Edge Computing:
from standard to actual infrastructure deployment and software development

White Paper

October 2019

Authors: Dario Sabella, Andrew Alleman, Ellen Liao, Miltiadis Filippou, Zongrui Ding, Leonardo Gomes Baltar, Srikathyayani Srikanteswara, Krishna Bhuyan, Ozgur Oyman, Gershon Schatzberg, Neal Oliver, Ned Smith, Sharad D. Mishra, Purvi Thakkar.

1 Introduction

Edge Computing is widely recognized as a key technology, supporting innovative services for a wide ecosystem of companies, ranging from operators, infrastructure owners, technology leaders, application and content providers, innovators, startups, etc.

The advent of 5G systems has also increased the attention to Edge Computing [1][2], as this technology supports many of the most challenging 5G use cases, especially those for which high end-to-end (E2E) performances are required. In fact, due to the presence of processing platforms and application endpoints in close proximity to end-users, Edge Computing offers a low latency environment and significant gains in terms of network bandwidth, which are key benefits for the deployment of 5G networks.

Edge computing can span a variety of network locations, form factors, and functions, as depicted in Figure 1 below. Centralized computing is performed deeper into the network/cloud, with applications addressing a large number of users, and edge platforms hosting multiple applications simultaneously. In contrast, distributed computing [3] takes place in proximity to end users, with applications being more attuned to specific endpoints and functions. Depending on the various deployment options considered, edge applications can be characterized by spatial and temporal proximity to clients, real-time responsiveness, interactivity, and mobility, and are well-suited for use cases such as industrial control, video analytics, interactive (XR) media and healthcare use cases.

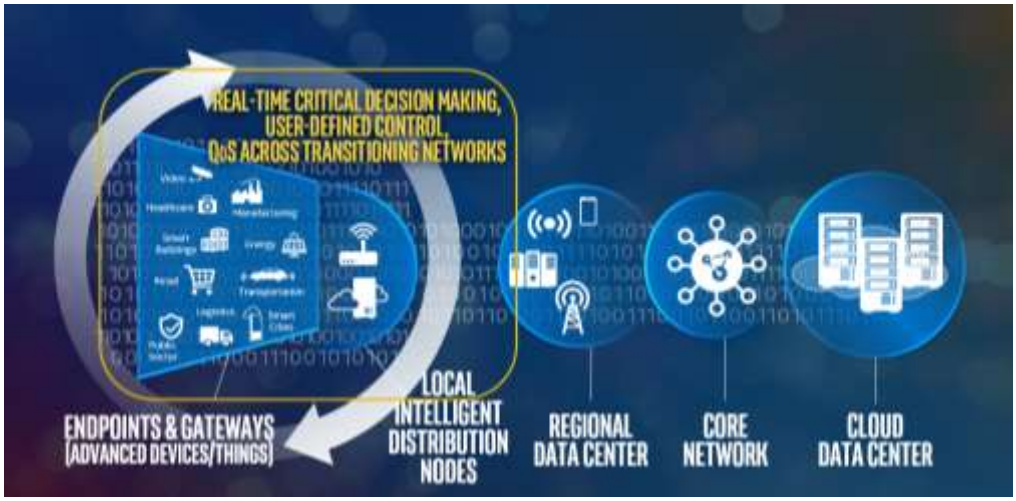

Figure 1 - Edge Computing spanning a variety of network locations, form factors, and functions

The area of Edge Computing (often termed after as “MEC” - Multi-access Edge Computing in this paper, to emphasize its access-agnostic nature) falls broadly within the scope of various initiatives, ranging from standardization bodies and industry groups and associations to open source communities and projects.

The aim of this white paper is to provide an overview of these initiatives, together with some Intel guidelines, to help the reader to better understand the various challenges for a real deployment of Edge Computing. The focus is to mainly address the needs of two stakeholder categories in the MEC ecosystem: **infrastructure owners** (e.g., operators and cloud providers) and **software developers** (e.g., applications / content providers, innovators and startups). In fact, both stakeholder categories are key to the success of MEC, and their engagement depends also on the way the various challenges are addressed.

In particular, this paper aims to:

- provide an overview of standardization efforts, including initiatives from industry groups, associations, open source communities and projects (e.g., Open Network Edge Services Software (OpenNESS)), to highlight the key deployment findings from an infrastructure point of view (especially for the MEC deployment in 5G systems), as well as to draw a coherent mapping from a SW development perspective (e.g. including all components needed to build an edge cloud, according to the current initiatives in the space);
- derive some deployment considerations related to Edge Computing (for the first category of MEC stakeholders), with special emphasis on the recurrent question "Where is the edge?", complemented by a comparative analysis of the different deployment options, together with a suitable performance evaluation in few use cases of interest;
- describe an exemplary case study customized to use cases relevant to the automotive vertical segment, thus, offering to the reader an E2E example involving topics such as orchestration, security and onboarding. Such an example aims to constitute a useful reference for the software developer (i.e., the other main category of MEC stakeholders), when it comes to answering to the other recurrent question "How to use MEC from application development point of view?").

The overall market success of Edge Computing will depend, in fact, on how infrastructure owners will be able to address all deployment aspects and also on how software communities will create a wide set of applications and innovative services. Intel is committed to engaging the entire ecosystem, considering standard solutions and open source projects, towards ensuring interoperability, while opening the market to proprietary implementations and added-value propositions.

2 Edge Computing: standardization efforts, industry groups and open source initiatives

Edge Computing incorporates the benefits of virtualization and cloud computing to enable high-powered computing capabilities as close as possible to subscribers. The set of use cases supported by this technology is very wide, and spans many vertical market segments, e.g., transportation, connected cars, industrial automation, retail, healthcare and social assistance, media & entertainment, use cases for smart cities and internet of things, and others. The market potential for Edge Computing is huge[1], nevertheless, actual deployment of this technology depends on both its maturity and its specified definition by relevant standardization bodies, industry groups and open source projects. For this reason, the aim of this section is to present the main initiatives in this area from different angles and perspectives, to analyze complementarities and synergies, and also potential overlaps and gaps. Finally, a functional mapping depicts these activities by projecting them onto the various elements of the edge architecture, as a possible guide for all stakeholders, i.e., both application developers and infrastructure owners. In fact, to avoid market fragmentation, a coherent approach should be adopted by stakeholders, to align implementations and enable interoperability, also by reducing costs.

2.1 The standardization landscape

Standardization in the area of Edge Computing started in ETSI at the end of 2014 with the creation of the Industry Specification Group (ISG) on MEC [4]. Currently, ETSI ISG MEC is still the only international standard available in the technology field, however, new emerging initiatives have started in 3GPP aiming at the integration of MEC in 5G systems. Other Standards Developing Organizations (SDOs) have also started working around Edge Computing from different perspectives/angles (e.g., the Internet Engineering Task Force - IETF, for security aspects). The following overview is limited to current standardization efforts, and will be complemented by subsequent sections on other initiatives, such as industry groups and open source initiatives.

2.1.1 ETSI ISG MEC

According to the ETSI ISG MEC definition, MEC offers to application developers and content providers cloud-computing capabilities and an Information Technology (IT) service environment at the edge of the network. This definition is quite compact and summarizes the main technologies introduced by ETSI for Edge Computing: cloud-computing capabilities and an IT service environment. In fact, MEC does not only introduce cloud computing at the edge of the network, but also offers the capability to expose edge services to application developers.

Following the developments of the ETSI MEC standard, MEC provides an environment in proximity to the end user, ultra-low latency, high bandwidth, and access to added value information through MEC

[1] https://www.ericsson.com/en/blog/2018/9/edge-computing-successa-distributed-cloud-approach

Application Programming Interfaces (APIs), such as real-time access to access network, context information, location awareness and others. MEC will permit to open the cloud infrastructure to operators, service providers and third parties, i.e., application developers and content providers, helping to meet the demanding Quality-of-Service (QoS) requirements of new 5G systems. Due to its access-agnostic nature, MEC guarantees a smoother deployment, independent of the underlying Radio Access Network (RAN) (e.g., Long Term Evolution (LTE), 5G New Radio (NR), or other non-3GPP radio access).

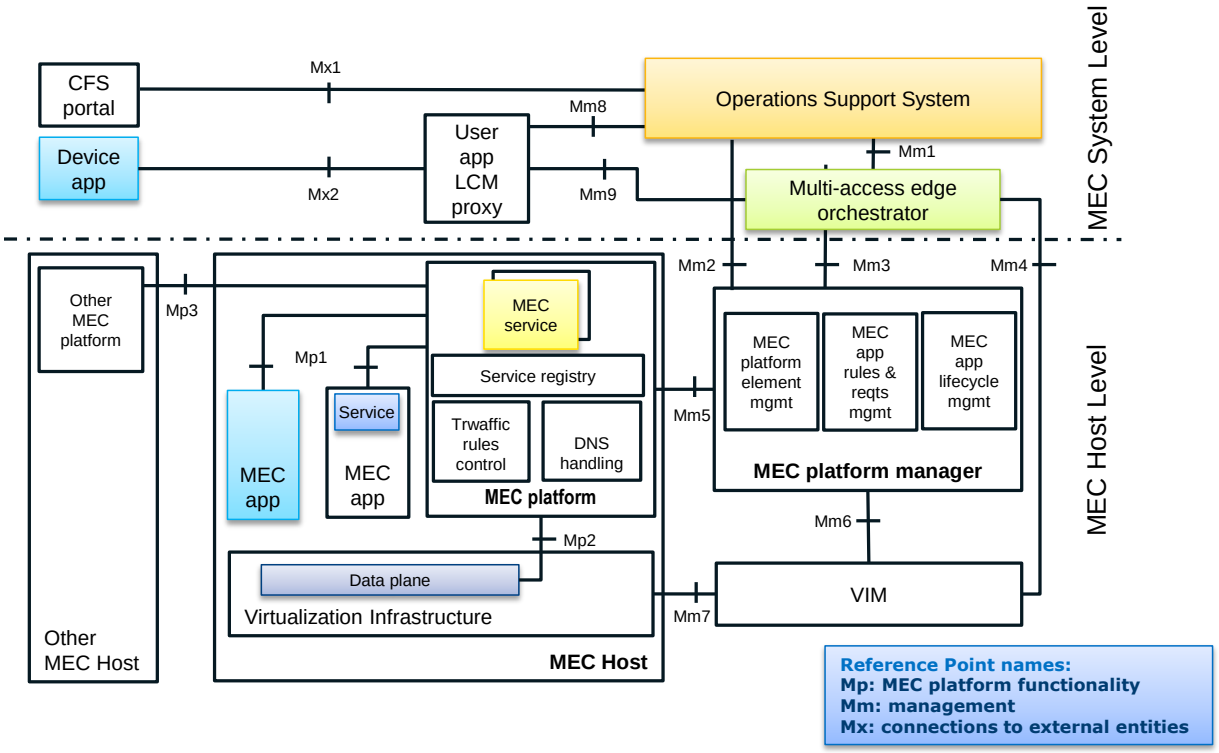

Figure 2 - ETSI MEC Architecture [5]

The MEC system reference architecture is defined in ETSI ISG MEC Group Specification (GS) MEC003 [5] and it is visualized in Figure 2. In particular, a MEC system incorporates two levels: the MEC host level and MEC system Level. The former consists of the MEC host, the MEC platform and the Virtualization Infrastructure Manager (VIM), while, the latter is composed by the MEC Orchestrator, the Operations Support System (OSS) and the user application (or, "User App") Lifecycle Management (LCM) proxy, which acts as entry point for the requests coming from a device application ("Device App") toward the MEC system. The MEC host facilitates MEC applications ("MEC Apps"), offering a virtualization infrastructure that provides computation, storage and network resources, as well as a set of fundamental functionalities ("MEC services") required to execute MEC Apps, known as the MEC platform. The MEC platform is, thus, a key component in the MEC system, and is characterized by the following main functionalities: enabling applications to discover, advertise and consume MEC services, and providing the virtualization infrastructure with a set of rules for the user traffic forwarding plane (such rules are based on the policies associated to the MEC applications). The MEC platform configures also local Domain Name System (DNS) handling, which assists the user traffic in reaching the desired MEC application, and communicates with other peer MEC platforms via the Mp3 interface, which allows peer platform clustering.

ETSI ISG MEC specifies also a set of RESTful APIs [6] as standardized interfaces, which can be used by application developers to access Radio Network Information (RNI API), location information (Location API), and other types of data pre-processed either by the MEC platform, or, by instantiated MEC applications. In particular, thanks to the RNI API, context information from the RAN can be provided to user level applications or other services for network performance and QoS improvements. As a further example, with regards to Vehicular-to-Everything (V2X) use cases, ETSI ISG MEC is introducing a "MEC V2X API" to assist the MEC system in exposing a set of information to applications, which permit developers, car Original Equipment Manufacturers (OEMs) and their suppliers to implement Intelligent Transportation System (ITS) services in an interoperable way, across different access networks, owned and managed by different Mobile Network Operators (MNOs) and vendors.

2.1.2 3GPP support for Edge Computing

When it comes to standardization activities in 3GPP, Working Group (WG) SA1 is the working group to standardize 3GPP service requirements which are regarded as "stage 1" (i.e., high level requirements) and would trigger the related studies in downstream working groups, including SA2 for system architecture, SA5 for network management, SA3 for security aspects, and SA6 for application architecture.

