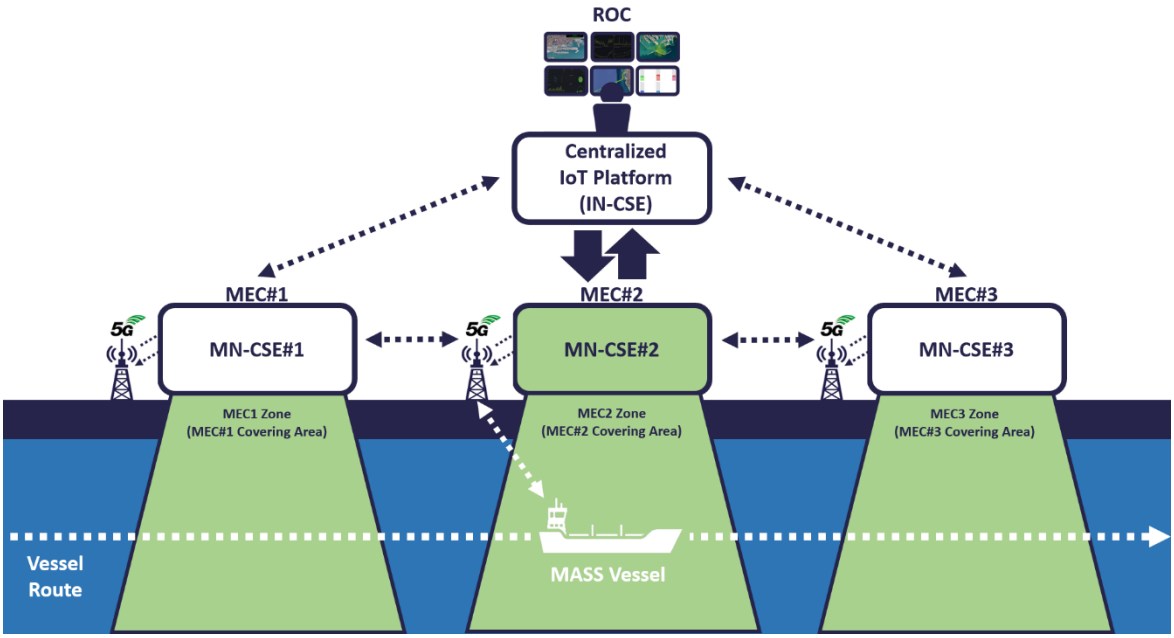

Figure 4: assisted manoeuvring scenario exploiting oneM2M and MEC.

This scenario proposes a collaborative architecture between oneM2M-compliant platform and ETSI MEC edge computing framework in order to provide assisted manoeuvring services to the unmanned ships in real-time within the seaport waters (see Figure 4). In this architecture, edge-located MN-CSE instances are deployed as MEC applications on distributed edge nodes in order to process data coming from the vessel when it approaches the assigned berth or departs from it. On top of this, a cloud-based oneM2M IN-CSE instance is deployed as a central hub for storing and managing data related to the vessel such as position, speed, course and environmental conditions. When the vessel enters the area covered by a MEC node, the cloud IN-CSE automatically offloads computational tasks (e.g., localization of other vessels) to the MN-CSE on the edge in order to guarantee low-latency processing capabilities for the generation of possible warnings. Using this approach, the MN-CSE can immediately analyse data coming from the vessel instead of relying on distant cloud servers. As the vessel transits through different MASS zones, the proposed oneM2M/MEC architecture allows to seamlessly move tasks between edge nodes, ensuring service continuity and supporting mission-critical services for the unmanned vessels (e.g., assisted manoeuvring, collision avoidance and situational awareness).

The Centralized IoT Platform (IN-CSE) is in charge of managing data coming from the unmanned vessel. It stores real-time information such as vessel localization, its speed and course as well as environmental parameters (e.g., meteorological conditions, wind speed and direction, sea current, etc.). Finally, it undertakes offloading decisions to the MN-CSEs deployed at the edge. In this scenario, the MN-CSEs are deployed as MEC applications on the MEC platform acting as a local IoT platform in charge of processing data offloaded from the Centralized IoT Platform (IN-CSE). Based on the information coming from the vessel, the MN-CSEs can generate warnings in real-time and optimize its transmission to the vessel by assessing network conditions through the MEC platform.

