Cloud-based XR Services: A Survey on Relevant Challenges and Enabling Technologies

Theodoros Theodoropoulos[1], Antonios Makris[1], Abderrahmane Boudi[2], Tarik Taleb[3], Uwe Herzog[4],
Luis Rosa[5], Luis Cordeiro[5], Konstantinos Tserpes[1], Elena Spatafora[6], Alessandro Romussi[6],
Enrico Zschau[7], Manos Kamarianakis[8], Antonis Protopsaltis[9], George Papagiannakis[8], and Patrizio Dazzi[10]
[1]Department of Informatics and Telematics, Harokopio University, Athens, Greece
[2]ICTFICIAL, Espoo, Finalnd & Ecole Nationale Supérieure d'Informatique, Algiers, Algeria
[3]University of Oulu, Oulu, Finland
[4]Eurescom GmbH, Heidelberg, Germany
[5]OneSource, Coimbra, Portugal
[6]HPE, Cernusco Sul Naviglio, Italy
[7]SeeReal Technologies, Dresden, Germany
[8]ORamaVR & FORTH-ICS & University of Crete, Heraklion, Greece
[9]ORamaVR & University of Western Macedonia, Kozani, Greece
[10]CNR, Pisa, Italy

In recent years, the emergence of XR (eXtended Reality) applications, including Holography, Augmented, Virtual and Mixed Reality, has resulted in the creation of rather demanding requirements for Quality of Experience (QoE) and Quality of Service (QoS). In order to cope with requirements such as ultra-low latency and increased bandwidth, it is of paramount importance to leverage certain technological paradigms. The purpose of this paper is to identify these QoE and QoS requirements and then to provide an extensive survey on technologies that are able to facilitate the rather demanding requirements of Cloud-based XR Services. To that end, a wide range of enabling technologies are explored. These technologies include e.g. the ETSI (European Telecommunications Standards Institute) Multi-Access Edge Computing (MEC), Edge Storage, the ETSI Management and Orchestration (MANO), the ETSI Zero touch network & Service Management (ZSM), Deterministic Networking, the 3GPP (3rd Generation Partnership Project) Media Streaming, MPEG's (Moving Picture Experts Group) Mixed and Augmented Reality standard, the Omnidirectional MediA Format (OMAF), ETSI's Augmented Reality Framework etc.

Index Terms—**Edge Computing, XR services, Holography, Cloud Computing.**

I. Introduction

The term Cloud-based services refers to applications which can be consumed by users via the Internet. These applications run on shared computational resources which are distributed over multiple locations. This paradigm is extremely beneficial in the context of computationally intensive applications since it enables users to remotely have access to the necessary computational resources. Extended Reality (XR) is a class of computationally intensive applications whose aim is to minimize the gap between the digital and the physical world. It encompasses a wide range of applications such as Virtual Reality (VR), Augmented Reality (AR) and Mixed Reality. XR applications are extremely demanding in terms of computing and storage resources since they require various XR assets, such as 3D models. In the context of the monolithic way of developing applications, these resources would have to be incorporated in the dedicated XR equipment, thus making this endeavour prohibitively expensive and/or bulky. Cloud-based XR applications [1] can provide a solution to this problem.

Recently, with the rapid emergence of XR applications, including Augmented, Virtual and Mixed Reality and Holography, there have been numerous challenges that the scientific community needs to overcome. These challenges are intertwined with the very fabric of theses types of applications. While each type of application presents a distinct set of Quality of Experience (QoE) and Quality of Service (QoS) requirements, there are certain characteristics that they all have in common and that need to be taken into consideration when contemplating the frameworks that next-gen XR applications will be built on. The most significant one is the need for ultra-low and ultra-high bandwidth. Studies have shown that in order to provide an acceptable end-user experience in regards to XR applications, the end-to-end latency shall be less than 15ms and the bandwidth should be able to scale up to 30 Gbps. Well established network concepts such as Best-Effort and simple traffic differentiation are not able to meet such demanding requirements. Even despite the rapid advances of 5G technologies, the process of facilitating this class of applications still remains quite challenging. Edge Computing may help alleviate some of the burden which is caused by Cloud-based XR applications. Edge Computing enables the functionality of data processing to be conducted closer to either where the services are consumed or where the data is generated, thus reducing the overall end-to-end latency and the required bandwidth. Finally, Cloud-based XR applications run on multiple heterogeneous resources. Thus, it is vital to incorporate certain management and orchestration