More specifically, with regards to 3GPP support for Edge Computing, various services characterized by extreme Key Performance Indicator (KPI) values needed to support Ultra Reliable, Low Latency Communication (URLLC) have given Edge computing an important role in a 5G system. In particular, in 3GPP Rel-15 TS 22.261, two service requirements set the tone for Edge Computing in support of 5G services:

- Resource efficiency: To meet the various KPIs defined for 5G, the 5G network shall support mechanisms for minimizing user plane resource utilization by including in-network caching and application in a Service Hosting Environment closer to the end user.
- Efficient user plane: Based on operator policy, the 5G network shall be able to maintain user experience (e.g., QoS, QoE) and support routing of data traffic between a UE attached to the network and an application in a Service Hosting Environment for specific services, and modifying the path as needed when the UE moves or application changes location during an active communication.

As a note, a "Hosted Service" is defined as a service containing the operator's own application(s) and/or trusted 3rd party application(s) in the Service Hosting Environment, which can be accessed by the user. A "Service Hosting Environment" is defined as the environment, located inside the 5G network and fully controlled by the operator, where Hosted Services are offered. In other words, the presence of this Service Hosting Environment is a first sign of the requirements for Edge Computing in 5G systems, where SA1 WG defined the assumptions providing the landing requirements that are currently going on in the other 3GPP WGs, for more specific standardization (e.g. SA2, SA5 and SA6).

2.1.2.1 Release 15 and 16

In SA2, the following definition for Edge Computing has been provided in 3GPP TS 23.501 [7] since Rel-15:

> *"A concept that enables operator and 3rd party services to be hosted close to the UE's access point of attachment, to achieve an efficient service delivery through the reduced end-to-end latency and load on the transport network."*

To enable the Edge Computing feature in 5G system architecture, the design of the 5G system architecture requires several key enablers[2]:

1. Applicable to both non-roaming and Local-breakout roaming cases;
2. Flexible placement of User Plane Function (UPF): concurrent access to two (e.g., local and central) data networks;
3. Support of Multi-homed Protocol Data Unit (PDU) Sessions: using either "Uplink Classifier" (UL CL) or multi-homed IPv6 (Branching point).
4. The Session and Service Continuity (SSC) mode 3: a connection through a new PDU Session Anchor point is established before the previous connection is terminated to allow for better service continuity and the IP address (IPv4 or IPv6) is not preserved in this mode when the PDU Session Anchor changes.
5. Support of Local Area Data Network (LADN)
6. Application Function (AF) influence on traffic routing: the AF can make a request to influence traffic routing for a target UE by indicating traffic description (Data Network Name - DNN, Single Network Slice Selection Assistance Information - S-NSSAI, Application IDs, traffic filters, etc.), traffic routing requirement per Data Network Access Identifier (DNAI), etc., or request to subscribe the notification of user plane management events.

According to the definitions introduced by SA2 in TS 23.501 [7], we could also identify a sort of mapping between ETSI MEC entities and 3GPP entities, in order to understand how MEC system can be deployed in a 5G network:

- **User Plane Function**(s) (UPFs) "*handle the user plane path of PDU sessions. […]. A UPF that provides the interface to a Data Network supports the functionality of a PDU session anchor*."
  - According to this definition, the logical UPF in the 3GPP architecture may correspond to some functionalities defined in ETSI for the **MEC Data Plane** (ETSI GS MEC 003);
- **Application Function** (AF) in 3GPP contains the following high-level functionalities: application influence on traffic routing, access network capability exposure, interact with the policy framework for policy control;

[2] However, until Rel-16, the following features are not supported and are out of scope of SA2: Discovery of target application server, Application Function (AF) triggers mechanism for traffic redirection and application context transfer from a Source Application Server (S-AS) to a Target Application Server (T-AS), and AF changes.

- According to this definition, the logical AF in the 3GPP architecture may correspond to some functionalities defined in ETSI for the **MEC Platform** (ETSI GS MEC 003);

- Finally, the 5G system architecture introduces the Data Network (DN) receiving user plane traffic from the UPF. **Local DN** deployments can be the perfect examples of environments hosting **MEC applications**, in contrast to a remote DN (or, a central DN).

In summary, Figure 3 shows an example of possible MEC mapping to the 5G system architecture and the related correspondence of logical entities introduced by the two standard bodies.

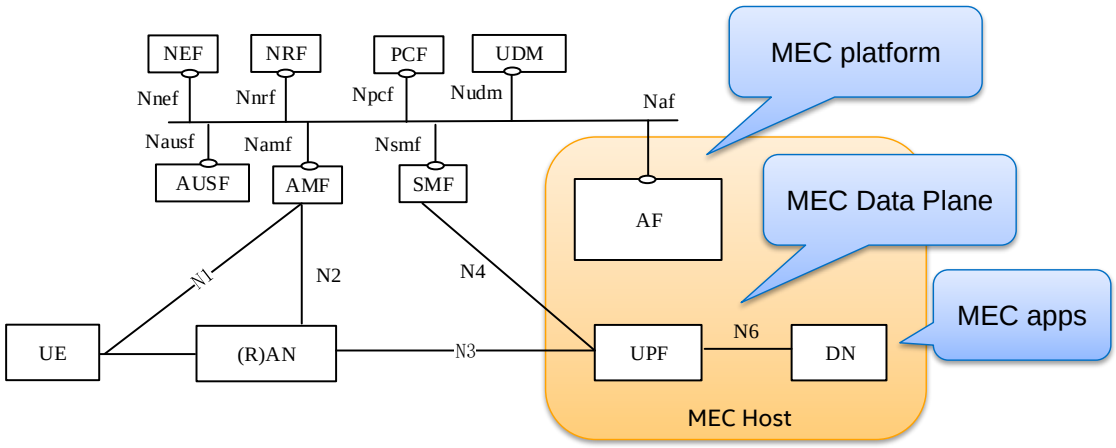

Figure 3 - Example of MEC mapping to the 5G system architecture

A more detailed example, depicted below in Figure 4, could further help to understand how to deploy MEC in 5G systems under the assumption of considering a Network Function Virtualization (NFV)-based network.

Based on the ETSI NFV framework, to consider the coordination between UPF and MEC application deployment/management and to support 5G edge computing, the UPF Virtualized Network Function (VNF) and the Edge Computing VNF (i.e., the MEC App) should be deployed at locations where the end-to-end QoS requirements, including 5G QoS (i.e. on UE – UPF connection) and QoS for the N6 interface, would be met (see Local DN below). When MEC application(s) are available/installed at an Edge Computing VNF, a trusted AF (i.e., the MEC Platform) can request the Session Management Function (SMF) via the Policy Control Function (PCF) to influence UPF (re)selection and traffic routing. All MEC entities can be implemented as VNFs, and run in the same NFV Infrastructure Point-of-Presence (NFVI-PoP), as in the example below, or in different locations.

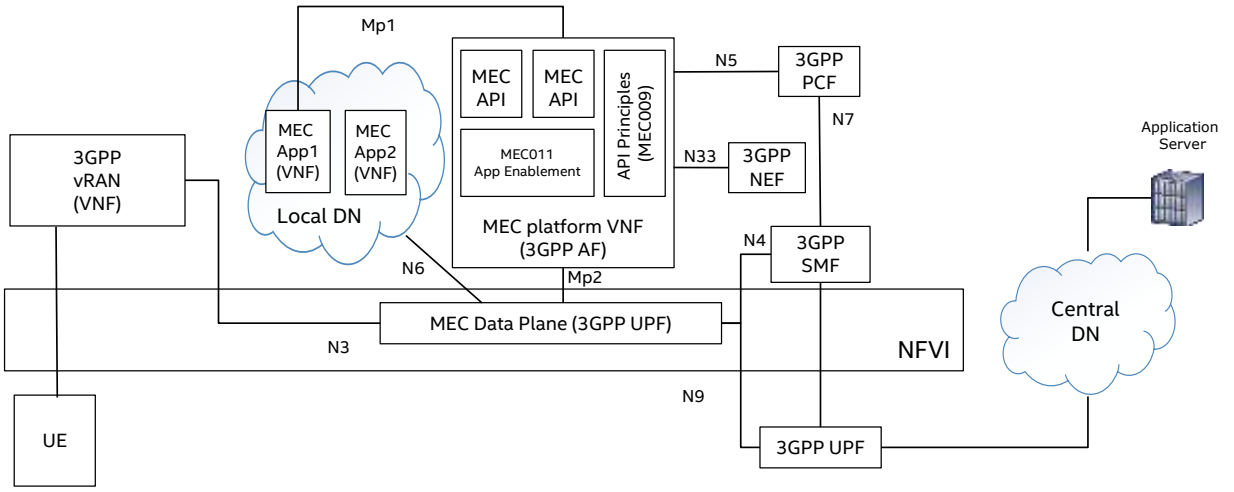

Figure 4 - Example of MEC deployment in an NFV-based 5G network

SA5 is responsible for network deployment and management aspects in 3GPP network (management plane). Currently in this WG, a Rel-16 study item [8] is studying the coordination between 3GPP management system and Non-3GPP management system including the ETSI NFV management & orchestration, and Edge Computing management systems.

2.1.2.2 Release 17

Standardization efforts are currently ongoing for the Rel.17 definition of the specifications. Based on existing solutions defined in Rel-15 and Rel-16, a new study [9] in Rel-17 has been agreed in SA2 with several objectives, including the following:

1. Enable support for seamless changes of the Application Server serving the UE within an Edge Computing Environment (ECE) or between the ECE and the cloud, including discovery of the IP address of the Application Server in the ECE;

2. Provide deployment guidelines for typical Edge Computing use cases including e.g., URLLC, V2X, Augmented Reality (AR)/Virtual Reality (VR)/ Extended Reality (XR), Unmanned Aircraft System (UAS), integration of satellite access in 5G (5GSAT), and Content Delivery Network (CDN).

Moreover, SA6 WG is currently working on a Rel-17 study of the application architecture for enabling edge applications [10]. Due to many concurrent activities in 3GPP, and the existence of the ETSI MEC standard, some coordination within 3GPP and between ETSI and 3GPP is envisaged, in order to guarantee coherence among the various specifications, and thus avoid market fragmentation that could, instead, increase the costs related to the adoption of Edge Computing. Here below are possible areas of coordination, in order to ensure an alignment on the various activities in 3GPP and ETSI:

- SA2 and SA6: coordinate respectively system architecture and application architecture (e.g., definition of different entities, edge network providers, edge service subscription, relationship between AF and Edge Enabler Server, APIs via network exposure functions, e.g., location information of the UE, UE identifier, etc.);
- SA6: Coordinate between MEC-related definition and ETSI MEC (in progress);

- SA5: Coordinate with ETSI MEC / ETSI NFV and support the management of 3GPP Edge Computing with alignment to the outcome of SA2, and SA6 in Rel-17;
- SA3: address the security aspect of mobile edge computing, e.g., in EDGE-1, EDGE-2, EDGE-3, and EDGE-4;
- All groups: Need more inputs on practical deployment use cases in areas of URLLC, V2X, AR/VR/XR, UAS, 5GSAT, and CDN.

2.1.3 Other SDOs: focus on security aspects

Security decisions are, in principle, very complex, in part due to the rich stakeholder ecosystem of Edge Computing. Different stakeholders bring differing expertise, capabilities and resources. A user workload may execute in an environment involving multiple stakeholders, each supplying various aspects of a deployment solution. Security decisions are linked to an understanding of which stakeholders are authorized to supply which expertise, capability and resource. Edge orchestration and Service Level Agreements (SLA) provide important context for security decisions by identifying stakeholders and contracted (expected) support.

The American Council for Technology-Industry Advisory Council (ACT-IAC) published a report[3] that outlines a new approach aimed at cloud and edge security, known as "Zero Trust (ZT)", which has the potential to substantially change and improve an organization's ability to protect their systems and data. ZT is a security concept anchored on the principle that organizations need to proactively control all interactions between people, data, and information systems to reduce security risks to acceptable levels. The ACT-IAC report defines five stages of ZT, the first three of which focus on this concept of identifying and assessing assets – data, users and devices. The latter two focus on continuous security risk management:

1. Establish User Trust
2. Gain Visibility into Devices & Activity
3. Ensure Device Trustworthiness
4. Enforce Adaptive Policies
5. Zero Trust (end goal)

Edge standards will evolve to embrace these security imperatives. Trusted computing technology addresses the device aspects of "Zero Trust" stages and includes capabilities such as hardware "Root of Trust" (RoT) designs that establish attestable cryptographic identities for RoT components. Even simple Internet-of-Things (IoT) platforms may consist of multiple components (e.g. CPU, IO controllers, memory, network interface, sensor/actuator), one or more of which could contain a hardware root of trust. More sophisticated platforms may have various forms of acceleration components (e.g., Graphic Processing Unit (GPU), machine learning processors, Field Programmable Gate Array (FPGA)). Datacenters construct racks containing reconfigurable compute clusters, storage pools and network equipment. Every configurable component could contain a RoT element that helps realize "zero trust".

The IETF security area defines standards for interoperable attestation and Trusted Execution Environments (TEEs). A TEE is a hardened environment for running very security sensitive workloads.

[3] https://www.actiac.org/system/files/ACT-IAC%20Zero%20Trust%20Project%20Report%2004182019.pdf

Often, they are a subset of a traditional workload consisting of the most security-relevant operations such as cryptography, user authentication, biometrics, privacy, access control and rights management. Most TEE technology is built using hardware RoT building blocks.

ZT security requires assessment of the various computing environments that host edge frameworks and services known as 'attestation'. Before an edge orchestration entity schedules workload execution on a container cluster or a mesh of serverless nodes, the security properties of the intended computing resources are assessed to determine if the sensitivity of workload data and processing matches the protection capability of assigned resources.

2.1.3.1 IETF Trusted Computing Standards

The IETF Remote Attestation Procedures (RATS) working group defines interoperable attestation architecture, information models, trustworthiness properties and protocol bindings. Existing and emerging standards aimed at Internet connected Cloud, Edge and IoT systems will be able to support attestation capabilities. Protocols, thus enabled, will become part of a pervasive infrastructure for assessing trust.