The MEC platform is hosted on physical edge nodes (gNodeB) and runs the oneM2M MN-CSE as a MEC application, allocating needed computing resources. The MEC platform is in charge of dynamic management (deployment and/or migration) of the MN-CSE instances based on the vessel mobility within the MASS zone. It also interacts with the Centralized IoT Platform (IN-CSE) by providing execution environments for offloading tasks and exploits edge storage capabilities for the local data processing.

By combining oneM2M platform and ETSI MEC edge computing framework, it is possible to offload assisted manoeuvring services from the oneM2M IN-CSE to the MN-CSE, exploiting the MEC platform for the allocation of the needed resources. This enables real-time decision-making and provides the service continuity for the mission-critical services required by the unmanned vessels while these are moving through multiple MEC zones.

3 EXISTING ONEM2M IMPLEMENTATIONS

Since its establishment, the oneM2M standard has been widely adopted across industries, research institutes, and academic institutions as a global framework for interoperable IoT systems. Over the past decade, a rich and resilient ecosystem has evolved around oneM2M, fueled by numerous open source and commercial implementations developed by diverse organizations. This sustained growth has been reinforced by recurring interoperability testing events, global hackathons, and academic engagement, where oneM2M platforms have been integrated into university courses and hands-on IoT education. As a result, oneM2M implementations now range from lightweight embedded CSEs for constrained environments to full-featured, cloud-ready platforms supporting edge intelligence and cross-domain integration. This section introduces representative open source platforms — ACME, tinyIoT, and Mobius—each exemplifying different strengths of the oneM2M ecosystem in terms of flexibility, portability, and deployment depth.

3.1 MOBIUS

The MOBIUS oneM2M CSE, developed and maintained by Korea Electronics Technology Institute (KETI), offers a production-grade implementation of the oneM2M standard tailored for real-world interoperability and service integration across diverse IoT domains. Developed using Node.js, MOBIUS is designed with modularity and portability in mind, making it suitable for deployment across cloud servers, enterprise systems, and edge gateways. Its compliance with oneM2M Release 2 specifications ensures standardized data handling, while its event-driven architecture allows for responsive service operations and asynchronous data processing.

Core Functionalities

MOBIUS fully implements the oneM2M Common Service Functions (CSFs), supporting key operations such as Application Entity (AE) registration, container-based data storage, ContentInstance creation, subscription and notification handling, and resource discovery. It also supports time series data modelling via specialized container types, and can interwork with external platforms and data sources through RESTful APIs. MOBIUS is suitable for both public deployments and enterprise-grade private IoT platforms, with verified use cases in smart city infrastructure, industrial automation, agriculture, and public safety systems.

Enterprise-Grade Deployment

One of MOBIUS's distinguishing features is its robust backend, designed for horizontal scalability and high availability. The platform supports multiple database engines including MySQL and PostgreSQL, and integrates seamlessly with Apache Kafka for event streaming and data pipeline extensions. It offers a Docker-based deployment model, enabling rapid scaling and orchestration in cloud-native environments. For mission-critical systems, MOBIUS supports distributed deployments with inter-node synchronization and redundancy mechanisms.

Developer Tooling and Interfaces

To facilitate development and testing, MOBIUS includes a dedicated web-based interface (Mobius Viewer) for resource visualization and inspection, as well as Postman collections and Swagger-based API documentation. Developers can interact with MOBIUS through standard oneM2M protocol bindings such as HTTP and MQTT, and the platform includes utilities for load testing, monitoring, and device simulation. Extensive logging and analytics modules allow for detailed diagnostics and performance optimization.

Interworking and Edge Support

MOBIUS supports interworking with legacy and vertical-specific IoT systems through its Interworking Proxy Entity (IPE) modules. These allow integration with Modbus, OCF, and LwM2M devices, making it a versatile bridge between heterogeneous IoT protocols. For edge deployments, MOBIUS can be run in lightweight containers and paired with edge computing frameworks to offload processing and reduce latency, enabling localized control loops and real-time response mechanisms.

