MEIX: Evolving Multi-Access Edge Computing for Industrial Internet-of-Things Services

Lionel Nkenyereye, JaeYoung Hwang, Quoc-Viet Pham, and JaeSeung Song

***Abstract*—The Internet of Things (IoT) has introduced advanced vertical use cases in many sectors such as smart factories and smart cities. Most of the use cases are managed by a cloud-based IoT service layer platform that may affect the quality of service due to network communication latency. The early concept of edge computing has proven significant results in scheduling lightweight tasks for edge nodes and highly computational resources for the cloud. To support latency-critical IoT services, we propose an interworking architecture between the IoT service layer platform and Multi-Access Edge Computing (MEC) at the network edge. The core idea of our architecture is to virtualize and instantiate required IoT service functions to the edge of the network, close to IoT devices that need ultra-low latency and high bandwidth. The interworking architecture is designed based on two global standards: oneM2M for IoT and MEC for edge computing. We present a system based on such standardized techniques, MEIX, and show how it can support mission-critical IoT services at the edge nodes. MEIX introduces a novel component that enables running virtual IoT common service functions to support the interoperability of IoT platform and network functions on the top of MEC-based systems.**

***Index Terms*—Internet of things, IoT service layer, multi-access edge computing, oneM2M standard, virtual network functions.**

I. Introduction

The proliferation of sensor devices has promoted a huge number of Internet of things (IoT) services, but network communication latency diminishes the user experience. These IoT devices regularly transmit a huge amount of sensed data generated to a cloud-based IoT platform and processed by big data analytic or machine learning algorithms. The accuracy of the output data from computation units and the response time is significant to low-latency IoT applications. Furthermore, the cloud-based IoT platform adds smartness to IoT applications and provides common service functions (CSFs), including device management, database storage, visualization, AI system, and more, to various IoT services. However, cloud-based IoT CSFs offers very high communication latency and is therefore not suitable for latency-sensitive IoT services such as autonomous cars and, smart healthcare as discussed [1].

The deployment of IoT applications in a 2-tiered way, cloud-devices, does not meet the requirements of ultra-low latency, location awareness services and mobility of end devices. An evolved three-tiered architecture including a middle layer between the cloud and things labeled as the Fog computing. Yuam *et al.* [2] proposed the Fog driven radio access network (Fog-NFV) to support computing-intensive tasks to multiples Fog-RAN nodes and to meet the requirements of low-latency IoT applications. Similarly, the European Telecommunications Standards Institute (ETSI) established the Industrial Standard group (ISG) to develop the concept of multi-access edge computing (MEC). MEC pushes the computing applications, data, and services away from the central nodes to the network edge, enabling IoT applications with awareness of device geo location and context. Based on the ETSI ISG MEC standard architecture, evolved IoT architecture based on this standard is proposed. The work [3] proposed MEC-based IoT platform to conceptualize the benefit of the ETSI ISG MEC architecture framework by proposing a gateway middleware as a service. The MEC based IoT platform leverages orchestration (way to automatically arrange and manage software services and resources), multi-tenancy and network slicing. That is why, MEC-based IoT platform is highly favored by telecom service providers to support critical IoT applications; MEC-based IoT is more suitable for MEC mobile networks than Fog-RAN. Since oneM2M (a common platform for IoT) [4] has defined a common platform for IoT, which can interoperate with a wide range of technologies used to connect end-devices, the proposed design in our current work is more suitable for industrial IoT services.

In this article, we introduce MEIX, a novel interworking architecture that allows the provisioning of virtual IoT platform at the MEC platform. MEIX enables virtualized IoT platforms to run at the network edge nodes to meet requirements for mission-critical IoT services. MEIX enables dynamic instantiation of IoT service platforms to the edge nodes with only required CSFs on the top of virtualization infrastructure and, subsequently, grant IoT devices access to required CSFs with low network latency. MEIX also allows IoT devices to discovery MEC services; therefore, proper IoT CSFs that can satisfy required latency and bandwidth can be instantiated to a selected edge node. In order to provide a globally interoperable solution, MEIX is designed based on two international de facto standards, i.e., oneM2M and MEC. MEIX manages the deployment of the oneM2M IoT platform instance to the MEC platform. The IoT platform instance (iInst) can be instantiated over multiple virtual machines (VMs) where each VM implements a slide of CSFs of the IoT platform. The IoT instance accesses the MEC services through the MEC platform, for example, registering itself to the MEC orchestration catalogue.

