Evolving Multi-Access Edge Computing to support enhanced IoT deployments

Lanfranco Zanzi*, Flavio Cirillo*, Simone Mangiante†,
Vincenzo Sciancalepore*, Fabio Giust*, Xavier Costa-Perez*, Guenter Klas†
* NEC Europe Ltd. Germany † Vodafone Group R&D, UK
E-mails: {name.surname}@neclab.eu, {name.surname}@vodafone.com

***Abstract*—The Internet of Things (IoT) ecosystem is getting momentum as early deployments have proven their value, e.g. in smart cities, and more advanced use cases are being considered, e.g. automotive, public safety, e-health, etc. Such advanced use cases introduce new stringent requirements that can not be supported by current solutions both in terms of latency and/or computing power. In order to meet these requirements in a cost-efficient manner the *Multi-access Edge Computing* (MEC) paradigm is considered here as currently being defined by the ETSI MEC Industry Specification Group. In this paper we propose an ETSI-compliant MEC architectural solution that allows for seamlessly integrating existing and future IoT Platforms. In addition, an *IoT gateway middleware* is presented that enables running low-latency and/or computationally intensive applications on generalized MEC-based systems.**

I. Introduction

THE penetration of *Internet of Things* (IoT) deployments is advancing at an increasing pace as the technology matures and the cost of the required equipment benefits from economies of scale. Early deployments focused on basic monitoring and sensing applications, e.g., in smart cities. Based on the proven value of these solutions more advanced use cases are being considered, e.g., in the automotive, public safety, and e-health fields, with more stringent requirements in terms of latency, computing power and coupling to the network infrastructure.

Nowadays, the raw data produced by distributed networks of sensors is usually centrally processed and analyzed in data centers to derive added value information and, eventually, trigger the corresponding actions. In these deployments, the so-called *IoT Gateway* is a key entity acting as a mediator between field sensors and cloud data centers. IoT gateways act as a bridge for narrow-band communication protocols of (typically) energy-constrained networks, e.g., Bluetooth and ZigBee, and broadband — wireless and wired — systems, e.g., optical fibers or LTE, used as transport channel toward cloud facilities. But, they offer very limited processing capabilities for cost reasons given the large number of gateways required in real deployments.

In order to meet both the latency and/or computing power requirements of the upcoming advanced IoT-based use cases as well as to leverage the upgrades in mobile networking, in this paper we consider *Multi-access Edge Computing* (MEC) as the technology able to boost IoT to more sophisticated deployments. MEC is envisioned as a key technology to transition to the fifth generation (5G) mobile networks. The European Telecommunications Standards Institute (ETSI) has chartered the MEC Industry Specification Group[1] (ISG) in order to define a multi-vendor edge environment toward which IT and Telco stakeholders can converge.

MEC allows to dynamically install the applications of IoT services on top of cloud facilities at the edge of the network, thus with low communication latency. This way, IoT gateway functions like data pre-processing and *things* management can be lifted to the MEC platform, allowing to install cheaper hardware with simpler functionalitie. Moreover, MEC resources can be efficiently shared among different IoT networks — providing isolation guarantees — leading to unexplored business opportunities based on the novel concept of *Network Slicing* [1].

In the light of the above considerations, this paper brings the following contributions:

- Detailed review of a state-of-the-art solution for a smart city deployment, indicating its limitations and technical challenges when addressing advanced use cases.
- A *taxonomy of future IoT use cases* for next generation networks along with the corresponding derived requirements.
- A proposal for an *ETSI-compliant MEC architectural solution* to facilitate the integration of IoT networks and service deployments.
- An *IoT gateway middleware* enabling high-computational applications with low-latency requirements running on generalized MEC-based systems.

II. Smart cities as the key use case for large scale IoT deployments

Smart cities represent the first step toward a multi-domain and inter-connected IoT world [2], aiming at the digital transformation and interconnection of urban processes, like vehicular traffic control, utilities supply, health care, manufacturing, entertainment, etc. (see a self-explaining overview depicted in Fig. 1).