Security and Governance

The platform implements robust security features including TLS encryption, resource access control policies (ACP), and secure AE authentication. MOBIUS supports multi-tenancy, enabling isolated logical spaces for different service domains or administrative groups. Its audit trail and resource versioning capabilities also facilitate regulatory compliance and traceability in sensitive IoT applications.

Technical Specifications and Requirements:

- Database Options: MySQL.
- Protocol Bindings: HTTP, CoAP, MQTT.
- Source Code: GitHub Repository https://github.com/IoTKETI/Mobius.
- Documentation: https://iotplatform.kt.co.kr/ (Korean/English).

Key Features:

- Compliance with oneM2M Release 2 and 3 specifications.
- Role-based access control and secure communication channels.
- Web-based resource viewer and API documentation tools.

Practical Applications:

- Smart City platforms for traffic, environment, and energy monitoring.
- Industrial IoT (IIoT) systems for machine condition analysis and predictive maintenance.
- Smart Farming platforms with sensor-actuator control loops.

- Disaster response systems with real-time telemetry and alerting.
- Education and research on IoT standards and data interoperability.
- Environmental monitoring in smart seaports (e.g., meteorological conditions, current meters, anemometers, bathymetry, etc.).

By combining industrial reliability with open innovation, MOBIUS serves as a highly practical and standards-compliant oneM2M platform. Its extensibility, edge readiness, and protocol interoperability make it a cornerstone for scalable IoT deployments in Korea and globally. With ongoing contributions from KETI and the oneM2M community, MOBIUS continues to evolve as a feature-rich ecosystem for the next generation of interoperable IoT services.

3.2 ACME

The ACME oneM2M CSE provides a practical implementation of the oneM2M standard, designed to simplify IoT development while maintaining strict compliance with global specifications. Its Python-based architecture ensures portability across diverse environments, from resource-constrained edge devices to cloud platforms. By abstracting the complexities of oneM2M's service layer, ACME CSE enables developers to focus on building interoperable IoT solutions rather than managing underlying infrastructure.

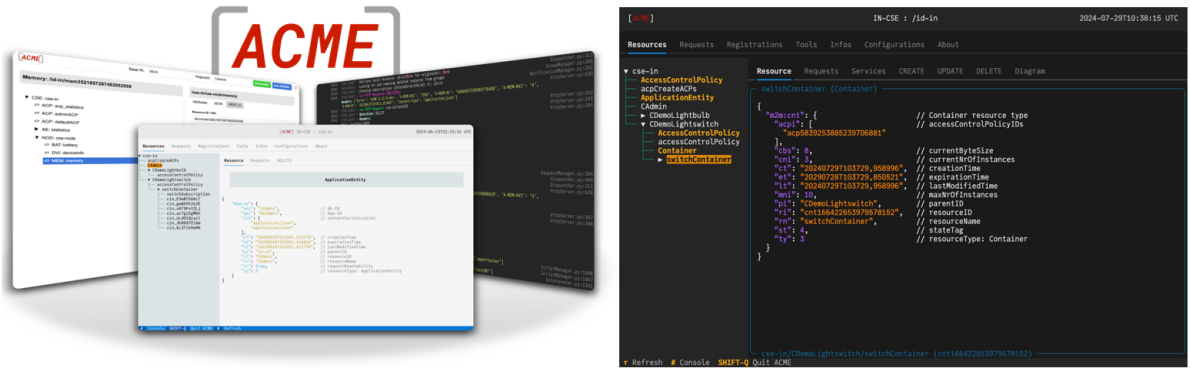

Figure 5: text UI of the ACME CSE.

One of the platform's key strengths lies in its comprehensive support for oneM2M resource types and functionalities. The implementation covers essential IoT operations including device registration, data management, and event notification, while also supporting advanced features like semantic discovery and time-series analytics. This makes it suitable for both educational purposes and real-world prototyping across various IoT domains such as smart cities, industrial monitoring, and healthcare systems. The project is actively maintained on GitHub under a permissive BSD 3-Clause license, encouraging both academic and commercial use.

ACME CSE distinguishes itself through flexible deployment options and developer-friendly tooling. The platform can run with minimal dependencies using its built-in file-based database, or scale to production environments with PostgreSQL integration. Developers benefit from multiple interfaces including a web-based UI for resource visualization, a terminal-based console for real-time monitoring, and comprehensive API access for programmatic control (see Figure 5). These features combine to create a versatile environment for testing oneM2M concepts and developing compliant

applications.