In light of the above considerations, this article brings the

Lionel Nkenyereye, JaeYoung Hwang, and JaeSeung Song (corresponding author) are with the Department of Computer and Information Security, Sejong University, Seoul 05006, Korea (e-mail: {nklionel, jssong}@sejong.ac.kr, forest62590@sju.ac.kr).

Quoc-Viet Pham is with the Korean Southeast Center for the 4th Industrial Revolution Leader Education, Pusan National University, Busan 46241, Korea (e-mail: vietpq@pusan.ac.kr).

following contributions:

1) A detailed review of the oneM2M and MEC standards to design a novel architecture that provides virtualized IoT common services at the edge of the network.
2) A design and implementation of a standard-based interworking architecture integrating Edge computing and IoT service functions for mission-critical IoT services.
3) An in-depth analysis of MEIX through a use case to show an IoT instance enabling access to virtual IoT CSFs over MEC can meet requirements for mission-critical IoT services.

We compose the remainder of the paper as follows: Section II presents MEC and oneM2M IoT platform standards. The proposed architecture, namely MEIX, is described in Section III. We discuss the value proposition of MEIX in Section IV. Finally, we draw the conclusion remarks with future works in Section V.

II. Background and Motivation

As MEIX is designed based on two global standards, we present an overview of oneM2M and ETSI MEC standards.

A. oneM2M standard platform

Many IoT platforms are deployed as silos with proprietary technology to provide a solution for a specific vertical because a lot of components and technologies are involved. Such proprietary solution causes the issue of interoperability and results in market fragmentation. To cope with such issues, a global standard initiative for IoT service layer platform known as oneM2M was established in 2012 with several major standards development organizations such as ARIB (Japan), ATIS (USA), CCSA (China), ETSI (Europe), TTA (Korea).

The common service layer of oneM2M provides a set of CSFs that is required to support various IoT applications. At the time of writing this paper, oneM2M has defined fourteen CSFs, such as Registration, Discovery, Security, and Device Management. A set of Nodes that are logical entities are classified into domains including field and infrastructure, as shown in Fig. 1. Infrastructure Node (IN) represents the IoT server (e.g., network and cloud server) that contains one CSE and probably multiple AEs. Middle Node (MN) with comprehensive communications functional components. Similar to IN, one CSE and multiple AEs can be accommodated into a single MN. IoT devices in the field domain represent and form Application Dedicated Node (ADN) and Application Service Node (ASN). AE is a logical entity that implements an IoT application, while CSE represents an instantiation of a set of CSFs that AEs can use.

B. MEC-based IoT platform

The ETSI ISG MEC architecture features Network Function Virtualization Infrastructure (NFVI) to install innovative applications and services on top of the edge cloud facilities or at the network edge. For instance, end-to-end network slicing based on MEC and central cloud for IoT services was proposed to dynamically place microservices encapsulated as Virtual Network Functions (VNFs) along the end-to-end path [5]. The high performance of this framework demonstrated the flexibility of deploying end-to-end slices in a very short time. Moreover, the MEC-based IoT platform in [3] conceptualizes the design of an IoT gateway as a middleware where multiple virtualized IoT gateway instances from different wireless access technologies and protocols can run.

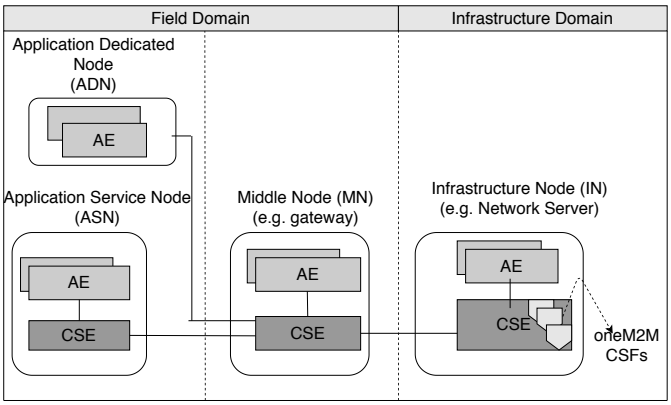


Fig. 1: oneM2M reference architecture.

C. Virtualization functions for the oneM2M service layer

The Network Function Virtualization (NFV) [6] utilizes virtualization technologies to replace hardware proprietary network entities with VNFs that run as software on VMs. NFV enables to slice an IoT platform in the form of multiple virtual IoT CSFs at the network edge. To achieve this, virtualization technologies allow the transformation of IoT CSFs from the common service layer to virtual IoT CSFs like software images. These virtual IoT CSFs include resources and service functions that have attributes specifically designed to meet the needs for IoT vertical markets such as smart building, industrial IoT, smart city. Therefore, virtual IoT CSFs can be deployed anywhere through he management mechanism and orchestration of the ETSI ISG MEC platform over the MEC host when needed, creating oneM2M CSFs as a service model for IoT utilization as depicted in Fig. 2.