Sensed data from IoT networks are used to detect safety threats or to measure and guide human behaviors for different purposes (e.g., pollution reduction, energy savings, public safety in crowded events). Such data are transmitted by means of several technologies and might need to be quickly processed (low-latency) or fully explored (high computational power)

[1] http://www.etsi.org/technologies-clusters/technologies/multi-access-edge-computing

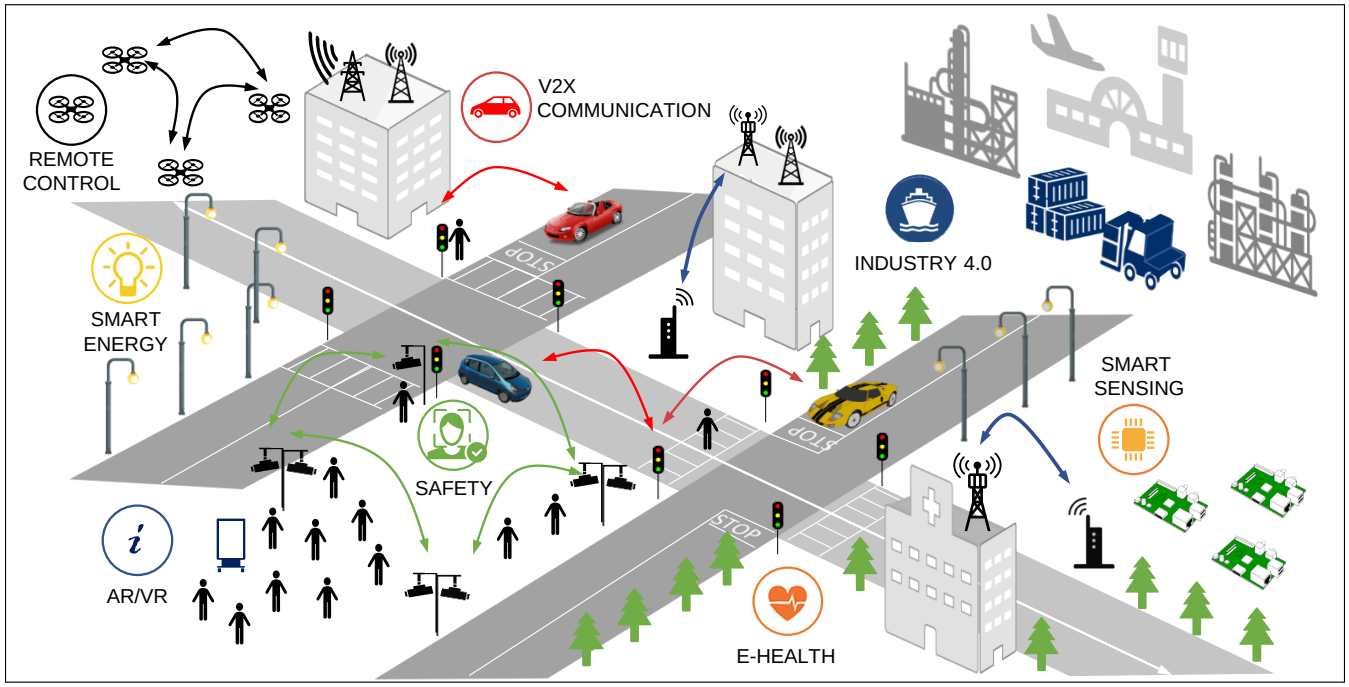

Fig. 1: IoT ecosystem in Smart City scenario.

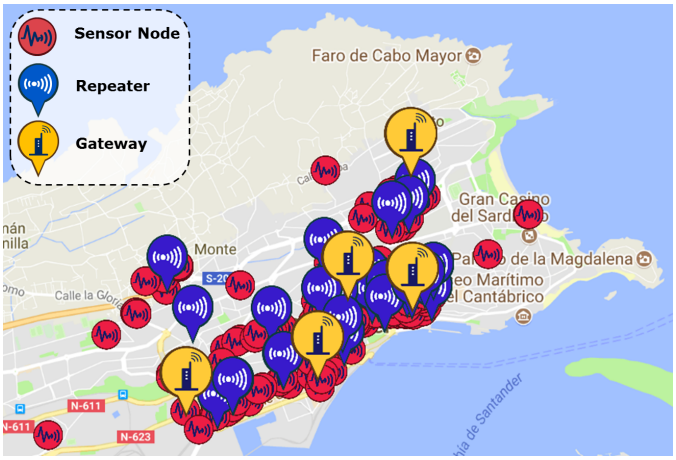

Fig. 2: Location of the IoT elements in the SmartSantander project.

based on different use cases. In the following we examine the existing smart city case-study in the context of the European project SmartSantander [3], showing deployed IoT use case features, current limitations and potential enhanced use cases.

A. The SmartSantander case-study

The deployment comprises 20000 static and mobile sensors installed within 35 square kilometers area in the city of Santander, Spain. The solution scope covers different purposes, such as environmental monitoring (for e.g., temperature and humidity checks for irrigation control of public garden) or vehicular monitoring (for e.g., traffic intensity and public parking management), as summarized in Table I. Recently to further deliver augmented reality services, 2500 RFID tags have been spread among the tourist attractions of the city.