Manuscript received December 13, 2021; revised February 8, 2022. Corresponding author: Theodoros Theodoropoulos (email: ttheod@hua.gr).

technologies that are able to accommodate the complexity that derives from latency-sensitive and bandwidth-sensitive Cloud-based XR applications. Taking these factors into consideration, it becomes apparent that in order to facilitate Cloud-based XR applications, an amalgamation of various Edge Computing, Orchestration, Network and application-oriented technologies is required. The purpose of this survey paper is to explore these technologies and to examine how they can contribute towards establishing an operational computing and network continuum that is able to facilitate the rather demanding requirements of Cloud-based XR applications.

This survey paper is organized in the following manner. In section II, several architectural frameworks that are able to carry out the various orchestration and management processes of Cloud-based XR applications are introduced. In section III, a number of Edge Computing and Storage solutions are explored. In section IV, various aspects of the 3rd Generation Partnership Project (3GPP) are explored in a manner that highlights their significance and potential to act as enablers of XR applications. In section V, some notable networking solutions that can enable the facilitation of the QoS requirements of Cloud-based XR applications are explored. In section VI, several XR-relevant standards are introduced. Finally, section VII demonstrates the conclusions regarding how these technologies may accelerate the implementation process of XR applications.

II. Management & Orchestration

The requirements of Cloud-based XR applications have to be accommodated in a landscape which is comprised of numerous, heterogeneous network assets. Nowadays, cloud-based frameworks are required to facilitate an unprecedented number of computational and network assets. Furthermore, these resources may be part of different domains and/or located at entirely different regions. Thus, the complexity of the orchestration of cloud-based resources is rather high. As a result, it is of paramount importance to leverage various orchestration and management technologies which can provide a certain degree of automation and guarantee that the QoE and QoS requirements will be met.

A. *ETSI Management & Orchestration*

In 2012, fostered by the main players of the telecommunication market, ETSI created an ISG (Industry Specification Group) gathering more than 150 companies and research institutions, with the aim of creating the first standardized NFV (Network functions virtualization) reference architecture, and technical specifications for its specific components. This standardization effort is now at its Release 4, and has spawned a project providing an open reference implementation of its model, named Open Source MANO (OSM). In June 2021, OSM launched its Release 10.

From an architectural point of view, NFV specifications describe and specify virtualisation requirements, NFV architecture framework, functional components and their interfaces, as well as the protocols and the APIs (Application Programming Interfaces) for these interfaces. ISG NFV sets also specifications defining (in structure and format) how the deployment must be done, which features must be activated for VNF (this information are included in the deployment template), how all artefacts must be organized to be computable by the MANO framework. In addition, ISG NFV specification also covers security aspects related to virtualisation as well as performance, reliability and resiliency matters. The 5G emergence induced ISG NFV to approach new technical contents, such as multi-site and multi-domain deployments and network slicing.

Related to new virtualisation technologies such as support for containerized VNFs (c-VNF) and container infrastructure management, in November 2020, ETSI has published its first specification enabling containerized VNFs to be managed in a NFV framework, namely ETSI GS NFV-IFA 040; the last revision is in [2]. This specification describes functions required for the management and orchestration of containers, the container infrastructure service management (CISM) responsible for maintaining the containerized workloads and manages the container, computation storage, network resources and their configuration, and the container image registry (CIR) responsible for storing and maintaining information of container software images.

The ETSI-MANO architectural framework identifies functional blocks and the main reference points between such blocks. The goal of the framework [3] is to obtain a high-level architecture where the software implementing the network functions, is decoupled from the specific type of hardware and software used to manage the physical infrastructure. The decoupling exposes a new set of entities, the Virtualised Network Functions (VNFs), and a new set of relationships between them and the NFV Infrastructure (NFVI). VNFs can be chained with other VNFs and/or Physical Network Functions (PNFs) to realize a Network Service (NS). To do this, it is necessary to have an upper level of management and orchestration for the virtualised resources.