The workflow of oneM2M virtual service layer for MEC system is discussed as follows. An industrial IoT solution team requests a set of virtual CSFs at a cloud-based IoT platform (which is considered as a physical server at the cloud). The requester virtual CSFs selects a set of pre-packaged images or pulls container-based images according to the description of services and requirements of the industrial IoT solution. For that, the implementer of virtual CSFs packages CSFs as VNF in a one zip file (including qcow2 disk image, image properties file (vCPU, memory), bootstrap configuration file (username, password)) which will be uploaded to the MEC system for orchestration and instantiation. If the implementer of virtual CSFs failed to prepare the VNF zip file, the industrial IoT solution team would revise the description of virtual CSFs. The virtual CSFs run as isolated resources in the MEC host of the ETSI ISG MEC architecture.

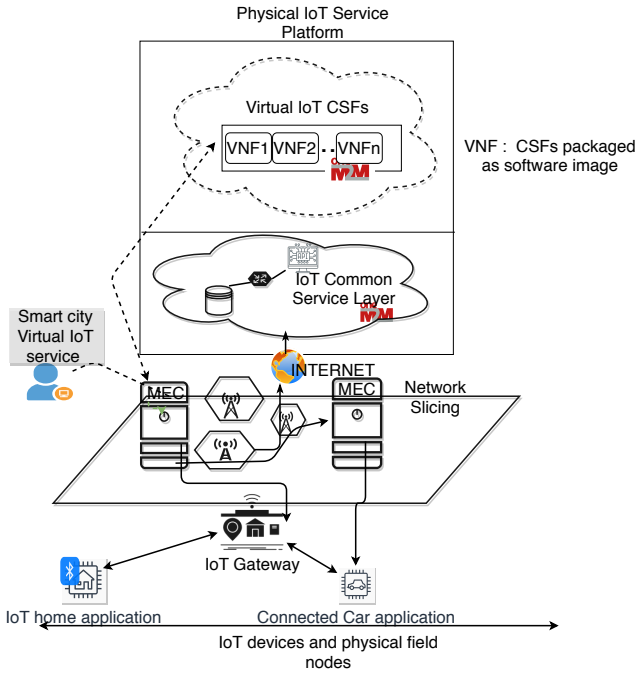


Fig. 2: oneM2M virtual service layer deployment for generalized MEC systems.

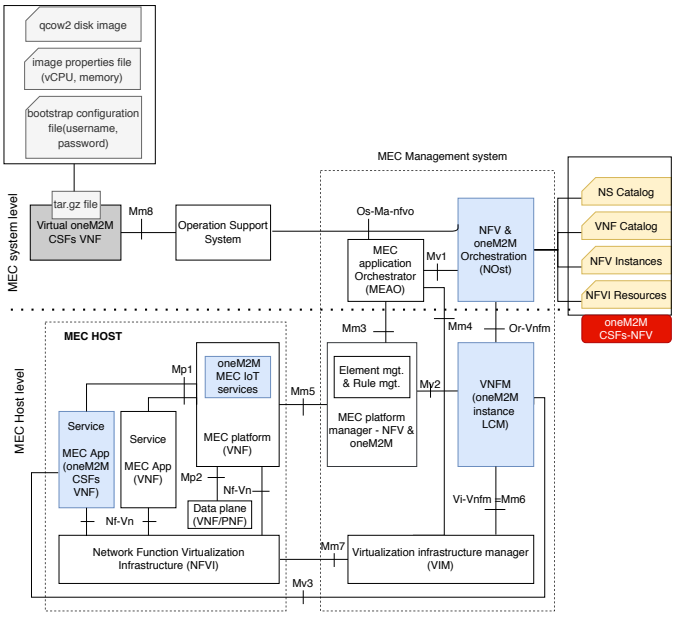


Fig. 3: The proposed reference MEIX architecture.

III. MEIX Architecture Design and Key Components

Based on the strong needs from the market, we propose the MEIX architecture design. The main idea is to virtualize required CSFs of an IoT platform and run it dynamically on the MEC platform. In particular, to guarantee the global interoperability, we propose new architecture elements and communication reference points for extending the MEC solution to support the integration of an IoT platform developed based on oneM2M standard.