The IETF Trusted Execution Environment Provisioning (TEEP) working group defines protocols and formats for interoperable TEE provisioning using Open Trust Protocol (OTrP). OTrP requires support for attestation to ensure provisioning steps are performed between trustworthy endpoints.

The IETF Software Update for Internet of Things (SUIT) working group defines architecture, information model and data structures for the secure update of firmware and software of IoT and similar endpoints. Secure software update depends on update procedures that are protected by a trustworthy environment. In a ZT scenario, the software update server should verify the device receiving the update is the correct device and that the update was applied correctly. Attestation allows the update server to make this assessment even when the device being updated may be using compromised software.

2.1.3.2 DMTF

DMTF (formerly known as the Distributed Management Task Force) creates open manageability standards spanning diverse emerging and traditional IT infrastructures including cloud, virtualization, network, servers and storage. DMTF's Redfish® is a standard designed to deliver simple and secure management for converged, hybrid IT and the Software Defined Data Center (SDDC). Both human readable and machine capable, Redfish® leverages common Internet and web services standards to expose information directly to the modern tool chain.

The Security Protocol and Data Model (SPDM) Specification (DSP0274) provides message exchange, sequence diagrams, message formats, and other relevant semantics for authentication, firmware measurement, and certificate management. This specification for additional security defined by SPDM has a goal of aligning component authentication and integrity objects across the industry and is being designed to be referenced by other standards organizations.

Peripherals and devices implementing SPDM have the ability to report firmware measurements securely using embedded device identities. When combined in a comprehensive network-wide attestation capability, an orchestrator, user, scheduler, regional operations center, gateway, concentrator, cluster manager, compliance auditor or other parties may be able to effectively apply "zero trust" objectives to edge deployments.

2.1.3.3 Trusted Computing Group

The primary focus of the Trusted Computing Group (TCG, https://trustedcomputinggroup.org), as the name implies, is trusted computing. It standardizes RoT mechanisms such as the Trusted Platform Module (TPM) and Device Identity Composition Engine (DICE). The TCG also defines APIs for interacting with trusted computing modules and trusted computing infrastructure.

The TCG Embedded System Workgroup (ESWG) and DICE work group define RoT capabilities and profiles that meet the special needs of constrained environments. Interestingly, the RoT building block technologies are relevant in the datacenter and edge computing environments since all are cost conscious but require strong security.

The Infrastructure Workgroup (IWG) defines trusted computing infrastructure and architecture for managing the lifecycle of a trusted device. Trustworthy device identity often involves the device manufacturer and related supply chain entities that create the RoT, embed attestable identities and certify them in a way that supports automated appraisal.

Trusted computing lifecycle also involves assembling databases of components, firmware and system software that is designed to be secure. Before these components begin to function as a cohesive solution, a security decision is required that determines whether each component is authorized to participate in the solution. Attestation (described more in detail in the following section) is a security capability that addresses this need, attestation verifiers use this data to compare with actual reported values that may indicate the presence of untrustworthy software or an invalid configuration.

2.1.3.4 Attestation and trust in edge environments

Attestation is a security capability that allows stakeholders to evaluate security conditions that determine whether each component of a solution is authorized to participate in that solution. Different stakeholders may request attestation of a device in order to be assured that an agreed upon configuration is indeed in place. Once stakeholder appraisals are satisfied, workload executions can commence.

Security enforcement may need to be utilized by many stakeholders which requires use of enforcement points not under the direct control of a particular stakeholder. Consequently, Edge Computing platforms must have hardened and trusted execution capabilities that are commonly acceptable to the other stakeholders supplying a portion of the solution.

Attestation is a simple conversation (See Figure 5) between edge nodes at a point before exchanging data or performing important functions. It might make sense for an orchestrator to verify the Function-as-a-Service (FaaS) nodes designated for workload execution or for a MEC host to verify the 5G equipment it depends on for connectivity before blindly using it.

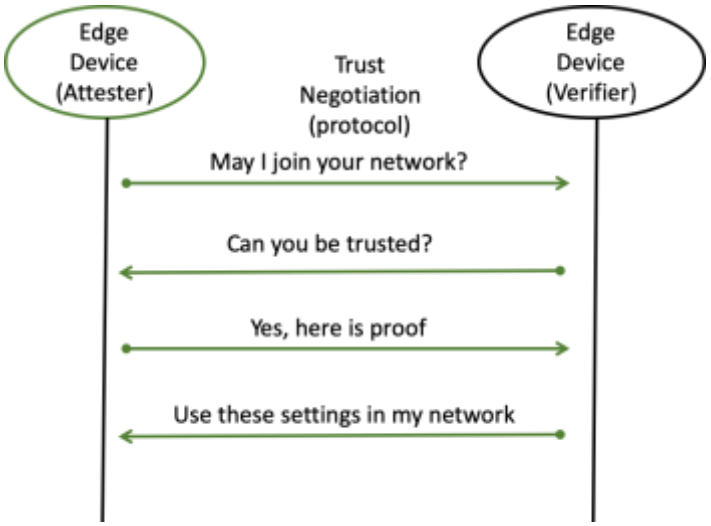

Figure 5 - A simple attestation conversation

Attestation and other security functionality rely on a RoT environment (See Figure 6) that is hardened to resist a variety of software and hardware attacks. It is also a constrained environment because complexity itself is a security concern. Vulnerabilities may lurk undetected simply because it is too difficult to exercise all possible device states and determine if it has undesirable security properties. A RoT environment may be reduced to only the essential functions as a way to improve its trustworthiness. In some cases, there may be multiple RoT environments in the same device that cooperate to supply a more complete set of trusted functionalities. For example, the TPM defined by TCG is a root-of-trust for storage and reporting (attestation), but does not provide a RoT for securely booting the device.

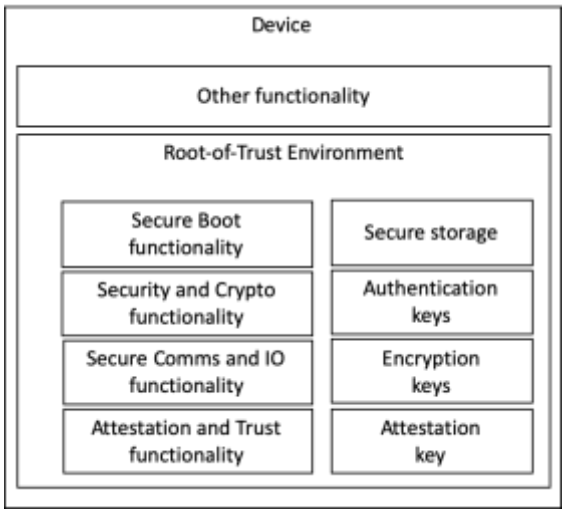

Figure 6 - A RoT environment in an edge device

Attestation reveals the security and trustworthiness attributes of the edge device to a verifier or other relying party so that the security risk can be assessed according to security best practices and good manufacturing processes. Trust in the attestation functionality is another layer of functionality that needs to be hardened. The TCG Device Identity Composition Engine (DICE) is a root-of-trust that anticipates layering of trustworthy software so that complexity of trusted layers can be managed.

2.2 Industry groups and open source initiatives

Industrial associations, as non-standard bodies triggered by vertical market segments, are key players able to influence the ecosystem of Edge Computing adopters. In addition, open source projects and related initiatives offer complementary tools for the acceleration of implementation of edge-based solutions. Some key representatives of these groups of stakeholders are described in the following.

2.2.1 Vertical associations on automotive

The automotive market is one of the key vertical segments driving the introduction of Edge Computing. Figure 7 below depicts a typical automotive scenario, where, multiple vehicles, potentially belonging to different car OEMs and other devices (e.g., smartphones and other Vulnerable Road Users - VRUs) are connected to infrastructure (Road Side Units - RSUs) and a cellular network (RAN). The client application instances are generically able to communicate with server applications, i.e., at edge clouds, remote clouds, and/or OEM/private clouds).

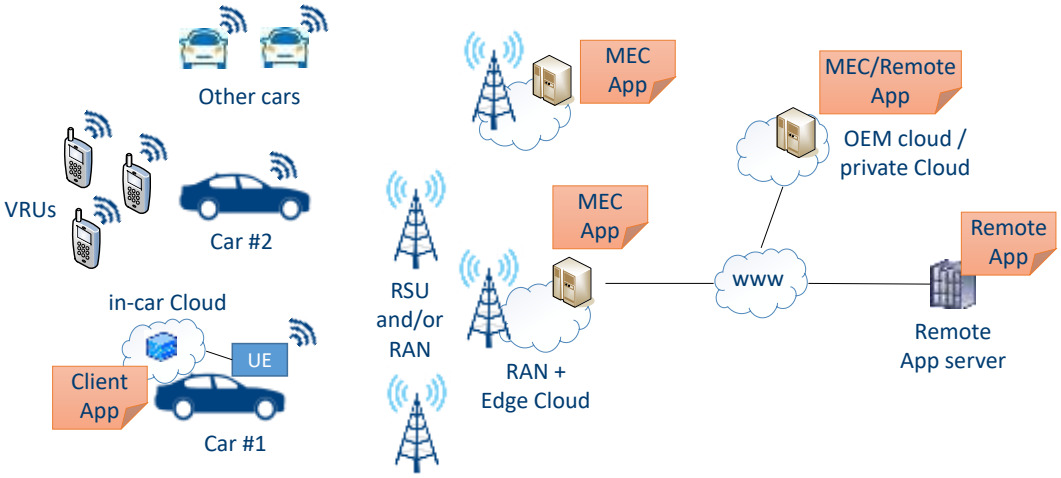

Figure 7 - Edge Computing support for automotive scenarios

Two main industry groups are relevant in this segment: 5GAA (5G Automotive Association, www.5gaa.org) and AECC (Automotive Edge Computing Consortium, https://aecc.org).

2.2.1.1 5GAA

Created in September 2016, 5GAA brings together automotive, technology and telecommunications companies to work closely together to develop end-to-end connectivity solutions for future mobility, smart cities and intelligent transportation services. The automotive industry comes with the vehicle platforms, hardware and software solutions and the telecommunications industry with connectivity and network systems, devices and technologies.

Several Edge Computing related activities have taken place in 5GAA. In December 2017, 5GAA published the white paper[4] "Toward fully connected vehicles: Edge computing for advanced automotive

[4] http://5gaa.org/wp-content/uploads/2017/12/5GAA_T-170219-whitepaper-EdgeComputing_5GAA.pdf

communications" under the leadership of Intel. In February 2018 it organized the 5GAA Open Workshop[5] on "Edge Computing and V2X", where Intel both coordinated the effort and participated to the event. At the Mobile World Congress 2018, 5GAA announced Edge Computing as one of the key supporting technologies for many V2X services for connected vehicles and for autonomous driving. In May 2018 a 5GAA internal paper titled "Towards a Future 5G Vision 5GAA Strategy for Edge Computing" was finalized under the leadership of Intel. Several work items in 5GAA have relationship to Edge Computing and in May 2019 a work item dedicated to Edge Computing was kicked-off, as well. The project has targets to demonstrate the potential and added value of Edge Computing for automotive services in a multi-MNO, multi-vendor and multi-OEM environment, whereby Intel has the overall lead and also sub-tasks responsibilities.

2.2.1.2 AECC

Automobile Edge Computing Consortium (AECC) was formed in August 2018 initially by Toyota, Intel and Ericsson. Currently, the members include companies across different industries and still growing. The mission of AECC is to help automotive manufacturers, OEMs and the supply chain to accommodate growing requirements by evolving current network architectures and computing infrastructures. Work by the Consortium helps industry stakeholders to set the new route for connected cars by increasing network and computing capacity. The shared strategic goal accommodates automotive big data smartly between vehicles and the cloud by using Edge Computing and more efficient system design.

Edge Computing is believed to be a crucial technology to enable service scenarios such as intelligent driving, high-definition maps, V2Cloud cruise assistance, extended services including Mobility as a Service (MaaS), along with finance and insurance. A concept of "distributed computing on localized networks" was introduced in an AECC white paper[6] to address the problem of big data for automobile and optimize the current mobile communication network architectures and cloud computing systems. This concept is characterized by three key aspects: localized network, distributed computing and local data integration platform. The technical details including the key issues and solutions identified for AECC system was published in AECC TR in Sep, 2019[7].

2.2.2 Vertical associations on industrial automation

The industrial automation market is another key vertical segment driving the introduction of Edge Computing. As illustrated in Figure 8, focusing on manufacturing scenarios (also, known as "Factory of the Future" – FoF), Edge Computing is expected to bring significant gains when it comes to performance metrics such as production line efficiency via reliable, data analytics-based process monitoring, as well as process control through the design of controllers and actuation policies based on low-latency sensory information. The proximity of Edge Computing hosts to the system's end points is a decisive feature towards (i) enhancing system ("plant") stability by improving Communication Service Reliability (CSR)

[5] http://5gaa.org/news/c-v2x-edge-computing-the-winning-technologies-for-connected-vehicles-and-autonomous-driving/

[6] https://aecc.org/wp-content/uploads/2018/02/AECC_White_Paper.pdf

[7] https://aecc.org/wp-content/uploads/2019/09/AECC_WG2_TR_v1.0.2.pdf

and (ii) addressing the problem of accumulating and processing an explosive amount of data at a single remote (cloud) server.