Figure 1 represents the complete architectural framework defined by ETSI. The Network Functions Virtualisation Infrastructure (NFVI) represents the infrastructure of an NFV environment, providing hardware and software resources for the instantiation of network functions: CPUs, GPUs, memory, storage, network devices and software to virtualize resources (i.e., hypervisor-based or Container-based virtualisation). The NFVI infrastructure is designed to be distributed, so the NFVI nodes are decentralized in many locations (PoP's) supporting the locality and latency requirements necessary in some use cases. Inside this component, ETSI distinguishes three different domains: Compute Domain [5], Hypervisor Domain [6], and Infrastructure Network Domain [7]. In addition, it guarantees communication between VNFs and the Network Functions Virtualisation Orchestrator (NFVO), and between NFVI and NFVO. Virtualized Network Functions (VNFs) are the result of the NFVI layer virtualisation, implementing specific network functions like routers, load balancing, firewalls as software applications. Element Management (EM) performs the typical management functionality for one or several VNFs. It is responsible for FCAPS functions (Fault, Configuration, Accounting for the usage of NFV, collecting

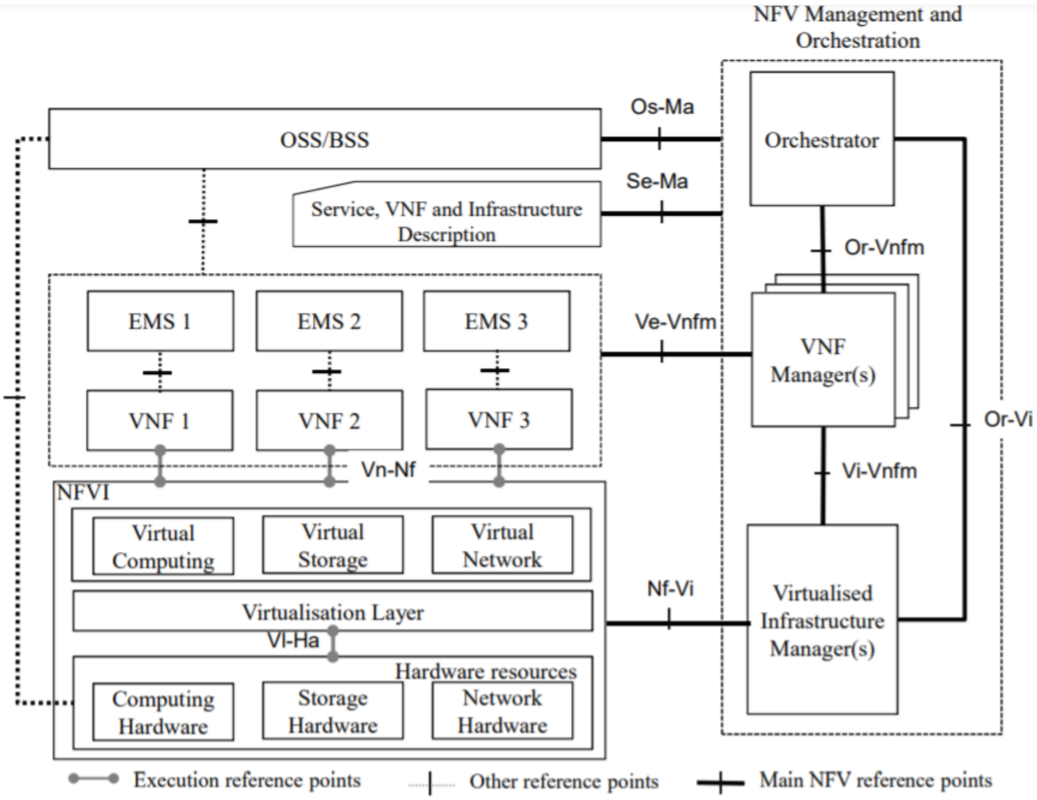

Fig. 1. NFV Reference Architectural Framework [4]

Performance measurement results for the functions provided by the VNF, Security management) for the VNFs. Operations Support Systems and Business Support Systems (OSS/BSS) provides essential business functions and applications such as operations support and billing. In a world that is quickly evolving, it is necessary for traditional OSS/BSS to adapt itself to new service flexibility, real-time service variations.