A. Overview of MEIX Reference Architecture

oneM2M based IoT platform and ETSI MEC based edge computing platform run independently at different layers. In order to enable MEC to host virtualized IoT platform instances, we propose the orchestration of IoT platform instances through an orchestrator compliant with the NFV orchestrator (NFVO) compliant entity to help standardizing the functions of virtual networking and a broker for the creation of new service features. Within MEIX, a virtualized IoT platform called *omInst* is orchestrated as an instance that runs on top of the MEC host. On the other hand, a service oneM2M MEC IoT service that supports an interworking between oneM2M and MEC is included in the MEC service catalogue like other MEC services, such as the Radio Network Information Service (RNIS), Domain Name Service (DNS), etc.

Fig. 3 depicts the MEIX architecture that is built on top of the MEC reference architecture. In order to enable the virtualization of oneM2M-based IoT service functions and offloading of tasks that require data processing, we adapt the well-known NFV framework in MEIX. The entities are categorized into the MEC system level and host level entities.

B. MEC Host for Virtualized IoT Functions

In MEIX the MEC applications for oneM2M appear like VNFs, therefore we can reuse the existing ETSI NFV functionality. The MEC host level consists of the MEC host and associated management entities. The MEC host is further composed of the MEC platform, the MEC applications and the NFVI. The MEC system level entities (e.g., orchestrators for NFV and MEC applications) are located on top of the MEC host entities to handle management and operation related tasks. The MEC platform (MEP) acts as a proxy between mobile network and the MEC applications for oneM2M. It allows MEC applications to expose and consume oneM2M CSFs, which are provided as MEC services via the Mp1 interface. MEP also sets the policy and configuration rules for forwarding user plane traffic to MEC applications for oneM2M. MEP interacts with the mobile network via the Mp2 interface. This allows MEP to get a set of information such as statistics of RAN on UEs and evolved Node B (eNB) [7].

C. Orchestration of oneM2M MEC instance

The core orchestration functionalities, i.e., onboarding and instantiation of IoT service functions is handled by the enhanced NFV Orchestrator (eNFVO). eNFVO handles and manages the lifecycle of IoT service functions through having an interface, named Mvo, with oneM2M-based IoT service platform. To manage the orchestration of MEC applications on a virtualized platform from the NFV's point of view, MEC ETSI ISG integrated the NFV orchestrator (NFVO) parallelly with the MEC Application Orchestrator (MEAO). The former interacts with the Operations Supports Systems (OSS)/Business Support System (BSS) [8] to support the orchestration and instantiation of *omInst* on the top of MEC host. MEAO maintains a catalog of MEC applications. Meanwhile, the Life Cycle Management (LCM) of the oneM2M MEC

IoT service as well as the configuration were assigned to the MEPM via the Mm5 interface.

D. oneM2M MEC IoT service

The oneM2M MEC IoT service is provided within its MEP after the *omInst* is running on the MEC host. oneM2M MEC IoT service may be discovered by other MEC IoT applications once registered and when the *omInst* access the MEP platform through the Mp1 interface. It features the concept of a broker (mediation gateway) for allowing other MEC-based IoT platform to get data upload by IoT devices to the *omInst*. It directs the request for resource discovery as well. In addition, oneM2M MEC IoT service can achieve the following scenario: a mediation gateway for interoperability, M2M devices (ASN, MN), and application entities are provided and managed. In this case, the oneM2M MEC IoT service should expose via the Mp1 interface the necessary APIs to discover oneM2M common functions resources and allow the interoperability with other IoT platform as well as the IoT application entity.

Based on the proposed architecture, two deployments scenarios of *omInst* would be featured: (i) *omInst* provides oneM2M resources for semantic interoperability through the oneM2M MEC IoT service (acting as a broker to ensure), (ii) *omInst* interacts with oneM2M entities without the oneM2M MEC IoT service. For both scenarios, the *omInst* relies on abstraction technique that, subsequently, masks the underlying technology by allowing bindings to different communications protocols and stacks (HTTP, CoAP, and MQTT). In the scenario (ii), the *omInst* would provide the necessary CSFs that allow to: discover components that analyze where requests originate; use the discover components for semantically annotated resources; retrieve and forwarded data whenever a new or changed data occurs. To semantically annotate resources, *omInst* could use semantic query function like the semantic query language SPARQL [9] for discovery criteria. The discovery criteria would start by `<oneM2M MEC-CSE>` in the tree resource structure.