The IoT network is built as a 3-tier architecture, composed of i) wireless sensors, ii) repeaters and iii) IoT gateways, as depicted in Fig. 2. The first two tiers communicate with each other via 802.15.4 interfaces whereas the third tier forwards the data coming from the low-rate and limited-power interfaces to the computational servers located within the city premises, by means of Wi-Fi, GPRS/UMTS or wired interfaces.

An interesting approach to create a big data analytics platform able to gather all data generated by the sensors of the SmartSantander deployment is represented by CiDAP [4], which requires computational features in cloud premises to expose collected information to external applications. However, the authors of this work point out that most of the sensor data are processed and analyzed in more than 60 seconds, making real-time applications (e.g., dynamic route calculation for ambulance based on real-time traffic information) unsuitable for this environment. Therefore, such applications are allowed to retrieve data directly from the IoT gateways requiring intelligence (and additional power) on such devices. Nonetheless, this approach does not satisfy low-latency requirements, mainly due to the absence of cross-layer optimization between transport and processing facilities typical of today's legacy solutions.

SmartSantander is an attempt, together with many others in Europe and in the rest of the world, to grow smart city capabilities and enable digital transformation. Nevertheless, smart cities are continuously evolving, aiming at incorporating future-looking use cases as those showcased in the next paragraphs.

B. A glimpse on future evolution of smart city use cases

The impelling need of getting high processing ability on the edge premises is further supported by a number of advanced

TABLE I: SmartSantander deployment summary (c.f. [3]).

Node Type		Amount of Nodes	Sensors	Radio I/F
Gateway	General Purpose	26	N/A	IEEE 802.15.4, IEEE 802.11, Digimesh, GPRS/UMTS
	Irrigation	3	N/A	
	Traffic	2	N/A	
Repeater	Temperature	74	Temperature, Acceleration	IEEE 802.15.4, Digimesh
	Light	553	Light, Temperature, Acceleration	
	Noise	58	Noise, Acceleration	
	Gases	13	Temperature, CO, Acceleration	
	Traffic	9	N/A	IEEE 802.15.4
	Weather	3	Temperature, Relative Humidity, Soil Moisture, Solar Radiation, Rainfall, Windspeed, Atmospheric Pressure, Acceleration	
	Irrigation	23	Pluviometer and Anemometer sensing temperature, relative humidity, soil moisture, soil temperature,	
	Water Flow	2	Water Flow, Acceleration	
	Agriculture	19	Temperature, Relative Humidity, Acceleration	
Parking Sensors and Tags		723	Ferromagnetic sensors buried under the asphalt for occupancy and authorization	Proprietary
Traffic Sensor		59	Road Occupancy, Vehicle Counting, Vehicle Speed Monitoring	IEEE 802.15.4
Mobile Node	Bus	95	CO, Particles, NO_2, Ozone, Temperature, Relative Humidity, Speed, Odometer, Location	IEEE 802.15.4,IEEE 802.11, GPRS
	Car	80		GPRS
Augmented Reality Tag		2500	Presence (+ metadata)	NFC
Participatory Sensing Smartphone		6500	Multiple	IEEE 802.11, GPRS/UMTS
Augmented Reality Smartphone		∼14000	Presence (+ metadata)	
Total:		**31 Gateways 1516 Fixed Nodes 175 Mobile Nodes 2500 Tags**	**3029 Fixed Sensors 1750+ Mobile Sensors 20000+ Smartphone Sensors**	

IoT use cases, which are envisioned to underpin future smart cities. In particular, we explore i) autonomous (and remote) driving and advanced traffic monitoring, ii) public safety and assistance of large crowds, iii) industrial automation scenarios, shedding the light on feasibility aspects and future requirements.

1) Augmented context awareness for autonomous driving and road safety: A huge number of sensor networks are already deployed along pedestrian and vehicular roads, such as monitoring cameras, traffic and visibility sensors. These systems are usually single-purpose platforms deployed in different time periods belonging to few verticals, representing independent environments that hardly combine in a single homogeneous platform due to vendor-specific features. In order to address novel and advanced scenarios, measurements from a network of sensors must be treated seamlessly together with information coming from other platforms in the same contexts. Two applications of this data melting-pot are augmenting the context-awareness of autonomous driving systems and enhancing emergency assistance for manned driven vehicles. A real deployment of such applications would require external sensors with an holistic view of the environment to achieve state awareness and perform traffic balancing and routing optimization, a joint combination of local and remote video analysis for identifying unexpected obstacles (e.g., pedestrians or animals crossing the road), sensor fusion algorithms for inferring a myriad of diverse situations in a complex scenario like the urban environment, together with a reliable alerting system able to reach the driver even in unfavorable conditions.