The platform's modular design supports various communication protocols including HTTP, CoAP, MQTT, and WebSocket, allowing seamless integration with diverse IoT ecosystems. Security features such as attribute-based access control and TLS encryption ensure secure operations, while the support for cross-resource synchronization enables distributed deployments. These capabilities position ACME CSE as a robust foundation for building federated IoT systems that span multiple administrative domains.

Technical Specifications and Requirements:

- Minimum System Requirements: Python 3.10+ (tested up to 3.13).
- Supported Platforms: Linux (including Raspberry Pi OS), Windows, macOS.
- Database Options: TinyDB (file-based/in-memory) or PostgreSQL.
- License: BSD 3-Clause (open source, permissive).
- Source Code: GitHub Repository.
- Documentation: Official Website.

For researchers and educators, ACME CSE offers unique advantages as a teaching tool for oneM2M standards. The included scripting engine allows for extending platform functionality, while the request recording feature provides visibility into protocol interactions. These features, combined with detailed documentation and an active development community, make ACME CSE particularly valuable for academic institutions and standardization studies.

Key Features:

- Full support for oneM2M Releases specifications.
- Multiple protocol bindings (HTTP, CoAP, MQTT, WebSocket) with TLS security.
- Advanced data management through TimeSeries and FlexContainer resources.
- Attribute-based access control with time-restricted policies.
- Interactive Web and terminal interfaces for administration.
- Scripting engine for functionality extension.

Practical Applications:

- Smart city sensor networks with distributed data aggregation.
- Industrial equipment monitoring with time-synchronized analytics.
- Healthcare telemetry systems with secure data transmission.
- Academic research in IoT interoperability and standards.
- Prototyping of oneM2M-compliant services and applications.

By combining standards compliance with practical usability, ACME CSE serves as an accessible entry point to oneM2M while providing enough depth for serious IoT development. The platform continues to evolve through community contributions, ensuring alignment with emerging IoT requirements and use cases.

3.3 TINYIOT

The tinyIoT oneM2M CSE, developed at Sejong University, is a high-efficiency, standards-compliant open-source implementation of the oneM2M platform, written entirely in C. Initially designed to meet the strict performance requirements of embedded and resource-constrained environments, tinyIoT also runs reliably in general-purpose computing systems, making it suitable for deployment across a broad range of platforms—from microcontroller-based edge devices to full-scale Linux servers and cloud infrastructure. Its architecture emphasizes performance, portability, and compliance with international IoT standards.

Performance and Deployment Flexibility

The C-based implementation of tinyIoT enables deterministic execution and fine-grained control over system resources. It is optimized for edge deployments such as Raspberry Pi, OpenWRT routers, and industrial gateways. However, the platform is not limited to constrained devices. tinyIoT has been successfully deployed in standard Linux environments (e.g., Ubuntu, CentOS) and Docker containers, allowing it to serve as a development, simulation, and even production-grade backend for oneM2M services.

Developers and researchers can build and test IoT services using tinyIoT on local machines or servers, and later deploy the exact same implementation to embedded targets—ensuring seamless portability and consistency.

Core Capabilities

Despite its minimal footprint, tinyIoT supports key oneM2M features, including:

- AE registration, hierarchical Container/ContentInstance management.
- Group management and various discovery options.
- Subscription/Notification handling with precise control over resource state.
- Time-based resource expiration and access control policies (ACP).
- RESTful HTTP API interface for integration and testing.
- Efficient internal data store using file-based or in-memory persistence.

TinyIoT also includes customizable modules to extend support for additional features without increasing baseline complexity.

Developer Tooling and Integration

The platform offers a modular build system, example configurations, and testing scripts to accelerate onboarding. Developers can interact using standard HTTP REST bindings, and gain deep visibility into system behaviour through detailed logs and diagnostic outputs. A C-based test client is included to simulate IoT devices, and the platform is compatible with oneM2M conformance tools.

Because of its transparent behaviour and fine-tuned implementation, tinyIoT is used in academic institutions and standardization testbeds to explore interoperability, protocol correctness, and system-level optimization.