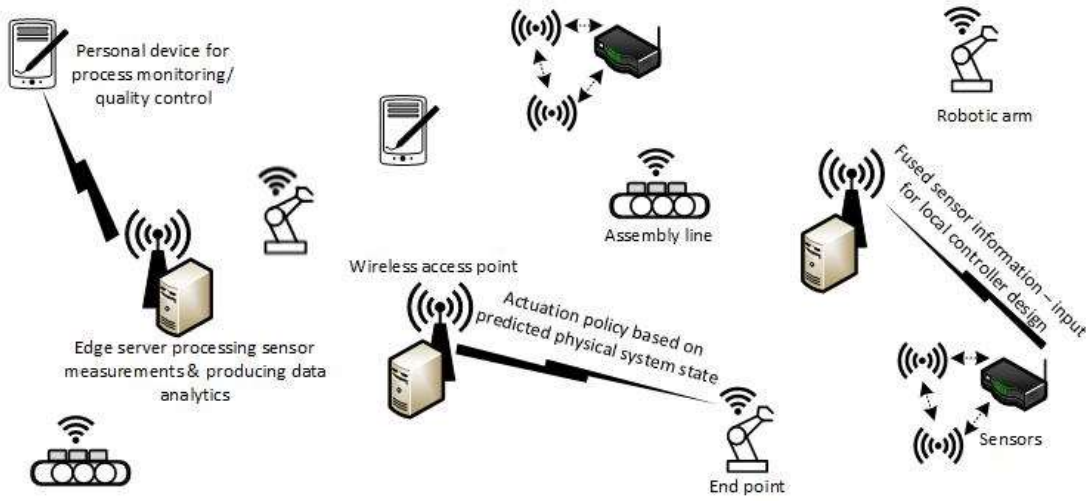

Figure 8 - Edge Computing support for industrial automation scenarios

Two main industry groups are relevant in this segment: 5G-ACIA (5G Alliance for Connected Industries and Automation, https://www.5g-acia.org) and IIC (Industrial Internet Consortium, https://www.iiconsortium.org).

2.2.2.1 5G-ACIA

The performance improvements delivered by 5G communications, as compared to previous generations of wireless communications, are for the benefit of not only the (well established) enhanced Mobile Broadband (eMBB) use cases, but also a newly introduced set of use cases grouped into two additional service types: URLLC, as well as massive Machine Type Communications (mMTC – also termed after as "massive IoT"). Driven by the latest technical developments and regarding the two latter service types, the industrial automation vertical segment has shown increased interest in improving the efficiency of manufacturing processes by replacing complex and costly wired communications infrastructure with 5G wireless links. To turn the goal of "smart factories" into reality, the Information & Communication Technology (ICT) and the Operational Technology (OT) stakeholder groups need to closely collaborate in order for the OT service particularities and performance requirements to be fully understood by the ICT ecosystem, in order for the latter to steer efforts towards fulfilling the needs of the automation industry. To accomplish that, the 5G-ACIA was established as a global forum of discussing and evaluating relevant technical, business and regulatory proposals concentrating on the vertical segment of industrial automation[8].

[8] https://www.5g-acia.org/fileadmin/5GACIA/Publikationen/Whitepaper_5G_for_Connected_Industries_and_Automation/WP_5G_for_Connected_Industries_and_Automation_Download_19.03.19.pdf

2.2.2.2 IIC

The Industrial Internet Consortium (IIC) was founded in March 2014 to bring together the organizations and technologies necessary to accelerate the growth of the industrial internet by identifying, assembling, testing and promoting best practices. More recently, OpenFog Consortium activities and members have been acquired into the IIC organization so that IIC can continue the progress OpenFog has made toward accelerating the adoption of fog computing.

The relevance of Edge Computing to IIC is evident from the recently published IIC architecture [11], which is divided in three tiers: edge tier, platform tier and enterprise tier. The **edge tier** collects data from the edge nodes, using the proximity network. The **platform tier** receives, processes and forwards control commands from the enterprise tier to the edge tier. It consolidates processes and analyzes data flows from the edge tier and other tiers. It provides management functions for devices and assets. It also offers non-domain specific services such as data query and analytics. The **enterprise tier** implements domain-specific applications, decision support systems and provides interfaces to end-users including operation specialists. The enterprise tier receives data flows from the edge and platform tier. It also issues control commands to the platform tier and edge tier.

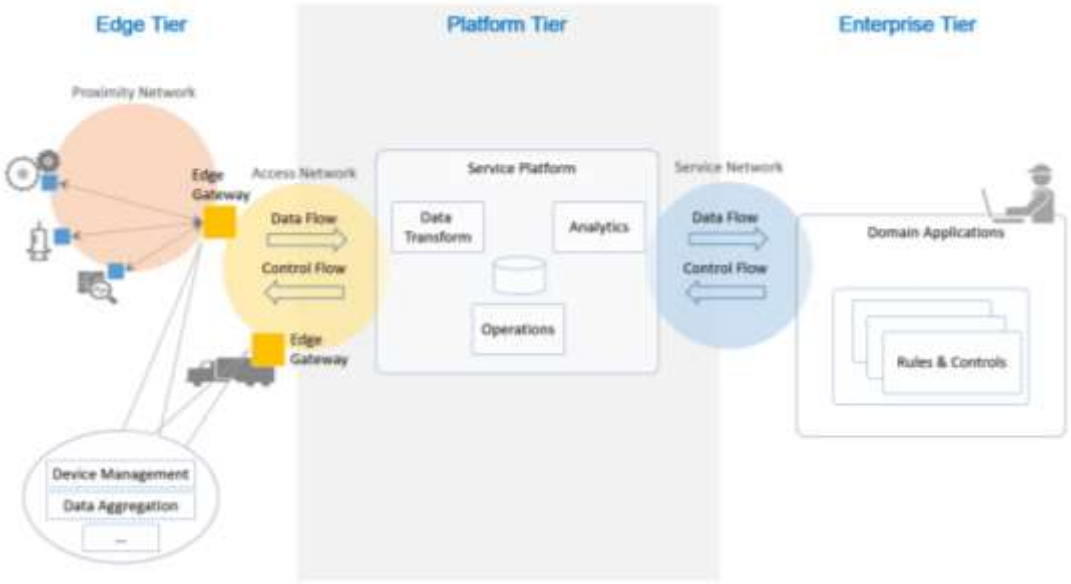

Figure 9 - Three-tier industrial IoT system architecture (Source: IIC [11])

Many benefits are also identified by IIC on Edge Computing:

- Latency: The edge can provide latency in milliseconds, while multiple hops and long transmission distances to the platform tier is in the 50-150 ms range. Latency to centralized data centers and the public cloud is even greater.
- High throughput: The throughput available to the user from the edge, served via cached or locally generated content, can be orders of magnitude greater than from a core data center.
- Data reduction: By running applications (such as Machine Learning (ML)/ Deep Learning (DL) ones) at the edge, operators and application vendors can substantially cut down the amount of

data that has to be sent upstream. This cuts costs and allows for other applications to transfer data.

- Isolation: A number of environments are not always connected to the Internet over high speed links. The edge is able to provide services during periods of degraded or lost connections.
- Compliance: Edge applications can help with privacy or data location laws.
- Accuracy: Combine local results at the edge with global results at the platform to get a more comprehensive view of the data in real-time.

Edge Computing has been implemented in a variety of Industrial IoT (IIoT) deployments; however, the need to modernize edge architectures became apparent with the emergence of cloud computing. The rapid decline in processor and memory cost enables more advanced decision-logic closer to where the data is created, at the edge. The industry has learned that a "one-size-fits-all" approach has never been adequate for IIoT. It is also true that the IIoT system designers always know where the edge boundary is, and what devices in the system can be categorized as edge devices. System designers are challenged to implement an architecture that is managed, orchestrated, trustworthy and secure.

2.2.3 Vertical associations on AR / VR

The ability to leverage cloud computing and Edge Computing can be instrumental in reducing computational cost and complexity for the client devices when it comes to handling the high quality and interactivity performance demands and corresponding processing requirements associated with immersive media, including workloads such as decoding, rendering, graphics, stitching, encoding, transcoding, caching, etc., which may all be performed over the cloud and/or edge. Low latency, high throughput networks such as 5G could allow instantaneous access to remotely stored data, while offering a local computing experience similar to a datacenter-based system. Such a high capacity network with low latency characteristics also enables responsive interactive feedback, real-time cloud-based perception, rendering, and real-time delivery of the display content.

With such an edge-based approach, it is sufficient to use low-cost thin client devices with minimal built-in functions. For VR and AR, these include the display screen, speakers for output, vision positioning and hand-controller sensors for input. The thin client simply uses the network to pass inputs to processing functions at the cloud or edge, and receives the necessary images to display.

Cloud VR/AR/XR brings together the significant advances in cloud computing and networks with a high degree of interactivity to provide high quality experiences to those who were previously priced out of immersive technologies. This approach may, thus, help significantly increase the number of VR client devices sold, and create major revenue opportunities from increased consumption VR/AR services for content providers and operators.

A currently ongoing GSMA Cloud VR/AR initiative [12] aims to address this opportunity from the perspective of mobile operators, who aim at leveraging their infrastructure to monetize VR/AR/XR services, cooperating with content providers. MPEG's Network-Based Media Processing (NBMP) specification ISO/IEC 23090-8 [13], aims to specify metadata formats and APIs for intelligent edge media processing as an enabler to offload compute-intensive media processing to the edge. For instance, such a capability can be relevant to 3GPP's Framework for Live Uplink Streaming (FLUS) service in TS 26.238, in which videos captured by one or multiple omnidirectional cameras (without built-in stitching) may be sent separately to the cloud or edge via an uplink connection, where they may be stitched to create 360°

videos and then encoded and encapsulated for live distribution. Edge enhancements enabled by 5G also help in improving viewport-dependent 360° video delivery, where high quality viewport-specific video data (e.g., tiles) corresponding to portions of the content for different Fields of View (FoVs) at various quality levels may be cached at the edge and delivered to the client device with very low latency based on the user's FoV information, as described in 3GPP TS 26.118 [14], MPEG's ISO/IEC 23090-2 [15] as well as in VR Industry Forum (VRIF) Guidelines [16]. Finally, VRIF is currently working on delivering broader interoperability guidelines and implementation best practices around cloud VR/AR/XR with Edge Computing, also considering the distribution of volumetric video content.

2.2.4 LF edge Akraino

Akraino Edge Stack is a project created in February 2018 by the Linux Foundation, with the aim to create an open source software stack to improve the state of edge cloud infrastructure for carrier, provider, and IoT networks. This project will offer users new levels of flexibility to scale edge cloud services quickly, to maximize the applications or subscribers supported on each server, and to help ensure the reliability of systems that must be up at all times.
This open source software stack intends to provide critical infrastructure to: enable line speed processing, enable high throughput, reduce latency, improve availability, lower operational overhead, provide scalability, address security needs and improve fault management.

The main motivation of Akraino Edge Stack is that there are many open source projects that provide component capabilities required for Edge Computing. However, there is no holistic solution to address the need for fully integrated edge infrastructure. Akraino Edge Stack, a Linux Foundation project initiated by AT&T and Intel, intends to develop a **fully integrated edge infrastructure solution**, and the project is completely focused towards Edge Computing.

- AT&T's seed code will enable carrier-scale Edge Computing applications to run in virtual machines and containers. AT&T's contributions, which will include support for 5G, IoT, and other networking edge services will enhance reliability and enable high performance.
- Intel upstreamed its Wind River Titanium Cloud portfolio of technologies to open source in support of additional blueprints in Akraino. Intel is also a premium member of LF Edge and Akraino community, and is actively participating in various Akraino Blueprints.

Akraino Edge Stack community is focused on edge APIs, middleware, Software Development Kits (SDKs) and will allow for cross-platform interoperability with 3rd party clouds. The Edge Stack will also enable the development of edge applications and create an application with the VNF ecosystem. The Akraino Edge Stack is intended to support any type of access methodologies such as Wireless (4G/LTE, 5G), wireline, Wi-Fi, etc.

Akraino is also a complementary open source project, and interfaces with existing projects namely Acumos AI, Airship, Ceph, DANOS, EdgeX Foundry, Kubernetes, LF Networking, ONAP, OpenStack, and StarlingX. The Akraino Edge Stack is a collection of multiple blueprints, which are defined as the declarative configuration of entire stack i.e., Cloud platform, API, and Applications. The intent of Akraino Edge Stack is to support VM, container and bare metal workloads. Section 4 contains an example of Akraino blueprint, customized for the automotive case study.

2.2.5 OpenNESS

Open Network Edge Services Software (OpenNESS) is an open source software toolkit that enables the deployment of edge services on diverse platform and access technologies. It is foundational software, because, "out of the box", it supports the deployment of edge services in a network, but can also be extended in functionality into a commercial platform, or re-used to add capability to existing edge platforms. The primary goal of OpenNESS is to reduce the "deployment impedance" experienced by network operators, Independent Hardware Vendors (IHVs), and Independent Software Vendors (ISVs) in deploying edge services. OpenNESS provides a variety of features to achieve this goal:

- it exposes platform and hardware diversity to edge applications and orchestrators;
- it addresses the requirements of both On-Premise deployments (with access latency in the 1-10 ms range) and Network Edge deployments (with access latency in the 10-40 ms range);
- it supports the major access network technologies in use in edge networks, including wireline, WiFi, and LTE and 5G mobile networks;
- it supports the extension of public cloud services into the edge network;
- it supports a variety of Artificial Intelligence (AI) and media computing application frameworks, allowing edge services to take advantage of leading technologies in these areas;
- it incorporates a controller that provides for secure on-boarding of service instances, with APIs that support integration with orchestrators;
- it provides infrastructure to allow applications to advertise themselves as services, and for applications to subscribe to those services;
- it uses, and integrates with, industry frameworks such as Kubernetes, Openstack, and ONAP;
- and it is implemented in a microservices style to support extension and software re-use.