Another use-case, with even more challenging requirements, is the introduction of automated driving solutions on highways. Once again, this kind of solution must be able to collect and analyze data coming from heterogeneous sources, starting from global traffic video streams and ending with the sensor co-located with the vehicles. An example of piloting those kind of scenarios in the real world is the European project AUTOPILOT [2] that aims to enhance the safety of automated driving with the means of surrounding smart objects. Obtaining real-time information about the overall context state is crucial to handle automated vehicles moving along with human driven vehicles and other vulnerable road users such as cyclists and pedestrians. Automated vehicles are requested to precisely identify and possibly predict complex situations and quickly react without any human input. This would require high computation resources but, at the same time, very low latency between the detection of the issue and the corresponding alarm delivery. Matching both requirements in nowadays systems is tough given the limited computational power of programmable gateways that necessarily delegate heavy processing tasks to remote data centers, further increasing the experienced traffic latency.

2) Public safety in multi-domain smart cities: Typical applications of IoT in smart cities are related to public safety. Piloting projects, such as MONICA[3], aim to launch a series of large and small security applications during big public events. An example is the usage of digital signage (e.g., advertisement displays or projectors) to steer crowds in case of emergency situations like fires or flash floods. Computing expensive sensors fusion algorithms would be used to identify the exact location of the danger and infer its future development. At the same time, estimating the crowd distribution and its mobility behavior is crucial in order to derive an optimal rescue strategy, counting on location information gathered from personal devices and global monitoring systems. Finally, the computation output needs to arrive well-timed at the distributed network of actuators in order to orchestrate the displays and steer the crowd to safe places.

Another application is the support to security services during big events to handle potential threats. Also in this case, the computation and latency demands are unlikely to be met

[2]http://autopilot-project.eu/.

[3]http://www.monica-project.eu/

when the raw data traffic is characterized by very localized and unpredictable traffic peaks, like in mission critical situations or occasional public events. Many public entities support the *open data* concept, which consists of exposing sensor data to anyone who wants to leverage them for smart applications. Even though this would help to raise a positive IoT ecosystem, most of the information collected by sensors networks is sensitive and can not be openly shared. On the one side, the setup of IoT networks involves complex bureaucracy which forces specific data typologies to be stored in trusted physical locations (e.g., locally in the municipality premises), rather than on remote uncontrolled data centers. On the other side, a fully distributed approach based on powerful devices deployed in the urban context locally running applications opens new safety issues. For example, in facial recognition applications [4], softwares and complementary datasets, e.g., target databases, are often protected by secret. Running this kind of applications directly on top of cameras which might be directly exposed to vandalism attacks may not be a good solution.

3) System integration for large scale industrial automation: Industrial IoT (IIoT) and the related term "Industry 4.0" refer to the automation of modern factories operating a large number of connected smart resources like robots and sensors. The focus of IIoT is on the exchange and real-time control of mission critical information: in fields like energy, oil and gas, health care, reliability and accuracy are not optional. Machines must monitor physical processes and react often through decentralised and autonomous decisions.

Factory operators have already started to use analytics and machine-learning algorithms to predict consumption of raw materials and optimize their process control and supply chains in real time. In order to achieve proactive maintenance, many industrial settings seek technologies that enable real-time monitoring, anomaly detection and alerts, failure prediction, and predictive servicing of critical equipment. In an IIoT system a delayed response, or even a network interruption, causes data loss leading to unsynchronised, un-optimised processes and loss of money. With a high volume of sensitive data to be processed and analysed in real time, tactical decisions must be made locally where they matter and data security and reliability emerge as keystones of a robust solution.

While a self-contained IIoT may be easily designed and tailored for a single, isolated factory, meeting all its specific requirements, the same cannot be said in larger industrial environments where multiple IIoT systems with different requirements must be interconnected and orchestrated. A harbour is an example of such a heterogeneous ecosystem:

- multiple tasks must be performed and synchronised over a geographically wide area: ship mooring, containers management, control tower operations, freight transport logistic, etc.
- each task needs specific sensors and requirements, which are often conflicting: precision and reliability for detecting and moving containers, bandwidth and computation to process content from video surveillance cameras, low latency and real-time response for monitoring sea and ship conditions, etc.
- a single connectivity technology cannot guarantee enough coverage or performance for all tasks, therefore different technologies (e.g. Bluetooth, WiFi, LoRa, LTE) must be used in different locations.