Interworking and Extensibility

While lightweight, tinyIoT is designed with real-world extensibility in mind. It includes a modular interface for implementing Interworking Proxy Entities (IPEs) to connect with external protocols such

as Modbus, DDS. Its C-based structure allows developers to embed custom logic directly within the platform, offering full control over timing, scheduling, and I/O.

The platform has been integrated with edge analytics modules, hardware abstraction layers, and blockchain proxies to explore secure and verifiable IoT data flows using open standards.

Technical Specifications and Requirements:

- Supported Protocols: HTTP, CoAP, MQTT, and webSocket.
- Persistence: In-memory or file-based SQLite (optional).
- Source Code: https://github.com/seslabSJU/tinyIoT.
- Documentation: https://seslab.sejong.ac.kr.

Key Features:

- Full support for oneM2M resource hierarchy and core CSFs.
- Highly portable C-based code suitable for embedded targets.
- Deterministic memory and task management.
- Minimal system overhead for real-time or power-sensitive applications.
- Modular build system and external interface support (IPEs).
- Clean logging and debug output for learning and analysis.
- Compatible with oneM2M-compliant tools, test suites, and digital twin systems.

Practical Applications:

- Embedded sensor networks and actuator control in smart farms.
- Real-time monitoring systems for industrial safety and energy optimization.
- Digital twin backends for physical system modelling.
- Teaching tool in academic settings for understanding IoT standards.
- Blockchain interworking for secure and verifiable sensor data pipelines.

By offering a clean, high-performance implementation of the oneM2M standard, tinyIoT serves as a bridge between embedded development and scalable IoT infrastructure. Whether used for field deployments, lab experiments, or conformance studies, tinyIoT embodies simplicity, reliability, and interoperability—hallmarks of effective IoT system design.

4 EXISTING MEC IMPLEMENTATIONS

Ever since its inception, ETSI Multi-access Edge Computing (MEC) has slowly gained popularity across diverse industries, benefitting from multiple emerging platforms and initiatives to support its practical adoption along the way [12]. Among these, the ETSI MEC Sandbox [13] has risen as a main reference implementation, specifically designed and developed by ETSI ISG to prototype, test, and validate MEC applications in a controlled and flexible environment. This section focuses on highlighting the ETSI MEC Sandbox as a primary example, describing its role in facilitating MEC innovation and enabling first-hand experimentation through the state-of-the-art cloud-based edge infrastructure.

4.1 ETSI MEC SANDBOX

Innovation, speed, and ease of access constitute the main ingredients of the ever-evolving landscape of edge computing. To integrate these aspects in the development process of MEC applications, the ETSI MEC Sandbox has been developed to provide flexible exploration and testing of applications in a controlled yet realistic environment. By providing a pre-integrated and standards-compliant framework, the barrier costs have been significantly lowered thus leaving more room for solution architects, researchers, and developers to innovate freely by providing them direct access to MEC APIs, network behaviour emulation tools and application lifecycle management features.

The Sandbox is designed to closely mirror ETSI MEC specifications, providing a robust testing environment that bridges the gap between high-level architectural concepts and their concrete implementation. At its core, the Sandbox provides hands-on access to ETSI MEC service APIs, which are built on RESTful principles and structured for interoperability across platforms [14]. This enables developers to rapidly prototype MEC applications without needing to set up a physical edge infrastructure. It also allows teams to validate the behaviour of MEC services, test various deployment configurations, and gain first-hand experience with the mechanisms underpinning edge computing ecosystems.

Key Capabilities: Enabling Realistic and Scalable Experimentation

One of the Sandbox's most compelling features is its capability of simulating diverse network conditions, including 4G, 5G, and Wi-Fi [15], in combination with various device mobility scenarios, from stationary endpoints to high-velocity terminals. This flexibility allows users to evaluate the behaviour and responsiveness of MEC services in realistic conditions. Access is streamlined through a single-page web application, which provides a user-friendly interface for deploying network scenarios, configuring terminal profiles, managing MEC application instances, and controlling the lifecycle of MEC services.