NFV Management and Orchestration (MANO) [8] is the component acting as coordinator/orchestrator of the whole NFV system. While the decoupling of VNFs from hardware shows many advantages in terms of flexibility, on the other hand, it carries out the need of a complex management layer that can handle different NFVs, different locations where they have to be instantiated in order to maintain appropriate service level, can allocate and scale in, out, up, down hardware resources to the VNFs, can manage the entire life cycle for each VNF, etc. These tasks (and many others) are referred to NFV MANO component. ETSI MANO provides methodologies to coordinate network service deployment operations and mechanisms for managing network functions. Some of the main functionalities offered by an NFV MANO module are as follows:

- NFVI virtualised resource management: availability, allocation and release
- NFVI performance management
- instantiation, resources scaling, updating, modification and termination of a VNF
- instantiation, scaling (up or down), updating, modification and termination of a Network Service (NS)
- VNF or NS on-boarding

In addition to these functions, there are many others that allow, for example, VNF and NS monitoring through predefined metrics. These modules are able to evaluate when it is necessary to scale up or scale down the resources allocated to specific network service or network function.

ETSI MANO is composed by three main functional blocks: NFV Orchestrator (NFVO), VNF Manager (VNFM) and Virtual Infrastructure Manager (VIM). In addition to these, MANO architecture encloses also some data repositories like NS and VNF descriptor Catalogue, NFV Instances repository and all the resources needed to the VNF and to NS to be instantiated (NS/VNF catalogue, NFV instances and NFVI resources). NFV Orchestrator is the core element in the management of NFV architecture, in charge of NFV infrastructure orchestration and life cycle management of Network Services. In these responsibilities it is supported, at a lower level, by Resource Orchestrator (RO) for the first one and by Network Service Orchestrator (SO) for the second one at a higher level. RO manages the resources orchestration with the help of one or more Virtualised Infrastructure Manager (VIM): it decides their allocation in one or more NFVI-PoP (Point of Presence of NFVI, typically a node with resources located physically in the same point), and keeps track of the instances and resources that have been allocated for each single VNF in repositories. The NSO works at a higher level, looking after the instantiation and management of Network Services as a whole, with their composing VNFs. The VNFM handles the life cycle of the NFV or NS (instantiation, update, query, scaling, and termination). It can handle a single VNF or more VNFs, and often works with Element Management System (EM or EMS). The VIM is dedicated to a high level management of all (compute, storage, network) NFVI resources or only for certain type of NFVI resource (e.g. compute-only, storage-only, networking-only). It is responsible for low-level orchestrating the allocation (including the optimization of such resources usage) of the resources needed for the deployment of VNFs, keeping an inventory of virtualised resources mapping on physical resources. It also collects and notifies to the affected components information regarding infrastructure performance and malfunctions. In summary, VIM keeps an inventory of many kinds of resources, validates the requests coming from the Orchestration layer/module and executes them into the proper infrastructure layer.

B. *ETSI Zero touch network & Service Management (ZSM)*

5G technology and network slicing are reshaping the classical approach of how the services and infrastructure were orchestrated in the past. Service, telecommunication, and infrastructure providers are now looking into more flexible ways of having fully automated and E2E service management spanning under the umbrella of distinct domains, both administrative and technological. To address such a problem, the ZSM specification group from ETSI was formed to discuss relevant use cases, requirements, and specify an end-to-end management reference architecture that allows such E2E service deployments.

Figure 2 shows the ZSM framework reference architecture proposed by ETSI [9]. This architecture is conceptually composed of six building blocks: Management Services, Management Functions, Management Domains, E2E Service Management Domain, Integration Fabric and finally, Data Services.

ZSM Management Services, exposed through specific endpoints, allow a more consistent and standardized way to expose different management capabilities across a multi-domain deployment. ZSM Service capabilities are offered (produced) and/or used (consumed) by Management Functions. Multiple capabilities can be combined to form broader abstractions of management features. Indeed, collaborative, and federated service orchestration models were also considered in ZSM, playing a relevant role in scenarios involving different operators. Such Management Services are then organized by functionality into Management Domains. Within each domain, there can be internal or exposed services, depending on whether their access is restricted to a domain or exposed to outside of the domain. ZSM specification also considers the possibility of hierarchy at the Management Domains where multiple domains are recursively stacked on top of each other. Moreover, the ZSM framework specifies an E2E Service Management Domain responsible, among others, for end-to-end (E2E) orchestration across different domains, E2E closed-loop management, E2E analytics, and data collection.