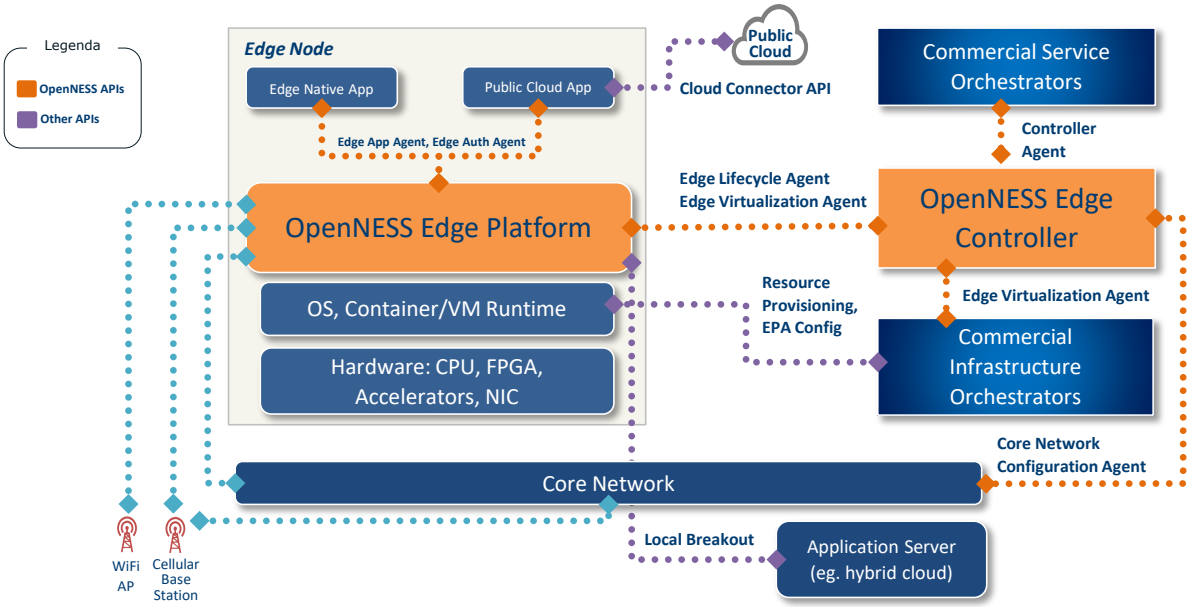

Figure 10 - Architectural diagram of OpenNESS

Figure 10 shows an architectural diagram of OpenNESS, with all the main components and interfaces with access network, core network, public cloud and other infrastructure orchestration entities. The

OpenNESS software framework is available as open source at www.openness.org. An example of commercial offer of MEC platform is represented by Smart Edge (https://smart-edge.com/), which is based on Intel architecture and designed to support enterprise MEC solutions. Smart Edge enables the computation of traffic and services on edge of the network, closer to the customer.

2.3 Functional mapping of edge computing activities

The MEC architecture, as defined by ETSI ISG MEC, does not specify at the same level of detail all the components within the framework. In particular, the implementation level ("stage 3") is only defined for some components really needed to ensure interoperability among different stakeholders, while other parts may need to be proprietary and not fully specified by the standard (thus, just at functional level). Further, some components are being defined by other organizations like SDOs, industry groups, or open source projects and communities (key examples of industry groups are given by those associations focusing on specific verticals like automotive (5GAA), or AR/VR as described in Section 2.2).

Figure 11 shows the different functional entities in the MEC architecture, and their respective SDO/project where they are specified. The MEC platform is a key component in the MEC system that enables applications to discover, advertise and consume MEC services, and provide the virtualization infrastructure with a set of rules for the user traffic forwarding plane as described in Section 2.1.1. OpenNESS is an open source reference platform for MEC described in Section 2.2.5. 3GPP defines the interfaces needed to implement MEC within the cellular edge and core network as described in Section 2.1.2.

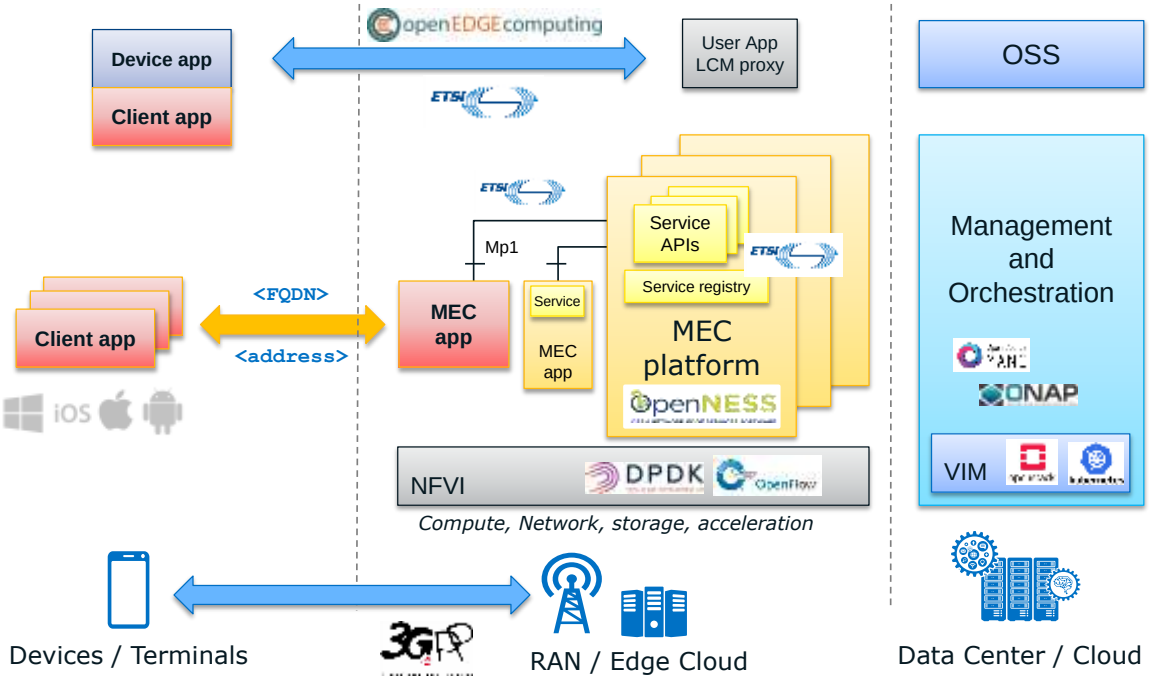

Figure 11 - Functional mapping of Edge Computing activities

As we can see in the above Figure 11, there are multiple solutions for the Management and Orchestration (MANO) part, at the right-hand side of this figure, and these components need also to be better discussed, especially from an Edge Computing perspective. The next section analyzes in more detail the main aspects related to edge orchestration, starting from some definitions, describing the various frameworks available, and finally elaborating on the specificities related to Edge Computing.

2.3.1 Management and orchestration

2.3.1.1 Orchestration definitions and taxonomy

Orchestration: The way to automatically arrange, coordinate and manage complex hardware and software services and resources. Different types of orchestration are possible:

Service orchestration (life cycle management): Orchestration of software and hardware services delivered across one or more deployment infrastructures (see Figure 12 below). Note: the ETSI NFV standard for MANO does not currently address service orchestration.

Infrastructure/resource orchestration (life cycle management): Orchestration of deployment infrastructures, composed of physical or virtual compute, storage and network resources, on which one or more services can run.

Virtual Infrastructure Manager (VIM): This term is interchangeable with **Resource Orchestrator**.

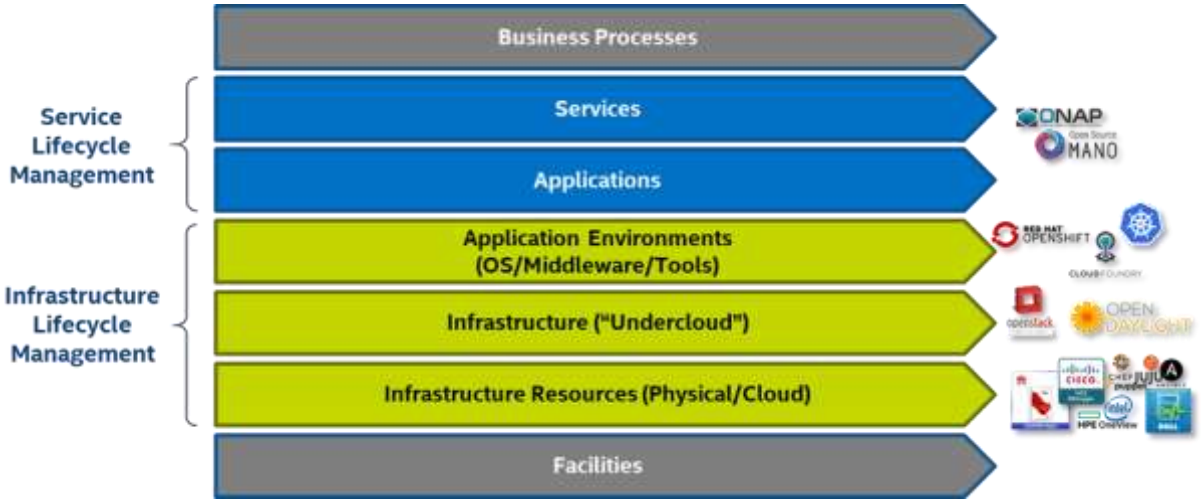

Figure 12 - Different levels of orchestration (lifecycle management)

2.3.1.2 OpenStack

OpenStack is a software platform for cloud computing, mostly deployed as Infrastructure-as-a-Service (IaaS), orchestrating Virtual Machines (VMs) and other resources to be made available to applications (VNFs) running on the infrastructure. The software platform consists of interrelated components that control diverse, multi-vendor hardware pools of processing, storage, and networking resources

throughout a data center. A few vendors (Red Hat particularly) offer their own branded OpenStack distributions. OpenStack is the most widely deployed resource orchestrator

2.3.1.3 Kubernetes

Kubernetes (commonly stylized as **k8s**) is an open-source container-orchestration system for automating application deployment, scaling, and management. Kubernetes, therefore, spans the application orchestration and resource orchestration layers. Many cloud services offer a Kubernetes-based platform or IaaS (Platform-as-a-Service (PaaS) or IaaS) on which Kubernetes can be deployed as a platform-providing service. Many vendors also provide their own branded Kubernetes distributions.

2.3.1.4 ONAP

The **Open Network Automation Platform (ONAP)** is an open source initiative created by combining AT&T's Enhanced Control, Orchestration, Management & Policy (ECOMP) and the Linux Foundation's Open Orchestrator (Open-O) projects for orchestrating the lifecycle of services and the component parts that constitute them.

2.3.1.5 Open Source MANO (OSM)

Open Source MANO is an ETSI-hosted project to develop an orchestration stack that includes service orchestration and that is closely aligned with the ETSI NFV standards. OSM was started by Telefonica and is based on Telefonica's OpenMANO initiative. OSM represents an alternative approach to ONAP for service orchestration and is generally considered smaller and simpler than ONAP, but it is not clear how broad its influence is on the industry as a whole.

2.3.1.6 Edge orchestration aspects

Edge Computing imposes a few unique challenges to orchestration. To name a few:

- **Mobility**: (1) Edge clouds may be small enough to reside on moving platforms, such as cars, ships or airplanes. Characteristics of the communication links to the edge, such as throughput and latency may be changing dynamically. (2) Additionally, services need to operate seamlessly as the mobile edge traverses between adjacent service providers.
- **Resource constraints**: resources, such as power, CPU, storage may be severely constrained at the edge.
- **Scale**: the number of edge clouds may be large.
- **Autonomy**: communication from the network core to the edge may be lost, necessitating autonomous operation for extended periods of time.

A few commercial edge orchestration solutions are available, primarily targeting the enterprise market. Typical architectures deploy a full orchestration stack in the core of the network, with lightweight, semi-autonomous orchestration functionality in each edge location. For Communication Service Providers (CoSPs), challenges are compounded by the requirement ('Mobility' (2) above) for interoperability between operators. Finally, Edge Computing is an opportunity for operators, but risks to become a strategic disadvantage especially to tier-1 CoSPs (since their leadership in cloud technology may not extend to edge orchestration).

Work towards edge orchestration solutions for CoSPs is ongoing in all relevant open source communities and in ETSI:

- The leading orchestration platforms (OpenStack, Kubernetes, ONAP) are currently focusing on solving the scale challenge for the orchestration elements in the core of the network. There is currently no real activity towards lightweight edge orchestrators.
- Akraino's scope includes a 'thin local' control plane.
- StarlingX is another edge computing stack. Focus is on edge infrastructure orchestration. For resource and service orchestration, StarlingX intends to use OpenStack and Kubernetes.

3 Edge Computing deployment aspects

A key aspect for the successful market adoption of Edge Computing is represented by the infrastructure owners' engagement. These important stakeholders are described in the following, together with the analysis of different deployment options offered by Edge Computing technology, and the related performance tradeoffs. Future systems, including **distributed computing** and **information-centric networking**, will be also described at the end of this section, as they can represent a natural evolution path of edge computing deployments.

3.1 Decision makers for the edge

Kicking off the Edge Computing market is a complex process, which needs to involve necessarily all stakeholders, from infrastructure owners to application developers. In fact, for a real market adoption all of them should have a role and should actively engage, enabling the deployment of MEC servers and also the creation of added-value applications exploiting edge services. In particular, if on one hand, the infrastructure owners (at different levels, e.g., at telco edge, enterprise, cloud providers) are key to the instantiation of MEC platforms and proper low-latency environments hosting MEC applications, on the other hand, it is also equally important that a wide ecosystem of application developers (hence, not only operators and service providers) will engage and help to expand and enrich the set of innovative services enabled by Edge Computing technologies.