A notable addition to the platform is the integration of Swagger UI [16], enabling direct API interaction from within the browser. This enhances the developer experience by offering immediate visibility into API structures, parameters, and expected responses, reducing reliance on external tools and easing the learning curve for new entrants to MEC. Beyond these built-in tools, the Sandbox includes a real-time API console that captures and displays ongoing interactions between applications and the MEC platform. This feature is especially valuable for debugging and optimization, as it allows developers to trace API calls, monitor response times, and observe event-driven notifications as they occur.

For advanced users and integrators, the capability of interfacing with MEC APIs from external applications is also provided, offering enhanced flexibility for integration testing and custom software development. This makes it possible to integrate the Sandbox into broader testing frameworks or development pipelines. In addition, MEC Sandbox's command-line API capabilities provide intuitive CLI access to core platform functions—including scenario orchestration, application lifecycle management, and real-time MEC service interaction—empowering developers and innovators to experiment and validate edge solutions efficiently [15].

Access to Core MEC Services Supported by the Sandbox

The ETSI MEC Sandbox provides access to a suite of essential MEC services that are critical for a wide range of applications. These services expose valuable network and contextual information to edge applications through standardized RESTful APIs.

Application Enablement API (GS MEC 011)

The Application Enablement API, defined in ETSI GS MEC 011 [6], is crucial for managing the lifecycle of MEC applications. It facilitates the registration and deregistration of applications with the MEC platform, enables the discovery of available MEC platform services, and supports event notifications related to service and application availability. This API is vital for building scalable and dynamic IoT solutions in MEC environments.

Radio Network Information Service (RNIS, GS MEC 012)

The Radio Network Information Service, based on ETSI specification GS MEC 012 [8], offers visibility into various aspects of the cellular network. This includes metrics on radio network conditions, measurements related to the user plane data transfer, and information about UEs connected to the radio network, such as their context and Radio Access Bearers (RABs). Radio Network Information Service (RNIS) also supports notifications for events like cell changes and RABs establishment or release.

Location API (GS MEC 013)

The Location API, defined by ETSI specification GS MEC 013 [9], enables applications to access location-related data for both user equipment (UE) and geographical zones. This includes precise geospatial coordinates as well as network-level location information. Furthermore, the service supports notification events, allowing applications to subscribe to updates on user tracking, zonal status changes, and traffic conditions.

Application Mobility Service (AMS, GS MEC 021)

The Application Mobility Service, defined by ETSI GS MEC 021 [11], supports the relocation of user context and even entire application instances across different MEC platforms. The Sandbox currently focuses on MEC-assisted context transfers, where the MEC system plays a role in triggering and facilitating the transfer.

MEC IoT API (GS MEC 033)

The MEC IoT API, specified in ETSI GS MEC 033 [17], is designed to streamline the provisioning and management of IoT devices within MEC environments. It supports interactions between IoT devices and applications hosted at the edge, enabling efficient data exchange and control.

Other Supported Services

In addition to the core services mentioned above, the ETSI MEC Sandbox also supports other important APIs, including the Traffic Management API (GS MEC 015) [10], Device Application Interface (DAI, GS MEC 016) [7], V2X Information Service (GS MEC 030) [18], and MEC Federation (GS MEC 040) [19]. The wide range of supported MEC services within the Sandbox highlights its comprehensive nature, allowing developers to experiment with various aspects of edge computing.

The inclusion of services like the MEC Federation API indicates a forward-looking approach, anticipating the need for interoperability in complex, multi-operator edge deployments. MEC Federation addresses the scenario where edge resources and services need to be shared across different administrative domains. By providing tools to experiment with this concept, the Sandbox prepares developers for the future of edge computing, where collaboration and resource sharing will be critical [20].

Standardized EDGE Ecosystem

By aligning tightly with ETSI MEC specifications, the Sandbox acts as more than just a development tool, as it becomes an enabler of interoperability and edge innovation. Its interactive architecture and curated API access provide a practical pathway for developers, researchers, and solution architects to understand, prototype, and implement MEC-aligned applications across sectors such as smart cities, connected vehicles, and industrial automation.

Ultimately, the ETSI MEC Sandbox plays a pivotal role in fostering a standardized, scalable, and developer-friendly edge computing ecosystem. It bridges the gap between conceptual standards and real-world applications, paving the way for an integrated future where intelligence is not only distributed but also universally accessible, secure, and responsive to the dynamic needs of modern applications.