Another key building block proposed in the ZSM framework is the concept of Integration Fabrics. They are meant to facilitate the communication between management functions. Indeed, ZSM defines two types of Integration Fabrics: Domain Integration Fabric and a Cross-Domain Integration Fabric. The first one is responsible for connecting services within the same domain. While the second is used to facilitate communication over distinct domains. Note that such fabrics are not only used as a communication bus between services but to facilitate the registration, discovery, and invocation of the different supported services. Finally, ZSM comprehends the concept of Data Services which allows to decouple and reuse the same management data across distinct management services.

The ZSM framework was conceived based on a set of principles such as modularity, extensibility, scalability, model-driven, open interfaces, closed-loop management automation, support for stateless management functions, resilience, separation of concerns in management, service composability, intent-based interfaces, functional abstraction, and simplicity. Together, these principles allow obtaining a future-proof design architecture that can be used to fully automate (i.e. Zero-Touch concept) the network and service management, which, again, can span across multiple domains, both administrative and technological. Likewise, the ZSM specification defines a set of requirements that shall be satisfied by a given ZSM implementation. They range from non-functional requirements such as the need to be vendor, operator, and service provider agnostic to functional requirements such as the proposed support for adaptive closed control loops or the E2E and cross-domain support. ZSM specification group also defined a set of security-specific requirements covering aspects such as the need for confidentiality and integrity of management data at rest, in-transit and in-use. Those requirements were based on a set of related use cases scenarios as described in [10]. The complete list of requirements can be found in the ZSM Reference Architecture specification.

More than E2E deployments, the underlying idea of ZSM is to achieve a level of automation where closed-loop processes and algorithms (e.g., machine learning-based orchestration mechanisms) can drive more efficient and flexible scenarios (e.g., a self-monitoring and optimization of the network) and ultimately reduce (or eliminate) the need for human intervention. Indeed, the concept of Closed Loops, which can occur at both Management Domain and E2E Domain levels, as shown in Figure 3, is further discussed in an additional ETSI specification [11].

Each of those loops is logically split into a set of ordered stages accordingly to some of the well-known closed-loop approaches (e.g., OODA loop (Observe, Orient, Decide, Act) or MAPE-K (Monitor, Analyse, Plan, Execute)). Closed-loop stages can be used for realizing different management functions, such as network fault diagnosis and mitigation. In such a network fault example, data is constantly being collected from the network (Observe) and whenever an abnormal condition is detected, the issue is analyzed (Orient). Then a subsequent evaluation decides what should be done (Decide) and finally, an appropriate solution is applied to fix the issue (Act).

C. *ETSI Experiential Network Intelligence (ENI)*

During the last decade, there have been numerous instances where the implementation of Artificial Intelligence in real-