Most big and medium-sized ports (e.g. Amsterdam, Valencia) are running "Smart Port" initiatives like the H2020 project Inter-IoT [5] facing similar problems: data and assets belong to different owners (shipowners, terminals, port authorities, etc.) with diverse technical/business requirements and collecting technologies; resulting information must be securely shared and processed in real time in order to improve overall mission critical operations. A Smart Port architecture relies on an IIoT platform to manage data from sensors and devices, and distribute that data to other stakeholders' components in real time. Such platform cannot cover all sensors and business needs in a single element, due to complexity and cost, but as a set of systems integrated through a central common system.

III. The advantages of deploying IoT at the edge

Future smart cities are meant to exhibit evolved features that leverage three technology pillars: i) low latency communications, ii) high computing capabilities and virtualization, and iii) heterogeneous access technologies and devices, as depicted in Fig. 3.

Because of such aspects, the network's edge is already widely recognized as a favorable location to deploy computing capabilities, and the industry and scientific community are already working on different solutions [5]. Low latency and high computing capabilities are necessary for real-time automated control systems wherein the event-decision-enforcement-feedback loop needs to be executed in a short, constrained time budget. In these regards, *fog computing* enables data processing and application logic to run at fog nodes scattered in the network, including end devices, edge and cloud resources. Many solution providers have gathered in the OpenFog Consortium [6] to define an open architecture for fog computing [6]. OpenFog is not meant to be a standard developing organization, but still its contribution is relevant to understand use cases and to foster their implementation. The airport scenario in [6] is an interesting example, which, similarly to what we have introduced previously about sea ports, shows how interconnected heterogeneous systems represent an increasing trend in the industrial IoT community. Cloud computing is essential to overcome the power and form factor limitation of IoT devices, by offloading data analysis and processing tasks to remote application servers. By virtualizing such application servers, it is possible to flexibly deploy them over different platforms and even to relocate them if necessary. Major cloud computing providers have expanded their offer with a custom IoT solution, as for instance in the case of Azure IoT Suite [7] and AWS IoT [8]. Similarly to remote clouds, edge computing platforms lend themselves to run a

[4]We refer the reader to an interesting newspaper article "British police arrest suspect spotted with facial recognition technology" available at http://www.telegraph.co.uk/technology/2017/06/07/british-police-arrest-suspect-spotted-facial-recognition-technology/.

[5]http://www.inter-iot-project.eu/

[6]https://www.openfogconsortium.org/

[7]https://azure.microsoft.com/en-us/suites/iot-suite/

[8]https://aws.amazon.com/iot/

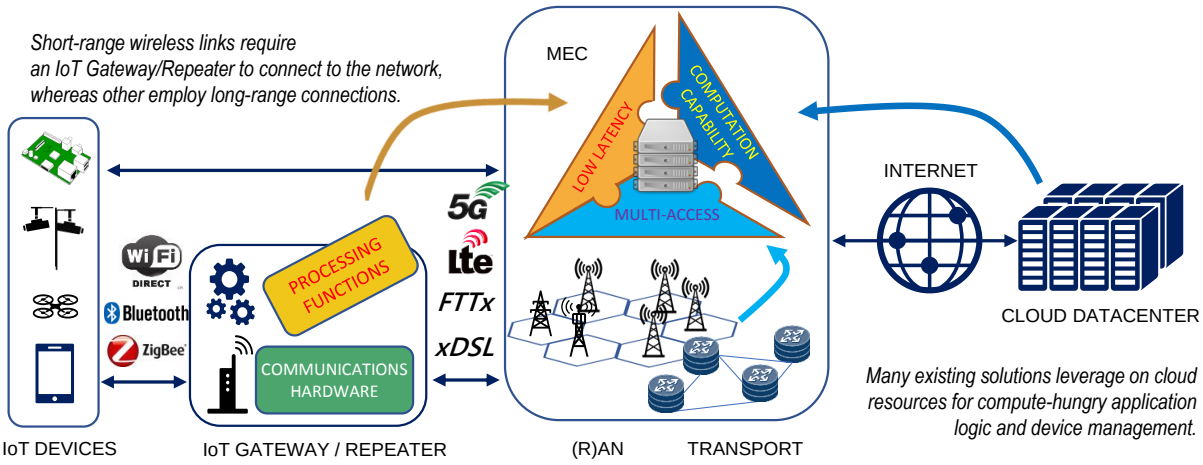

Fig. 3: A MEC-based IoT solution: Building blocks overview.

variety of IoT applications, but avoid the tromboning effects when transferring data meant to be generated and consumed locally. This aspect is tackled, among others, by Cloudlet Applications [9], AWS GreenGrass [10] and Azure IoT Edge[11].