5 CONCLUSIONS AND FUTURE WORK

In this white paper, the authors provided an overview of the existing challenges focused on deployment and integration of the IoT systems, highlighting the role of the Edge Computing and the importance of harmonizing IoT-related standards with the edge infrastructure. The authors explored the technical foundations of oneM2M and ETSI MEC standards and how these are filling the gap between standardization and innovation processes by paving the way for a connected future where intelligence is distributed, responsive and accessible. Furthermore, it was shown that the consequent identification and detailed analysis of use cases is a strategic activity in the effort to bridge the capabilities of ETSI MEC and oneM2M as those serve not only as illustrative scenarios for the application of these technologies, but also as essential instruments for guiding the integration efforts and shaping the future standardization efforts.

Finally, the authors provided a comprehensive investigation of existing oneM2M and MEC implementations, including their architectures, interfaces, key features, flexibility, portability and deployment requirements. By investigating how these components can be aligned and integrated to address the operational requirements of selected use cases, this white paper aimed at exploring the technical feasibility and practical benefits of interworking between the two frameworks. Such integration has been rooted in real-world application scenarios, ensuring that the proposed solutions remain grounded, scalable, and actionable.

The next step will be that of assessing the technical implications and deployment approaches to be tested through multiple proof-of-concepts (PoCs) and to identify potential standardization gaps that need to be addressed in the future by reference bodies.

ABBREVIATIONS

ACP	Access Control Policy
AE	Application Entity
AGV	Autonomous Guided Vehicle
AI	Artificial Intelligence
AIS	Automatic Identification System
AMS	Application Mobility Service
API	Application Programming Interface
ARIB	Association of Radio Industries and Businesses
ATIS	Alliance for Telecommunications Industry Solutions

BSD	Barkeley Software Distribution
CCSA	China Communications Standards Association
CLI	Command Line Interface
CoAP	Constrained Application Protocol
CSE	Common Service Entity
CSF	Common Service Function
DAI	Device Application Interface
DDS	Data Distributed Service
DECODE	Deployment and ECOsystem Development
DL	Deep Learning
ETSI	European Telecommunications Standards Institute
FL	Federated Learning
GS	Group Specification
HTTP	Hypertext Transfer Protocol
IIoT	Industrial Internet of Things
IMO	International Maritime Organization
IN	Infrastructure Node
IPE	Interworking Proxy Entity
IoT	Internet of Things
IP	Internet Protocol
ISG	Industry Specification Group
ITU-T	International Telecommunication Union
JSON	JavaScript Object Notation
KETI	Korean Electronics Technology Institute
LIDAR	Light Detection and Ranging
LoRaWAN	Long Range Wide Area Network
LPWAN	Low Power Wide Area Network
LwM2M	Lightweight Machine-to-Machine
M2M	Machine to Machine
MASS	Maritime Autonomous Surface Ship
MEC	Multi-access Edge Computing
ML	Machine Learning
MN	Middle Node
MQTT	Message Queuing Telemetry Transport
NR	New Radio
OCF	Open Connectivity Foundation
OpenWRT	Open Wireless Router
PoC	Proof-of-Concept
RAB	Radio Access Bearers
RESTful	Representational State Transfer
RFID	Radio Frequency Identification
RNIS	Radio Network Information Service

ROC	Remote Operation Center
RSE	Road Side Equipment
SAREF	Smart Applications REFerence
SMM	Security Monitoring and Management
SQL	Structured Query Language
STF	Specialist Task Force
TIA	Transportation Intermediaries Association
TLS	Transport Layer Security
TSDSI	Telecommunications Standards Development Society India
TTA	Telecommunications Technology Association
TTC	Telecommunication Technology Committee
UE	User Equipment
UI	User Interface
URI	Uniform Resource Identifier
V2X	Vehicle to Everything
VRU	Vulnerable Road Users

REFERENCES

[1] oneM2M, ETSI ISG MEC and, “Enabling Multi-access Edge Computing in Internet-of-Thnigs: how to deploy ETSI MEC and oneM2M,” [Online]. Available: https://www.etsi.org/images/files/ETSIWhitePapers/ETSI-WP59-Enabling-Multi-access-Edge-Computing-in-iot.pdf.