While Section 4 is mainly targeted to software developers and their needs, the present Section 3 is specifically conceived to deployment issues, which are of highest interest from an infrastructure owner point of view. In fact, in order to be convinced and adopt Edge Computing, the owner of an edge hosting environment needs first to better understand many aspects, from benefits and constraints, to gains and cost tradeoffs, related to all deployment options offered by the edge.

3.2 Deployment options for MEC

A suitable mean to conduct this comparative analysis is end-to-end (E2E) performance evaluation through proper dimensioning and system-level simulations, offering also the advantage to understand the system behavior in proper "what-if" scenarios of interest, in the view of a real deployment.

The E2E performance evaluation for different MEC deployment options is not a trivial task. The diversity of available mapping options from logical system architecture to physical servers is further complicated by varying MEC service consumption options. The standard "trial-and-error" approach often leads to extreme long planning and deployment process. Thus, Intel is conducting this analysis by using a highly configurable E2E simulation framework to facilitate efficient MEC deployment planning and performance evaluations prior to system provisioning. This comprehensive framework, including both radio and edge/core network components, also exploits the Intel CoFluent System Studio™ Technology, where multiple simulation models are integrated to provide necessary inputs. MEC servers at different deployment locations are simulated using the CoFluent model directly. The whole software execution flow is replayed according to different scenarios. The CoFluent model provides the flexibility to configure the flows easily. As shown in Figure 13, the MEC servers can be deployed at different levels in the

CoFluent behavioral model. Typical examples for MEC deployment options include: universal Customer Premises Equipment (uCPE), RAN-edge, Smart Central Office (CO) and Edge Data Centre (DC).

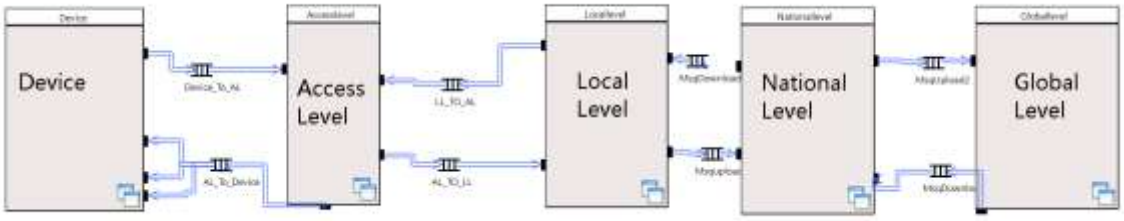

Figure 13 - Example of system modeling with different MEC deployment levels

To simulate meaningful KPIs (e.g., E2E latency and throughput), the CoFluent model leverages statistics from the MEC application model, MEC services model and execution of LTE/5G network (the radio and core network simulator) to simulate computing and communication cost with high accuracy and high speed. The interface between the CoFluent model and LTE/5G network simulation follows the packet flow for both uplink and downlink. Simulation results can be produced by comparing deployment options (e.g. different levels of the edge), in order to assess different MEC platform and MEC App workload models at various network load situations, for specific use cases of interest.

3.2.1 MEC deployment tradeoffs

When it comes to deploying MEC servers, and also instantiating MEC Apps, multiple tradeoffs can be identified, for an effective decision on physical and cloud resource usage. In particular, the processing power demands of customer devices, namely AR/VR, drones, and autonomous vehicles can be very heterogeneous, and, depending on the use case, they can require very low latency, typically measured in milliseconds. The place where processing takes place plays also a major role with respect to quality of user experience and Total Cost of Ownership (TCO). If, on one hand, the centralized cloud decreases the TCO, but fails to address the low latency requirement, placement at customer premises is nearly impossible with respect to cost and infrastructure. Considering the cost, low latency, and high processing power requirements, the best available option is to utilize the existing infrastructures (e.g. Telco's tower, central offices, and other Telco real estates): these will be the optimal zones for the edge placement, at least in a first place. Figure 14 depicts an example of the various performance tradeoffs related to the different edge deployment options.

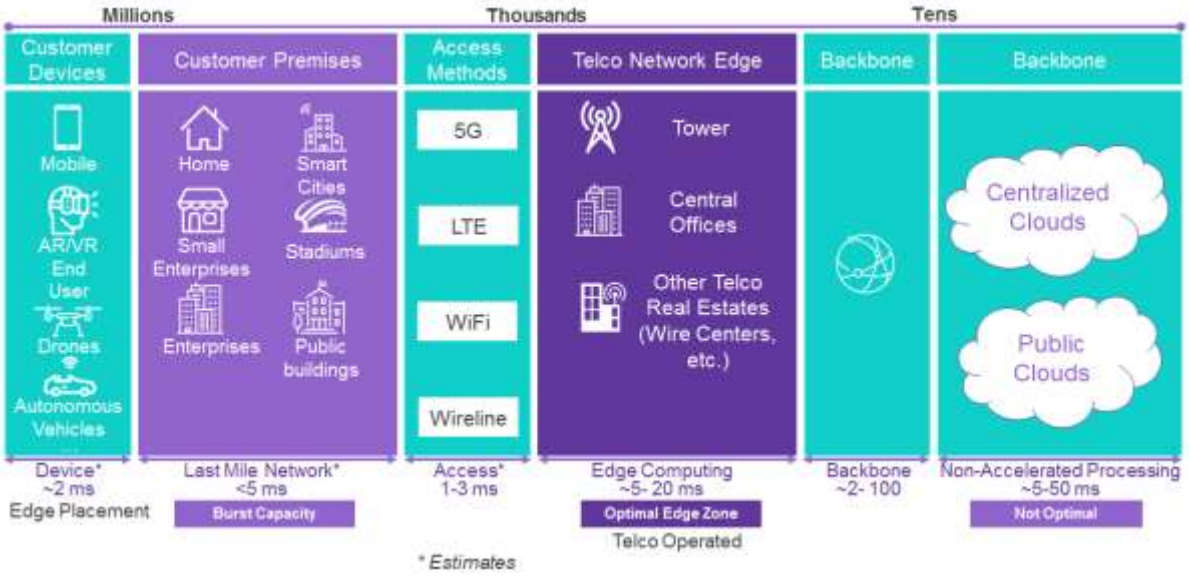

Figure 14 - Example of performance tradeoffs for various edge deployment options (source: Akraino website).

3.2.2 Case study: How MEC service consumption influences the E2E communication latency

Several options are available to decision makers when it comes to MEC deployments, and essentially they depend on the actual location of instantiating MEC applications, which are server-side endpoints of the communication between UEs and the edge. These locations range from uCPEs (mainly at the customer premises), to RAN-edge (thus, with MEC servers co-located with base stations), or to Smart COs, where the platform could be e.g., hosted as co-located with Cloud-RAN (CRAN) aggregation points, or again, to edge DCs at local/regional level, but also potentially also to remote DCs (even if, of course, this would be not a "real" edge deployment).

On the other hand, a specific MEC deployment option can be convenient or not, depending on the specific requirements, thus, a meaningful evaluation of these options should necessarily be tailored to a specific use case of interest. In this perspective, having the MEC system based on a virtualized infrastructure provides the required flexibility to adapt the MEC deployment and topology based on varying needs and convenient performance-cost tradeoffs.

In addition to that, MEC applications are often consumers of edge services (e.g., through MEC APIs), consequently, the actual performance tradeoffs depend also on the consumption of these services and the actual instance locations of both MEC Apps and a MEC platform (which can be co-located or even remotely accessible by MEC Apps).

As a first case study, we have evaluated in a simplified scenario the E2E latency (Round Trip Time – RTT) considering three different workloads (“low”, “middle” and “high”) of an exemplary CDN service, relevant to, e.g. In-Vehicle Entertainment (IVE) use cases in automotive environments.

We have also compared two MEC deployment options (depicted in the left-side of Figure 15):

- **MEC at the RAN edge** (consisting in co-location of edge hosts with Radio Access Points (RAPs)/ Base Stations (BSs)), and
- **MEC at a Smart CO**, e.g., at CRAN aggregation point, where multiple Radio Remote Heads (RRHs) are connected to a Baseband Unit (BBU) pool via e.g., fiber optical fronthaul links,

where these options are also combined with three different cases of MEC service consumption:

1. MEC service consumption at the same locality (i.e., the same MEC host),
2. MEC service consumption at different localities, where the MEC App is consuming a service running in the MEC platform ("remote app-to-service"), or
3. MEC service consumption again at different localities, however, the difference being in that the consumer MEC App is in need of a service-producing MEC App ("remote app-to-app").

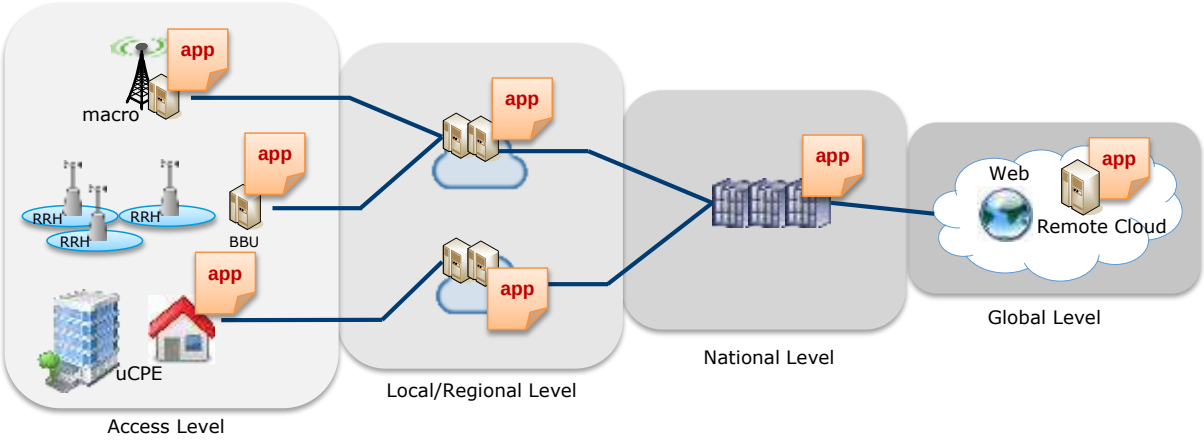

Figure 15 - MEC infrastructure deployment options

In this simple case study, we evaluated as initial KPI the RTT performance for each of the two focused MEC system deployments (illustrated in Figure 16), by taking into account all three workloads and all the above mentioned cases of MEC App service consumption. It is observed that, (i) lower E2E delay is caused when the MEC App is consuming local MEC Service APIs, (ii) higher E2E delay is caused when the MEC App is consuming remote MEC Service APIs (i.e., *not* running at the same MEC host as the MEC App), (iii) the considered MEC App computation load highly impacts the RTT, while, (iv) low gain is observed when moving from "smart" CO to RAN edge deployments. The last observation is coherent to a consolidated trend among the operators, initially oriented to early MEC deployments at the Smart CO (CRAN aggregation point), which is often a first convenient location of deploying MEC servers (of course, some other operators are instead preferring to consider edge RAN deployments, since better performing and more suitable for more challenging use cases).

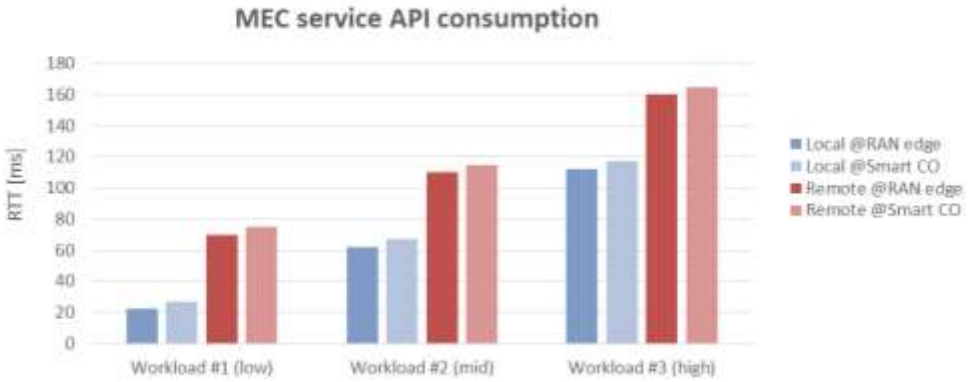

Figure 16 - Impact of different MEC system deployment options, MEC service consumption variants and workload sizes on two-way delay (RTT).

3.3 Evolution path of Edge Computing deployments

Edge Computing brings cloud computing closer to the users at the infrastructure edge. However, in the future, Edge Computing is likely to creep deeper into the edge and client devices in a more distributed manner. These approaches aim to leverage the increasing processing and storage capabilities at client devices towards reducing latency beyond what current Edge Computing can provide.

3.3.1 Distributed computing systems

Edge Computing can span a variety of network locations, form factors, and functions, as depicted in Figure 1 at the beginning of this white paper. Further into the network and cloud is more centralized computing, with applications addressing large numbers of users, and edge platforms hosting multiple applications simultaneously. Distributed computing is closer to end users with applications being more attuned to specific endpoints and functions. This “distributed edge” is characterized by spatial and temporal proximity to clients, real-time responsiveness, interactivity, and mobility, and is well suited for use cases such as industrial control, video analytics, together with interactive (XR) media and healthcare.

Figure 17 shows an oversimplified Venn diagram with some of the common and different attributes of edge processing in different locations. It should be noted that this is not meant to be exhaustive or prescriptive but rather illustrative.