The technologies above belong to a broad set of proprietary solutions that tend to focus on application level and service hosting rather than on the communication technologies underneath. In fact, the IoT ecosystem nowadays comprises a plethora of communication technologies, spanning from short range wireless like Bluetooth, Zigbee, WiFi to Low Power Wide Area Network (LPWAN) links using NB-IoT, LTE-MTC, EC-GSM-IoT and others. Some of them are based on industrial standards by IEEE and/or 3GPP whereas others are proprietary solutions, e.g., Sigfox and LoRa.

Nevertheless, in many situations it is paramount to leverage different deployments, either to aggregate data from distinct domains, or to merge legacy setups with newer ones into a single logical service. Therefore, it is necessary to abstract from the access technology underneath. The authors of [7] suggest to employ Software Define Networking to route data packets between the IoT device and the most appropriate fog node running the data filtering algorithms and the application's logic. Bringing IoT software components to the edge is the conceptual basis of EdgeX Foundry [12], which is an open source project purposed to implement an open framework for IoT edge computing. Although targeting a broader scope than IoT, a similar endeavor is attempted by the Open Edge Computing initiative [13] and by the Edge Computing Working Group within the Telecom Infra Project [14].

Because of its widely recognized position in the industry, the ETSI MEC Industry Specifications Group is sensibly regarded as a key entity to produce the enabling technologies for network operators to evolve their IoT service offering [8]. With the recent expansion to cover heterogeneous access, ETSI MEC provides a foundation not only able to support the three technology pillars in the opening of this section, but also outlines an open and standardized path for telco operators, vendors and IT players. For this reason, in the next we propose an IoT platform for ETSI MEC, which aims to consolidate different IoT technologies into a single body able to expose a homogeneous IoT service to customers in a multi-tenant fashion.

IV. ETSI MEC Enhancements to support multi-domain IoT deployments

In order to make edge clouds a standardized computing environment, ETSI has recently re-chartered the ISG formerly known as Mobile Edge Computing: the scope of the new Multi-access Edge Computing ISG is enlarged to embrace a variety of access technologies beyond cellular. The most relevant outcome of ETSI MEC is the definition of a framework and reference architecture [9], as well as a number of specifications for application enablement [10] and API design [11]. [15]

However, despite IoT being deemed a pivotal use case for MEC, the ETSI solution still lacks the due level of details when it comes to the components that are supposed to support IoT use cases. We hence propose a new architectural element for extending MEC capabilities to support IoT deployments, namely the MEC IoT platform described in the next section.

A. The MEC IoT platform

The MEC IoT platform is a software artifact meant to create a substrate (or *middleware*) where multiple (virtualized) IoT gateway instances, from different access link types and implementations, can run. This can be achieved in two steps. First, IoT gateways are split into lower (hardware) and upper (software) layers and the latter is migrated to the MEC facilities, hosted as software instances within the

[9]https://cloudlets.akamai.com/
[10]https://aws.amazon.com/greengrass/
[11]https://azure.microsoft.com/it-it/campaigns/iot-edge/
[12]https://www.edgexfoundry.org/
[13]http://openedgecomputing.org/index.html
[14]http://telecominfraproject.com/project/access-projects/edge-computing/

[15]Interested readers may find the full list of ETSI MEC specifications from the ISG's website http://www.etsi.org/deliver/etsi_gs/MEC/001_099/.

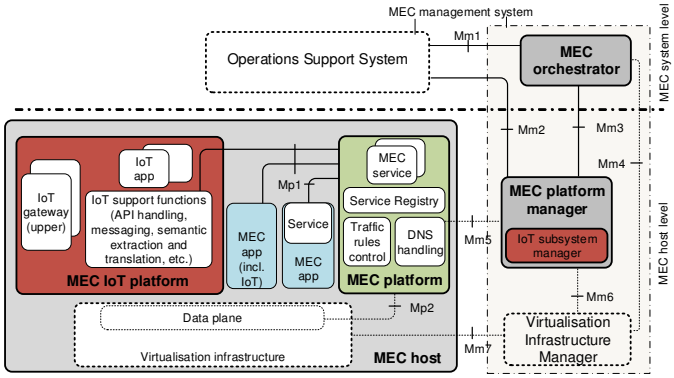

Fig. 4: The MEC IoT platform within the ETSI MEC architecture.