After the IoT devices discover the presence of *omInst*, the oneM2M MEC IoT service would pass the requests from the IoT devices to the *omInst*. The latter accepts REST API and exposes an HTTP port. The oneM2M MEC IoT service Mp1 APIs is responsible for the following actions:

1) expose *omInst* (VNF) to IoT devices
2) handle the incoming requests from IoT devices and maps them into service calls,
3) return appropriate responses and exchange messages between oneM2M MEC applications and underlying IoT devices resources.

E. MEIX architecture in 5G 3GPP standard

The MEIX architecture complies to the definition introduced by 3GPP SA WG2 in TS 23.501 [10] that proposes a sort of mapping between ETSI MEC entities and 3PP entities. The MEIX architecture leverages the mapping realized in TS 23.501 so that the instantiation of virtual CSFs as a service on the MEC system can be deployed in 5G networks. Therefore, User Plane Function(s) that provides the interface to a Data Network correspond to some functionalities of the MEC data plane (virtualization infrastructure). The Application Function in 3GPP includes high-level architecture in charge of controlling traffic routing, access network capability. According to this definition, the logical AF in the 3GPP corresponds to some functions enabled by the MEC platform. In summary, the MEIX is built on top of 5G network components; particularly the Local Data Network seemingly design to support component that hosts oneM2M MEC as a service like other MEC applications. It is assumed that all MEC entities can be implemented as VNFs, and operate in the same NFV Infrastructure Point-of-Presence [10]. Meanwhile, the work [11] called T-NOVA is considered as a reference in this study to design the orchestration management.

F. Remote control in manufacturing IoT use case

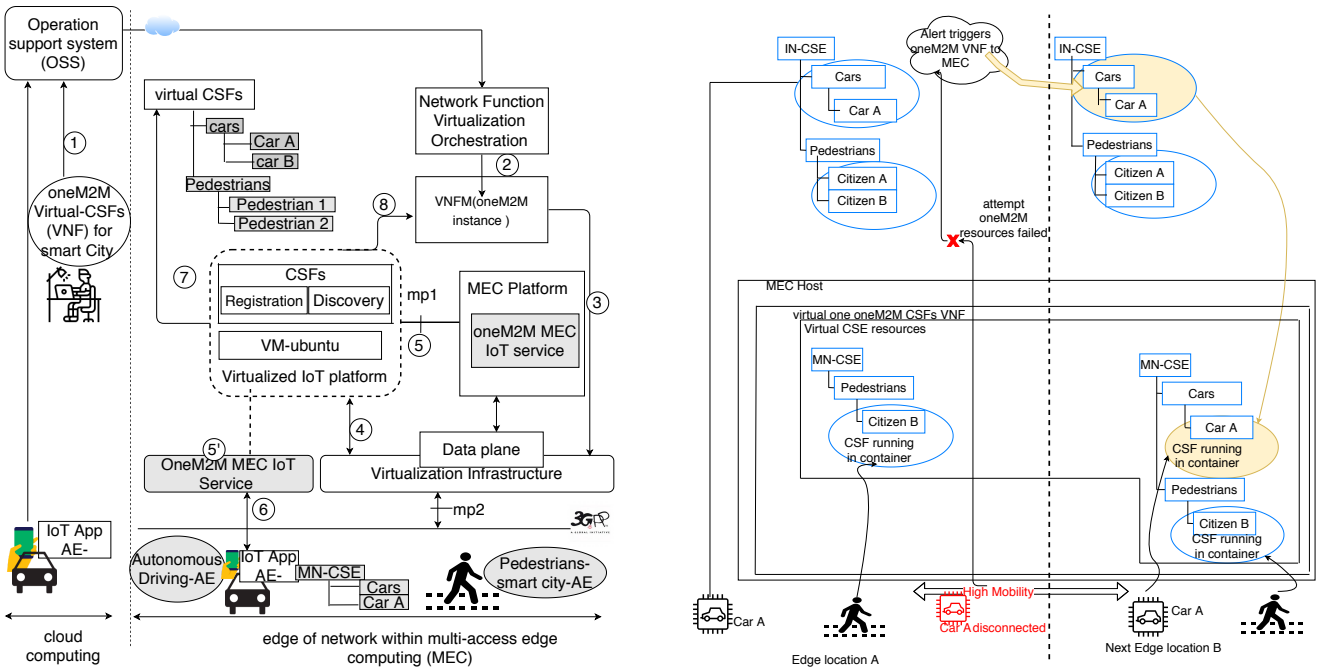
MEIX is beneficial for the IoT in manufacturing. Manufacturing (industrial) IoT envisioned to scale up different equipment located in different locations by using the remote production control solution. To reduce the latency while accessing the cloud resources, the manufacturing would specify to allocate the CSFs with the MEIX architecture. First, the company's manager at the remote control can define a set of Virtual oneM2M CSFs as VNF services to support analytics solutions. Next, the Virtual oneM2M CSFs are uploaded to the OSS system of the MEIX. The NFV and oneM2M orchestration instantiates the virtual oneM2M CSFs (omInst) on top of the MEC Host. Also, the manager can in real time monitor the computational usage of the omInst. The collection and analysis of the large-scale data sets are performed for supervising various field connected devices at lower latency compared to the use of IoT cloud-IoT platform. In summary, with MEIX, industrial IoT has the potential to handle proprietary devices since the oneM2M standard can interoperate with a wide range of networks and systems by providing necessary functions through API to the oneM2M server instantiated closer to the industrial's field.