[2] oneM2M, “What is oneM2M?,” [Online]. Available: https://www.onem2m.org/using-onem2m/what-is-onem2m. [Accessed 25 April 2025].

[3] European Telecommunications Standards Institute, “Multi-Access Edge Computing (MEC),” [Online]. Available: https://www.etsi.org/technologies/multi-access-edge-computing. [Accessed 25 April 2025].

[4] ETSI GS MEC 003, “Multi-access Edge Computing (MEC); Framework and Reference Architecture,” 2022. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/03.01.01_60/gs_mec003v030101p.pdf .

[5] ETSI GS MEC 010-2, “Multi-access Edge Computing (MEC); MEC Management; Part 2: Application lifecycle, rules and requirements management,” 2022. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/01002/02.02.01_60/gs_MEC01002v020201p.pdf.

[6] ETSI GS MEC 011, “Multi-access Edge Computing (MEC); Edge Platform Application,” [Online]. Available: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/011/03.03.01_60/gs_mec011v030301p.pdf .

[7] ETSI GS MEC 016, “Multi-access Edge Computing (MEC); Device application interface,” [Online]. Available: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/016/03.01.01_60/gs_mec016v030101p.pdf .

[8] ETSI GS MEC 012, “Multi-access Edge Computing (MEC); Radio Network Information,” [Online]. Available: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/012/02.02.01_60/gs_mec012v020201p.pdf .

[9] ETSI GS MEC 013, “Multi-access Edge Computing (MEC); Location API,” [Online]. Available: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/013/03.02.01_60/gs_mec013v030201p.pdf .

[10] ETSI GS MEC 015, “Multi-access Edge Computing (MEC); Traffic Management APIs,” [Online]. Available: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/015/03.01.01_60/gs_mec015v030101p.pdf .

[11] ETSI GS MEC 021, “Multi-access Edge Computing (MEC); Application Mobility Service API,” [Online]. Available: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/021/03.01.01_60/gs_mec021v030101p.pdf .

[12] ETSI, “MEC Ecosystem - MECWiki,” [Online]. Available: https://mecwiki.etsi.org/index.php?title=MEC_Ecosystem. [Accessed 7 May 2025].

[13] ETSI, “MECWiki,” [Online]. Available: https://mecwiki.etsi.org/index.php?title=MEC_Sandbox. [Accessed 28 April 2025].

[14] ETSI GS MEC 009, “General principles, patterns and common aspects of MEC Service APIs,” [Online]. Available: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/009/03.03.01_60/gs_mec009v030301p.pdf.

[15] MECWiki, [Online]. Available: https://mecwiki.etsi.org/index.php?title=MEC_Sandbox_Help. [Accessed 28 April 2025].

[16] “Swagger UI,” [Online]. Available: https://swagger.io/tools/swagger-ui/. [Accessed 28 April 2025].

[17] ETSI GS MEC 033, “Multi-access Edge Computing (MEC); IoT API,” [Online]. Available: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/033/03.01.01_60/gs_mec033v030101p.pdf.

[18] ETSI GS MEC 030, “Multi-access Edge Computing (MEC); V2X Information Services API,” [Online]. Available: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/030/03.03.01_60/gs_mec030v030301p.pdf.

[19] ETSI GS MEC 040, “Multi-access Edge Computing (MEC); Federation enablement APIs,” [Online]. Available: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/040/03.03.01_60/gs_mec040v030301p.pdf.

[20] ETSI GR MEC 035, “Multi-access Edge Computing (MEC); Study on Inter-MEC systems and MEC-Cloud systems coordination,” [Online]. Available: https://www.etsi.org/deliver/etsi_gr/MEC/001_099/035/03.01.01_60/gr_mec035v030101p.pdf.

[21] L. U. Khan, I. Yaqoob, N. H. Tran, S. M. A. Kazmi, T. N. Dang and C. S. Hong, “Edge-Computing-Enabled Smart Cities: A Comprehensive Survey,” *IEEE Internet of Things Journal,* vol. 7, no. 7, pp. 10200-10232, 2020.