MEC IoT platform. In Section III, we have already observed few (commercial) implementations of such softwarized IoT gateway approach. The second step would be to implement the middleware functionalities of the MEC IoT platform as an environment where the IoT gateway instances are interconnected, by means of an appropriate messaging service, a gateway instance registration and discovery mechanism, and a semantic extraction and protocol translation function. The purpose is to unify into a single logical entity the operations that are typically executed by IoT Gateways, such as identification and management of IoT devices and their secure communication. By hiding the complexity of the underneath IoT networks, the MEC IoT platform can expose a homogeneous method to control a variety of IoT devices grouped into a single logical set, comprising diverse deployments.

From the perspective of the ETSI MEC architecture, despite its internal complex logic, the MEC IoT platform would appear as a service provider MEC application, installed in a MEC host and enabled by the MEC platform through Mp1 interface [10].

In other words, MEC applications can discover the MEC IoT platform by querying the MEC platform's service registry, and interact through a defined IoT API exposed by the MEC IoT platform. This abstraction and the associated API would enable MEC applications to interact with deployed IoT devices with little or no knowledge of the actual deployment, thus hiding the complexity of the IoT network from end applications. The diagram in Fig. 4 depicts the IoT platform in the ETSI MEC reference architecture (simplified - the detailed architecture is available in [9]).

The ultimate task of the MEC IoT platform is to enable multiple and different IoT services by exposing the capabilities of a set of IoT devices like sensors and actuators to either built-in or MEC applications. Therefore, the MEC IoT platform needs to be populated with the appropriate IoT gateway instances and the set of associated devices. In addition, it should be configured with the traffic filters and policies to allow shared usage of the framework. An IoT subsystem manager is devised to perform these jobs. This logical entity may be integrated in the same MEC platform manager, which is already providing element management functions, as depicted in Fig. 4. Such a deployment is desirable for instance to enable the MEC management system to orchestrate an IoT service when distributed across multiple MEC hosts. However the management interfaces defined by ETSI MEC, namely Mm1, Mm2 and Mm3, and specified in [12], [13] do not support IoT-specific functionalities and should be extended for such purpose.

The MEC IoT platform is able to decouple the set of IoT devices from the application logic that leverages the capabilities of such devices (sensors and/or actuators). This enables to partition (or share) the IoT resources thereby allowing a simultaneous usage following the IoT-as-a-service paradigm. Therefore, the multiple services running within the MEC premises might utilize a higher level of data abstraction together with a common API and data format representation. Running services also need to interact with the deployed nodes and sensors for performing IoT tasks, such as device management or actuation. In order to realize a seamless interaction among service-to-service and service-to-device, it is desirable to create a single interface (the aforementioned IoT API for MEC). In addition, this API being open and standardized would lead to a broad and well supported ecosystem. All these characteristics might be fulfilled by the Open Mobile Alliance Next Generation Service Interface for Context Management (OMA NGSI [14]) and the FIWARE foundation, which adopted it and have developed components for several aspects in the field of IoT, cloud and edge analytics, and privacy and security [15]. In addition, the work of the NGSI standard is continuing (within the ETSI ISG Context Information Management – CIM[16]) in the direction of semantics and linked data in order to achieve high interoperability with the many data models present in the IoT world.

V. Value proposition of the MEC-based IoT Platform

The low-latency and high-computational features are blended together when the IoT subsystem is abstracted, installed and orchestrated in the ETSI MEC-based system. The MEC platform offers an IT environment where the IoT gateways can be softwarized and deployed for advanced use cases. This enables to build a seamless and multi-domain IoT platform where different technology sources are interconnected by means of standardized interfaces, but, at the same time, they can interact using common languages.

The advantage is two-fold: on the one hand the IoT service providers might easily migrate their own IoT applications to more powerful (and close to the edge) IT environments without loosing direct control over their private (and reserved) applications data; on the other hand, the edge cloud facilities (e.g., MEC hosts) can be directly deployed (and managed) by the IoT service provider.

This case might be generally envisioned as the business scenario wherein the IoT service provider becomes an IoT infrastructure provider (e.g., building on top of public or private infrastructures, such as airport areas, shopping malls or building blocks) able to open their own premises to other IoT service providers. However, this would require a multi-tenancy-enabled IoT platform where different "tenants" are

[16]Available online at http://www.etsi.org/news-events/news/1168-2017-02-news-etsi-new-group-on-context-information-management-kick-off-meeting

TABLE II: Qualitative analysis against commercial solutions.