IV. Value proposition of the MEIX architecture

The section discusses the value proposition of the MEIX architecture.

A. Virtual IoT CSFs as a service use case

This virtual IoT CSFs as a service facilitates the deployment of IoT applications customized to meet user requirements. As shown in Fig. 4a, a typical use case consists of an IoT platform based on oneM2M that provides oneM2M resource services to a tenant. The city council, as an IoT user consumer, selects type of virtual of CSFs tailored to meet the requirements of IoT applications (related to smart city) to different municipal domains, e.g., police, homeland security, public transportation companies during the step ①. The city council defines the policies for the IoT applications and VM computational resources through a Graphical User Interface (GUI). The GUI interface allows the IoT solution provider to define requirements in term of latency, bandwidth, type of



(a) Deployment of the Virtual IoT CSFs as a service.

(b) Synchronization of oneM2M with virtual oneM2M resources.

Fig. 4: Working flow of an use case under MEIX.

CSFs and computation resources. After this action, the virtual CSFs VNF file is uploaded to the OSS/BSS of the MEC management system. The NFV and orchestration module starts ②the provisioning of oneM2M virtual IoT CSFs resources.

The oneM2M MEC VNFM instantiates the *omInst* (VM) running on the top of the virtualization infrastructure ③④. Thereafter, *omInst* discovers the MEC platform by querying the MEC platform's service registry to expose oneM2M MEC IoT service ⑤. The oneM2M MEC IoT service is exposed as a broker to pass requests from IoT devices to the *omInst* ⑤'. The IoT ADN (here an autonomous driving application running over an oneM2M application entity) sends a request to the MEC platform in the process of accessing oneM2M CSFs through the *omInst*. The request is first forwarded to the oneM2M MEC IoT service ⑥. After checking the required CSFs and IoT application based on the request from IoT ADN, the oneM2M MEC IoT service delivers to the *omInst* to create resources of CSF that meet the request form the IoT ADN. The *omInst* has access to the CSF images registry to download the container images. After downloading the related container images, then *omInst* can initiate required virtual CSF image ⑦. Finally, the *omInst* exchanges life cycle information of newly resources of CSFs created to support requests from the IoT ADN ⑧. This information could help for mobility management to trigger the migration of the *omInst* while the IoT ADN goes out of the communication signal range. Then, IoT ADN can use virtual CSF services based on oneM2M MEC facilities.

When the car goes out of range, the mobility management strategy is deployed, it informs the *omInst* about the poor connection link information to the LCM management of the oneM2M VNFM to initiate migration of the running *omInst* to the next MEC platform. Network slice migration patterns have been discussed [6]. Based on slice mobility trigger proposed [6], group mobility trigger can be considered suitable since it can be applied to autonomous cars. The signal strength can be measured by the IoT ADN and reported back to the *omInst*, then the virtual CSF in charge of correlating measurement of signal strength will pull the group mobility trigger to follow the car until the low value of signal strength is reported. This low value of signal strength would trigger the VNFM of the *omInst* to start its migration. When the car is in range of a new base station, the virtual CSF in charge of device authentication after a successful migration of the *omInst* is used to identify the car. Therefore, the *omInst* would allow the synchronization only of oneM2M resources accordingly to the virtual oneM2M CSFs VNF being orchestrated on the MEC host. After going through these procedures, the *omInst* updates the application entity with new resources. Appropriate resources of *omInst* along with a set of traffic rules service provided by the MEC platform would enhance the continuity of resources to address the challenge of user mobility. Fig. 4b described the above scenario where the car is disconnected while oneM2M CSFs resources are running at the central IoT cloud. The pedestrians stay connected since the IoT application is accessing resources hosted at the network edge.