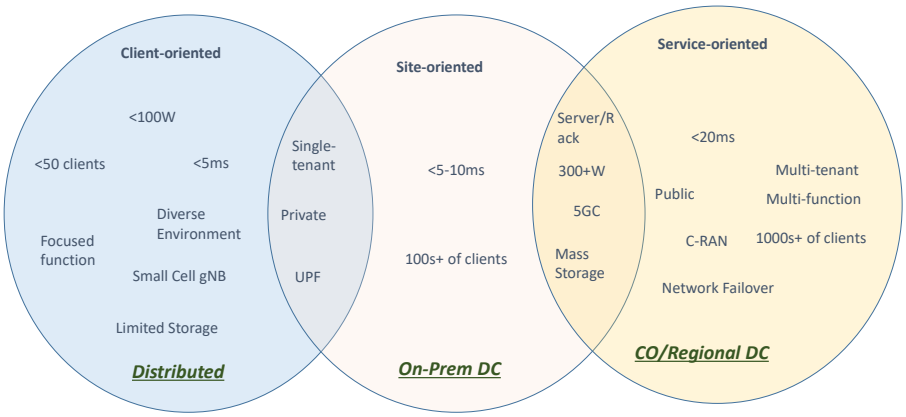

Figure 17 - Edge deployment variations

There are a number of possible ways to deploy and interconnect distributed edge processing nodes, such as the topology scenarios depicted in the following diagram, assuming a 5G System (5GS) architecture.

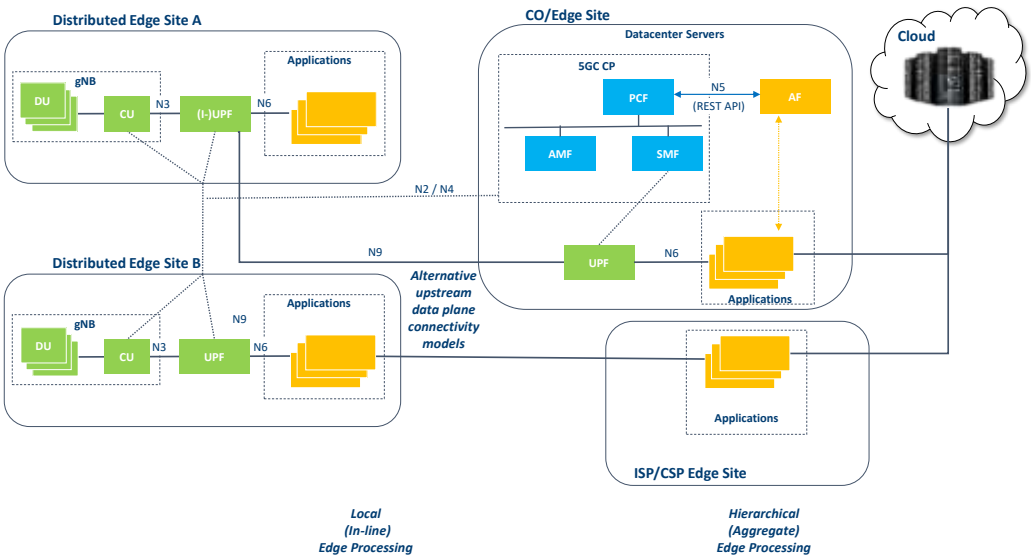

Figure 18 - Distributed edge deployment scenarios

A distributed Edge Computing site may stand alone and be self-contained with no upstream control or data connection, providing wireless connectivity and local processing, for example, in a small-scale industrial control system. However, in most cases it is necessary to connect to upstream processing for common control and aggregation across multiple access points. Two scenarios are shown in Figure 18; in the first (A) case, the distributed site provides applications running locally with access to data behind an *intermediate UPF*, while, also feeding into an *upstream aggregation UPF* in a common 3GPP 5G core network. This setup allows the common control plane to steer data flows toward either local or

upstream compute resources in a flexible manner by configuring appropriate forwarding rules in the intermediate UPF. The behavior of the 5G network connecting the distributed sites can be influenced by AFs interacting with the 5G core network through well-defined interfaces. In the second (B) case, the distributed site provides applications running locally with access to data behind a *local UPF* which terminates the 3GPP flow. The local distributed applications can connect upstream as needed via normal IP networking. In fact, this is not precluded in the first (A) case either – once data is broken out locally, it can be handled and distributed as needed. This allows data partitioning according to site-specific performance and security requirements.

The drivers behind this distributed edge architecture are to provide targeted, unconstrained low-overhead processing very close to clients for high performance, responsiveness, and reliability without complexity. Leveraging the 3GPP management/control structure and user plane with QoS, even in a private networking domain, provides a proven consistent framework for secure reliable wireless networking with mobility management.

Some key aspects of this approach include:

- flexible local native 5G network deployments: small footprint, highly distributable;
- local aggregation points with in-line processing (e.g., media/AI) close to clients;
- fundamentally optimized for high reliability and low latency;
- part of a hierarchical network solution where desired.

There is also a need for distributed access control software for workload balancing so as to simplify provisioning and reconfiguration across the local edge nodes. This is complementary to service-provider network distributed service offerings, providing an alternative client-centric perspective:

- workloads specific to clients and applications installed on them, not pre-defined;
- workload images and lifecycles are dynamic, depend on use case / app running at the client;
- each workload has specific performance and QoS requirements.

3.3.2 Information-centric networking

Internet Protocol (IP) is based on host centric connectivity. The first step to obtaining a piece of information is an IP address of the host. Next, a secure session needs to be established with the host after which the host starts to send the data packets. The network layer's only function through this process is to route the packets from the source host to destination client. Instead, in Information-Centric networking (ICN), you can ask the network for that content directly by its name. This could be over any connection the device has to another device or network (wired or wireless). The network then responds with that piece of information; in a sense the network is a database. A node sends an "interest packet" to the network, where the packet contains the name of the data it wants [18]. There is no reference to a physical location or an address. When a node receives the interest packet for named data, it first checks the contents of its own cache. If the data is not present there, the query gets forwarded based on a local routing table, and this process continues until the query arrives at a node that has the data. As the data traverses back to the requesting node, any intermediate node can cache the data, and then serve it if requested by another node. It is possible for any node to cache the data and serve it back because security is now contained in the data. Currently with IP, security is based on connecting to a trusted host. Since there are no end to end connections with ICN, the security is embedded with the

data itself. This structure enables the user to always get the data from the closest location, improving network efficiency and scaling.

ICN can also be used to orchestrate computation. Instead of sending an interest packet for a piece of information, the user (device) can request the execution of a function. The network then routes the information to the closest resource that can compute the desired functions and returns the processed data back. This is a powerful paradigm where the entire edge can be a computation server. If the user is in a new environment, without any knowledge of the closest edge server, the device can still request the network to orchestrate the computation. Named Function Networking (NFN), Named Function as a Service (NFaaS) and Remote Method Invocation over ICN (RICE) [17] are examples of implementing dynamic and distributed computation within the network. In NFN [19], the network's role is to resolve names to computations by reducing λ-expressions. NFaaS builds on very lightweight VMs and allows for dynamic execution of custom code. RICE presents a unified approach to remote function invocation in ICN that exploits the attractive ICN properties of name-based routing, receiver-driven flow and congestion control, flow balance, and object-oriented security.

4 Edge applications and ecosystem engagement: an automotive case study

Edge Computing represents cloud-computing capabilities and an IT service environment at the edge of the mobile network. It brings virtualized applications much closer to mobile users ensuring network flexibility, economy and scalability for improved user experience. It facilitates a service environment that allows seamless access experience and responsiveness for content, services, and applications. In this section we will explore platform capability needed to support a thriving application ecosystem using "Real-time Situation Awareness and High-definition Maps" as the example edge application.

4.1 Real-time situation awareness and high-definition maps

An automated, semi-automated or manual-driven vehicle is moving on a road (route), heading towards a specific road segment, which presents unsafe and unknown conditions ahead. This host vehicle is made aware in a timely manner of situations detected and shared by remote vehicles and/or road infrastructure nodes such as Road Side Units (RSUs). Situations may include such things as accidents, adverse weather, road conditions, traffic, and construction, among others. The shared situations are relevant along the host vehicle's navigation route or current road of travel.

The situation is shared via data already embedded or to be embedded in a high-definition map. A centralized Edge Computing node collects the situations detected and by remote vehicles and/or road infrastructure nodes such as RSU. The Edge Computing node then consolidates the information collected and intelligently distributes/reports the relevant information to the corresponding road users. This Edge Computing node may be part of or connected to a traffic management authority.

4.1.1 High-definition map

An HD map is necessary as a reliable off-board sensor containing carefully processed a-priori information to "detect" features that are not easily detectable by on-board sensors or to provide a redundant source of information for on-board sensors, including location-based operational design domain determination, environment modeling in adverse conditions and precise semantic understandings in complex driving situations. In situations where on-board sensors cannot reliably detect features, the HD map can be utilized as a more reliable redundant source of information.

The HD map consolidates static and dynamic information (e.g., vehicle position, pedestrians and obstacles) and is mandated for autonomous driving. Creating and distributing the map require many data transactions with high capacity as well as efficient processing to keep the information up to date.

This HD map must be able to accurately localize dynamic objects including vehicles, which is required for automated driving beyond the traditional route guidance. A large amount of data transfer is especially required to update the map. Data is collected from on-board cameras, radar sensors, and laser scanners (LIDAR), transferred and processed in the cloud or in the edge. Typically, what might get sent to the cloud are deviations (Map says X, but Camera says Y). These deviations are sent to the cloud or to the edge to update the HD map.

The completed map information is stored in the center server or the edge server and needs to be distributed to relevant vehicles in a timely manner.

4.2 Example of automotive blueprint for Edge Computing

Blueprints are defined in Akraino as the declarative configuration of entire stack i.e., Cloud platform, API, and Applications. Intend of Akraino Edge Stack is to support VM, container and bare metal workloads. Section 4.2.1 below contains an example of blueprint, customized for the automotive case study, based on the current developments in the Akraino project.

4.2.1 Connected Vehicle Blueprint

The Connected Vehicle Blueprint project in Akraino[9] focuses on establishing an open source MEC platform, which is the backbone for V2X Application. As per Akraino criteria and requirements, blueprint code that will be developed and used with Akraino repository are using only open source software components either from upstream or Akraino projects.

The goal is to establish a MEC platform for connected vehicle use cases, since an edge platform for deploying connected vehicle application does not exist in Akraino so far. In the table below, the main parameters and characteristics of this blueprint are listed:

Case Attributes	Description
Blueprint Name	Connected Vehicle Blueprint
Type	New Blueprint for the Edge
Blueprint Family	It is a independent blueprint, NOT a blueprint family yet.
Use Case	MEC platform used for Connected Vehicle.
Initial POD Cost (capex)	The Minimum Configuration: 4 Servers in total
Scale & Type	MEC Platform (1 Server) + 1 App Server (1 Server) + 2 Simulators (2 Server)
Applications	The MEC platform which can be used to connect vehicles, the general data flows are itemized below: 1. Grab the traffic/vehicle information 2. Dispatch the traffic/vehicle information to the corresponding edge process unit. Note well: The dispatch policy can be configurable. 3. Process the data in the Edge or Cloud and figure out the suggested action item for the vehicle driver Send the suggested action items to the vehicle driver
Power Restrictions	Less than 6KW. The Maximum Power consumption for each server is around 1500W, thus in total 1500 * 4 = 6000W

[9] https://wiki.akraino.org/display/AK/Connected+Vehicle+Blueprint

Infrastructure orchestration	• Docker + K8s • VM and OpenStack/StarlingX
PaaS	Tars and OpenNESS
Network	OVS, DPDK, VPP
Workload Type	Bare metal, VM, Container

The following figure clarifies also how OpenNESS fits into the Akraino Connected Vehicle Blueprint.

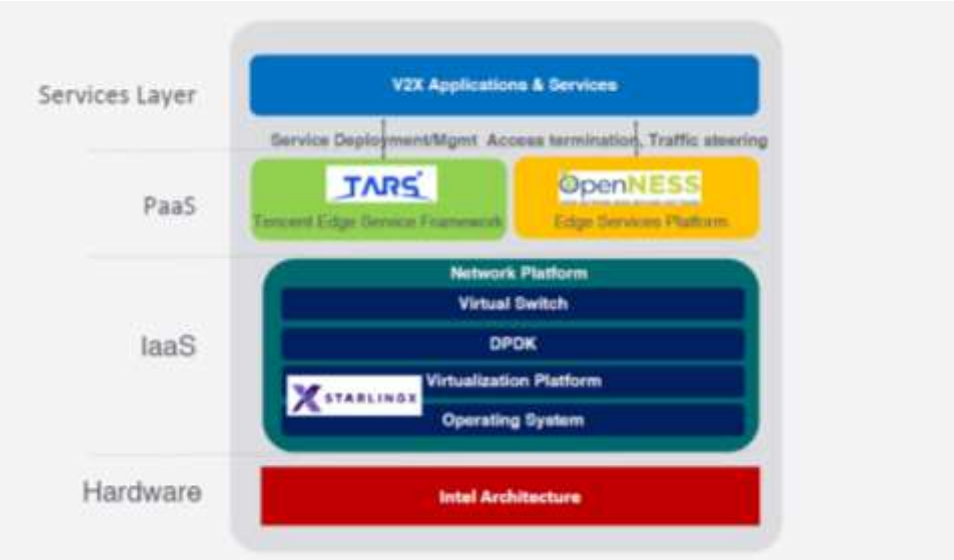

Figure 19 - Akraino Connected Vehicle Blueprint.

4.3 Life cycle management of edge applications

We see edge applications as an evolution of cloud applications. So, there is an inherent expectation for edge application life cycle management to leverage the best practices of cloud native applications e.g., DevOps and CI/CD (Continuous Integration / Continuous Delivery). With this paradigm the main requirements coming from two categories of stakeholders (Developers and IT Ops) are combined together (see figure below).