	Orchestration	Inter-operability	Open Access	Multi-tenancy	QoS at the edge
MEC-based IoT Platform	✓	Fiware / OMA NGSI	✓	✓	Network Slicing
Azure IoT Edge	Proprietary	Proprietary	✗	✗	✗
AWS Greengrass	Specific to AWS Lambda functions	MQTT	✗	✗	✗

willing to rent part of the same shared IoT infrastructure, tailored to their own provided services.

Therefore the MEC system brings into play the network slicing concept as the main enabler for a multi-tenancy platform. Heterogeneous IoT application requirements might be handled on the same platform assuring at the same time service-level-agreement (SLA) guarantees (e.g., public safety deployments for monitoring crowded events). Moreover, the data isolation property can be ensured among competing IoT service providers.

We have summarized the main advantages and limitations of our proposed solution in Table II comparing against commercial IoT deployment solutions, such as AWS Greengrass and Azure IoT Edge. Our proposal presents a full-fledged, open orchestration solution for an IoT platform by means of ETSI MEC-compliant interfaces. In particular, the OMA NGSI language can be used to realize a lightweight and efficient communication between IoT devices, compared to proprietary and Message Queue Telemetry Transport (MQTT) protocols. In addition, the MEC-based IoT platform evolves towards an open model when exposing collected data to the application layer. To the best of our knowledge, this is the first, non-proprietary ETSI MEC-compliant solution integrating the IoT facilities into edge computational nodes.

VI. Conclusions

The IoT market segment represents a huge opportunity for the involved stakeholders as the number of use cases expand both in terms of application areas and complexity. In this paper, we have analyzed the main limitations of an existing smart city deployment – SmartSantander case study – and derived future requirements for advanced use cases, such as autonomous driving, public safety and industrial IoT. Based on such requirements, we have proposed an ETSI MEC-based architecture to seamlessly integrate existing and future IoT platforms along with the required interfaces and protocols to enable communication between multi-technology sensors and IoT gateways through an IoT gateway middleware.

References

[1] V. Sciancalepore, F. Cirillo, and X. Costa-Perez, "Slice as a Service (SlaaS): Optimal IoT Slice Resources Orchestration," in *IEEE Global Communications Conference (GLOBECOM)*, Dec. 2017.

[2] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, Feb 2014.

[3] S. Krco et al., "D5.8 Final Testbed Manual," SmartSantander, Tech. Rep., Oct. 2014.

[4] B. Cheng, S. Longo, F. Cirillo, M. Bauer, and E. Kovacs, "Building a big data platform for smart cities: Experience and lessons from santander," in *IEEE International Congress on Big Data*, June 2015, pp. 592–599.

[5] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A Survey on Mobile Edge Networks: Convergence of Computing, Caching and Communications," *IEEE Access*, vol. 5, pp. 6757–6779, 2017.

[6] OpenFog Consortium Architecture Working Group, "Openfog reference architecture for fog computing," Tech. Rep., Feb 2017. [Online]. Available: https://www.openfogconsortium.org/wp-content/uploads/OpenFog_Reference_Architecture_2_09_17-FINAL.pdf

[7] X. Sun and N. Ansari, "EdgeIoT: Mobile Edge Computing for the Internet of Things," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 22–29, December 2016.

[8] D. Sabella, A. Vaillant, P. Kuure, U. Rauschenbach, and F. Giust, "Mobile-Edge Computing Architecture: The role of MEC in the Internet of Things," *IEEE Consumer Electronics Magazine*, vol. 5, no. 4, pp. 84–91, Oct 2016.

[9] ETSI MEC ISG, "Mobile Edge Computing (MEC); Framework and reference architecture," ETSI, DGS MEC 003, April 2016.

[10] ——, "Mobile Edge Computing (MEC); Mobile Edge Platform Application Enablement," ETSI, DGS MEC 011, July 2017.

[11] ——, "Mobile Edge Computing (MEC); General principles for Mobile Edge Service APIs," ETSI, DGS MEC 009, July 2017.

[12] ——, "Mobile Edge Computing (MEC); Mobile Edge Management; Part 1: Part 1: System, host and platform management," ETSI, DGS MEC 010-1, October 2017.

[13] ——, "Mobile Edge Computing (MEC); Mobile Edge Management; Part 2: Application lifecycle, rules and requirements management," ETSI, DGS MEC 010-2, July 2017.

[14] Open Mobile Alliance, "NGSI Context Management," OMA, OMA-TS-NGSI Context_Management-V1_0, May 2012.

[15] FIWARE Consortium, "FIWARE GE Open Specification," IoT Chapter D5.1.2, May 2013.