B. *Real time analytics functions*

Real-time data collection and analysis are crucial for massive IoT applications since a larger number of various sensors devices are deployed in the vicinity of edge computing nodes. In fact, the real-time analytical system from a multi-vendor poses a big challenge in case the real-time analytical system

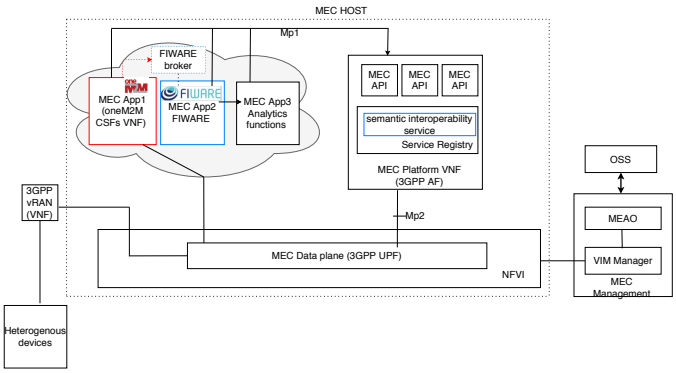


Fig. 5: Solution to the issue of interoperability with FIWARE for analytics functions.

cannot interpret accurately the format of data generated by the sensor devices. A solution is to allow the Global IoT Services (GIoTs) platform to be instantiated at the MEC host as an MEC application [12]. Another solution is using a mediation gateway to connect the FIWARE broker and oneM2M resources platform at the network edge. In that case, the MEC platform solves the issue of interoperability since the FIWARE and oneM2M platform can still exchange data using existing mechanisms of semantic interoperability as discussed in [12]. The use of virtual IoT resources would raise some issues such the synchronization with the IoT service at the cloud. Fig. 5 described the above real-time data collection where MEIX could help to solve the issue of interoperability with FIWARE for analytics functions.

C. Technical characteristics of the MEIX architecture

Enlivening the MEIX takes part in enhancing the interoperability of the existing MEC-based IoT platform. Thus, MEIX like MEC-based IoT platform [3] disclosed the following characteristics: orchestration framework; support communication protocols; application interoperability with another framework like FIWARE data models, multi-tenancy, and network slicing . Despite promising advantages of the MEC-based IoT platform for the commercial IoT platform, the MEC-based IoT platform still lacks the semantic annotations and Resource Oriented architecture (ROA) to uniquely access IoT resources. The MEIX can interoperate with the other multi-tenancy platform at the same infrastructure using a mediation broker (gateways). In addition, the MEIX being open and standardized would lead to a broad interoperability with well supported IoT platforms such as FIWARE. In fact, the Open Mobile Alliance Next Generation Service Interface for Context Management (OMA NGSI [13]) can implement open IoT APIs for MEC that could use the M2M functions exposed by the MEIX. Further, the ETSI ISG Context Information Management (CIM) can expose a mediation gateway service that features semantics and linked data to manage interoperability of different IoT platform requesting VNFs at the network edge so that the data models in the IoT world would not hinder the deployment of advanced IoT vertical cases.

D. Challenging Work and Open Issues

Although MEIX is a promising technology to cope with ultra-latency IoT applications, there are still some challenges that remain to be addressed in future work.

First, Service Capability Exposure Function (SCEF) APIs is now defined in 3GPP release15 TS 29.122 under the name T8 [14]. The T8 API provides access services of the 3GPP Network. However, ADN devices' connections to the oneM2M MEC instance are made over the 3GPP Network. For that, the interworking between the SCEF server and oneM2M MEC through Virtual CSFs VNF is an open issue.

Second, the NGSI-LD is a new protocol introduced with the formal agreement of OMA, which originally proposed by NGSI. The NGSI-LD aims to make it easier to find and exchange information with IoT platforms. Since in NGSI-LD, information not relevant to the application layer of that particular service is not explicitly considered, the flexibility of discovering and query information between NGSI-LD framework developed by ETSI ISG CIM and virtual CSFs VNF require the definition of domain-specific extensions to model type of information from oneM2M MEC instances such as details of the IoT or network technologies used to connect entities.

Finally, caching is a promising concept to achieve a trade-off between the storage and transmission rate [15]. The edge caching is applied to retrieve context information for reducing the content delivery latency. Hence, a new oneM2M CSFs about intelligent caching resources and cooperative caching policies at the network edge is required to be implemented beside to existing CSFs at oneM2M service layer. Therefore the edge caching policy likes a CSF (at the MEC Host) is crucial for improving the edge caching performance of virtual IoT resources deployed at the network edge.