Figure 20 - System requirements in a DevOps paradigm.

Edge cloud infrastructure must provide the following sets of service for Edge Application Life Cycle Management:

1. Application Provisioning Service
    a. Compute, Storage and Network Provisioning.
    b. Application Package and Artifact Provisioning.
    c. Security and QoS Provisioning.
2. Application Monitoring and Diagnostic Service
3. Application Auditing and Accounting Service.

As the application will be executing in a multi-tenant environment, it is expected that the edge cloud infrastructure provide strong hardware/software enforced application isolation to protect applications privacy, confidentiality and resource allocations. We also expect edge cloud infrastructure to provide various level of resource abstractions to the application

Abstraction	Compute	Storage	Network
IaaS	Virtual Machine	Virt BLK	Virt Net
CaaS	Container	Storage Plugin	CNM/CNI
Serverless	Functions	SQL/NoSQL/Object Storage API	HTTP API EndPoints

Edge applications have some unique infrastructure requirement that is not necessarily applicable to cloud native application e.g.:

- Secure, Scalable, Manageable and Multi-tenant;
- Edge/Cloud Asset (Pi to Rack) lifecycle management. (Config, Security and Ownership etc.);
- Intelligent Automatic fluid/seamless function delivery and execution (cloud to edge);
- Intelligent/Transparent Automatic State and Data mirror/shadow;
- QoS/SLA driven and real-time (for mission critical and time sensitive applications);
- Zero-touch (plug-n-play) edge;
- Serverless (No Ops, Transactional billing, auto application reliability/scalability);
- Uniform programing model.

5 Conclusions

Edge Computing (often named also as MEC) is recognized as a key technology, supporting innovative services for a wide ecosystem of companies. The focus is to address the needs of two stakeholder categories in this ecosystem: infrastructure owners (e.g., operators and cloud providers) and software developers (e.g., applications / content providers, innovators and startups). In fact, both stakeholder categories are key to the success of MEC, and their engagement depends also on the way the various challenges are addressed.

In particular, in this paper we provided an overview of standardization efforts, including initiatives from industry groups, associations, open source communities and projects (e.g., OpenNESS). We also derived some edge deployment considerations (of high importance for infrastructure owners), trying to answer the recurrent question "Where is the edge?". Since many use cases are in principle supported by MEC (and indeed, MEC performance depends on the specific KPIs derived from these use cases), in order to provide a more specific analysis we describe an exemplary case study customized to the automotive domain. Such an example aims to constitute a useful reference for the software developer (i.e., the other main category of MEC stakeholders), when it comes to answering to the other recurrent question "How to use MEC from application development point of view?").

In the end, the overall market success of Edge Computing will depend, on how infrastructure owners will be able to address all deployment aspects and also on how software communities will create a wide set of applications and innovative services. Intel is committed to engaging the entire ecosystem, considering standard solutions and open source projects, towards ensuring interoperability, while opening the market to proprietary implementations and added-value propositions.

Appendix - OpenNESS Applications and Services

The Open Network Edge Services Software (OpenNESS) Toolkit is an open source software toolkit that enables orchestration and management of edge services on diverse platforms and network technologies. OpenNESS was introduced in Section 4 of this White Paper. In this Annex, we present the behavior of OpenNESS Applications and Services.

The OpenNESS concept of a service is derived from, and similar to, a service in the ETSI MEC standard. In the ETSI MEC standard, an application may provide a service to other applications. The reader is directed to Reference [6] for additional details, while a simplified description is provided here.

- An ETSI MEC service registers itself by sending a message to the MEC platform. Other applications ("relevant" applications, as documented in [6]), which have subscribed to the service, receive notifications from the MEC platform that the new service is available. Thereafter, until the service deregisters itself, the service, according to its operation, sends notifications to the subscribed applications. The registration, subscription, and notification operations are carried out via RESTful APIs defined by the ETSI MEC standard. Subscribing applications arrange to receive notifications by providing the MEC platform with a callback reference URI, which is shared with the service.
- In addition to this functionality, the ETSI MEC standard defines several designated services, including Radio Network Information (RNI) and Location services. Their functional behavior, with respect to subscription and notification, is analogous to that of application services.

OpenNESS applications and services behave similarly to ETSI MEC Apps and services, but include additional functionality for secure communication between an application or service and the edge platform.

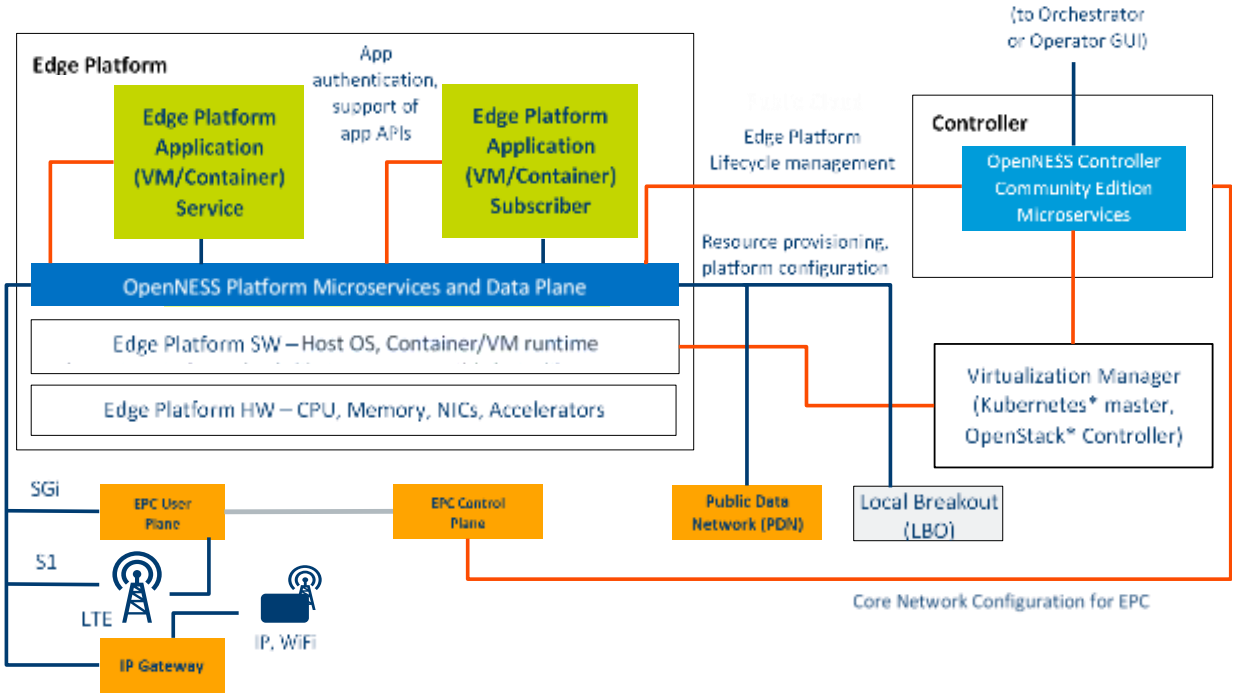

Figure 21 - OpenNESS applications and services.

The architecture of an edge platform, including applications, is shown in Section 2. The primary changes in the OpenNESS toolkit are:

- Communication between an application/service and the edge platform occurs over a secure socket, which is established by the application prior to registration or subscription operations.
- Services and applications find each other via an agreed identifier that is in a namespace, to support multiple independent applications.
- The edge platform provides the transport for the notification from service to all subscribed applications.
- The OpenNESS distribution does not include the standard ETSI MEC Services, such as RNI or Location. It is expected that system integrators will provide implementations of these services, which could be done via the OpenNESS APIs.

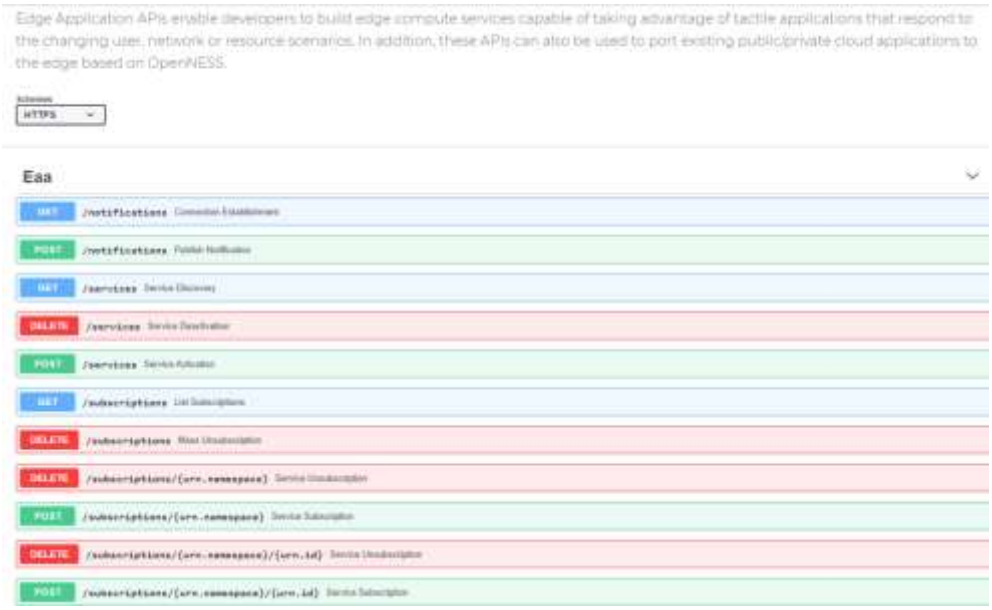

Figure 22 - OpenNESS Edge Application Agent (EAA) APIs

OpenNESS APIs are defined as RESTful APIs, documented according to the OpenAPI standard. They may be found at https://openness.org/api-documentation/?apieaa#/ .

The OpenNESS project is committed to working with SDOs and other special interest groups (SIGs) to enable the rapid deployment of edge services in the marketplace.

References

[1] iGR White paper: "The Business Case for MEC in Retail: a TCO analysis and its Implications in the 5G era", https://www.intel.com/content/www/us/en/communications/multi-access-edge-computing-brief.html

[2] Chetan Sharma Consulting: "5G Mobile Edge Computing: Redefining the Sports Experience", Link: https://builders.intel.com/docs/networkbuilders/5g-mobile-edge-computing-redefining-the-sports-experience.pdf

[3] Ericsson blog: "Edge computing success—a distributed cloud approach", Link: https://www.ericsson.com/en/blog/2018/9/edge-computing-successa-distributed-cloud-approach

[4] ETSI ISG MEC website, Link: https://www.etsi.org/technologies/multi-access-edge-computing

[5] ETSI GS MEC 003 V2.1.1 (2019-01), "Multi-access Edge Computing (MEC); Framework and Reference Architecture ", https://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/02.01.01_60/gs_MEC003v020101p.pdf

[6] ETSI MEC APIs, Link: https://www.etsi.org/committee/1425-mec

[7] 3GPP TS 23.501 System architecture for the 5G System (5GS), http://www.3gpp.org/ftp//Specs/archive/23_series/23.501/23501-g20.zip

[8] 3GPP SA5 Study on management aspects of edge computing, FS_MAN_EC, https://portal.3gpp.org/desktopmodules/WorkItem/WorkItemDetails.aspx?workitemId=800039

[9] 3GPP SA2 Study on enhancement of support for Edge Computing in 5GC, FS_enh_EC, https://portal.3gpp.org/desktopmodules/WorkItem/WorkItemDetails.aspx?workitemId=830032

[10] 3GPP SA6 Study on Application Architecture for enabling Edge Applications (FS_EDGEAPP), https://portal.3gpp.org/desktopmodules/WorkItem/WorkItemDetails.aspx?workitemId=830008

[11] Industrial Internet Consortium: "The Industrial Internet of Things - Volume G1: Reference Architecture", version 1.9, June 19, 2019, Link: https://www.iiconsortium.org/pdf/IIRA-v1.9.pdf

[12] GSMA Cloud AR/VR Whitepaper, available at: https://www.gsma.com/futurenetworks/resources-2/gsma-online-document-cloud-ar-vr-whitepaper/

[13] ISO/IEC 23090-8: "Information technology -- Coded representation of immersive media -- Part 8: Network-based media processing".

[14] 3GPP TS 26.118: "3GPP Virtual reality profiles for streaming applications".

[15] ISO/IEC 23090-2: "Information technology -- Coded representation of immersive media -- Part 2: Omnidirectional media format".

[16] VR Industry Forum Guidelines, available at: https://www.vr-if.org/guidelines/

[17] M. Król et al.: “RICE: Remote Method Invocation in ICN”, ACM ICN Conference, 2018

[18] Lixia Zhang, KC Claffy, Patrick Crowley, Christos Papadopoulos, Lan Wang, Beichuan Zhang, “Named Data Networking”, ACM SIGCOMM Computer Communication Review, Vol 44, No. 3, July 2014

[19] Christian Tschudin, Manolis Sifalakis, “Named Functions and Cached Computations”, IEEE 11th Consumer Communications and Networking Conference (CCNC), 2014

Intel Notices and Disclaimers

Intel provides these materials as-is, with no express or implied warranties.

All products, dates, and figures specified are preliminary, based on current expectations, and are subject to change without notice.

Intel, processors, chipsets, and desktop boards may contain design defects or errors known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No product or component can be absolutely secure.
Check with your system manufacturer or retailer or learn more at http://intel.com.

Some results have been estimated or simulated using internal Intel analysis or architecture simulation or modeling, and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance.

Intel and the Intel logo are trademarks of Intel Corporation in the United States and other countries.

*Other names and brands may be claimed as the property of others.