V. Conclusion

In this article, we have analyzed the concept of instantiating the IoT service layer at the network edge to enhance the QoS of IoT applications. Based on the strong needs from the market for deploying advanced IoT applications, we have proposed MEIX architecture design and key components. IoT platform uses NFV technology so that virtualized IoT platform can run as a VNF for exposing virtual CSFs as a service at the network edge. The MEIX complements the existing MEC-based IoT platform since the latter lacks the ROA concept for identifying IoT resources uniquely.

Acknowledgment

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2018-0-01456, AutoMaTa: Autonomous Management framework based on artificial intelligent Technology for adaptive and disposable IoT).

References

[1] Q.-V. Pham, F. Fang, V. N. Ha, M. J. Piran, M. Le, L. B. Le, W. Hwang, and Z. Ding, "A survey of multi-access edge computing in 5G and beyond: Fundamentals, technology integration, and state-of-the-art," *IEEE Access*, vol. 8, pp. 116 974–117 017, 2020.

[2] Y. Shih, W. Chung, A. Pang, T. Chiu, and H. Wei, "Enabling low-latency applications in fog-radio access networks," *IEEE Network*, vol. 31, no. 1, pp. 52–58, 2017.

[3] L. Zanzi, F. Cirillo, V. Sciancalepore, F. Giust, X. Costa-Perez, S. Mangiante, and G. Klas, "Evolving multi-access edge computing to support enhanced IoT deployments," *IEEE Communications Standards Magazine*, vol. 3, no. 2, pp. 26–34, 2019.

[4] J. Swetina, G. Lu, P. Jacobs, F. Ennesser, and J. Song, "Toward a standardized common M2M service layer platform: Introduction to oneM2M," *IEEE Wireless Communications*, vol. 21, no. 3, pp. 20–26, 2014.

[5] R. Sanchez-Iborra, S. Covaci, J. Santa, J. Sanchez-Gomez, J. Gallego-Madrid, and A. F. Skarmeta, "MEC-assisted end-to-end 5G-slicing for IoT," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[6] R. A. Addad, T. Taleb, H. Flinck, M. Bagaa, and D. Dutra, "Network slice mobility in next generation mobile systems: Challenges and potential solutions," *IEEE Network*, vol. 34, no. 1, pp. 84–93, 2020.

[7] S. Arora, P. A. Frangoudis, and A. Ksentini, "Exposing radio network information in a MEC-in-NFV environment: the RNISaaS concept," in *2019 IEEE Conference on Network Softwarization (NetSoft)*, 2019, pp. 306–310.

[8] E. G. S. Group, "Multi-access edge computing (MEC);framework and reference architecture," ETSI, 2019. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/02.01.01_60/gs_MEC003v020101p.pdf

[9] E. Prud'hommeaux and E. A. Seaborne, "SPARQL query language for RDF," ETSI, 2008. [Online]. Available: https://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/

[10] G. T. 23.501, "System architecture for the 5g system (5gs)," 3GPP, 2017. [Online]. Available: http://www.3gpp.org/ftp//Specs/archive/23_series/23.501/23501-g20.zip

[11] M. Kourtis, M. J. McGrath, G. Gardikis, G. Xilouris, V. Riccobene, P. Papadimitriou, E. Trouva, F. Liberati, M. Trubian, J. Batallé, H. Koumaras, D. Dietrich, A. Ramos, J. Ferrer Riera, J. Bonnet, A. Pietrabissa, A. Ceselli, and A. Petrini, "T-NOVA: An open-source MANO stack for NFV infrastructures," *IEEE Transactions on Network and Service Management*, vol. 14, no. 3, pp. 586–602, 2017.

[12] E. Kovacs, M. Bauer, J. Kim, J. Yun, F. Le Gall, and M. Zhao, "Standards-based worldwide semantic interoperability for IoT," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 40–46, 2016.

[13] O. M. Alliance, "NGSI context management," Open Mobile Alliance, 2012. [Online]. Available: http://www.openmobilealliance.org/release/NGSI/V1_0-20120529-A/OMA-TS-NGSI_Context_Management-V1_0-20120529-A.pdf

[14] G. T. 29.122, "T8 reference point for northbound apis," 3GPP, 2018. [Online]. Available: https://www.3gpp.org/ftp//Specs/archive/29_series/29.122//29122--g20.zip

[15] H. Zhu, Y. Cao, X. Wei, W. Wang, T. Jiang, and S. Jin, "Caching transient data for internet of things: A deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2074–2083, 2019.