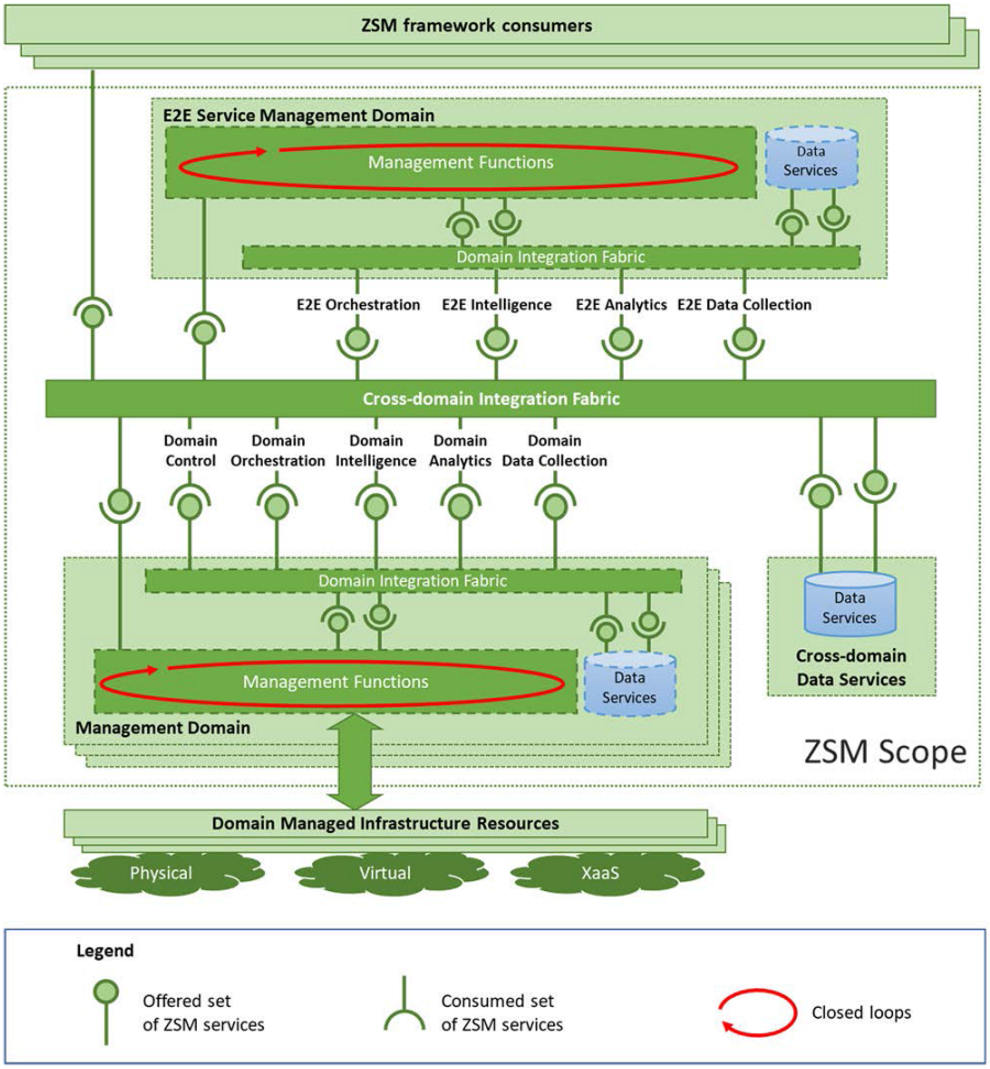

Fig. 2. ZSM framework reference architecture [9]

word applications has provided ground-breaking results. In order for networks to leverage the functionalities enabled by the use of Artificial Intelligence methodologies the European Telecommunication Standards Institutes introduced the Experiential Network Intelligence (ENI) framework. The purpose of this framework is to provide guarantees in regards to the network's ability to keep up with the Quality of Service requirements [12]. More specifically ENI is able to assist or direct network management systems based on network status and Service Level Agreements. The ENI entity is in charge of providing recommendations or commands to an Assisted System (AS), in order for intelligent network management to be established. It is possible for the ENI to communicate with the Assisted Systems via an Application Programming Interface (API) broker that performs the appropriate translations. Up to this point, three classes of AS have been identified based on the degree that the various Artificial Intelligence mechanisms influence the management and orchestration of the network [13].

The backbone of the ENI architecture is the implementation of closed control loops. These closed control loops are based on the Observe-Orient-Decide-Act (OODA) paradigm [14]. Furthermore, there are two distinct types of control loops. The inner control loops, each one of which consists of multiple loops that enable the basic OODA functionalities to be conducted in parallel. The outer control loops that guarantee that the overall OODA process is carried out in accordance to a desired output state. The various phases of the OODA loop process are implemented by utilizing some predefined functional blocks. The "Observe" phase is the product of the Input Processing and Normalization functional blocks. This structure is in charge of cleansing, curating and combining the various heterogeneous data inputs. Furthermore, the Processing and Normalization functional block consists of the Data Ingestion and the Normalization functional blocks. The purpose of these blocks is to handle the various data formats and then alter them in a way which is compliant with the format of the entities that are expected to receive them.

The "Orient" phase of the OODA loop is based on the use of the Knowledge Management, Context-Aware Manage-

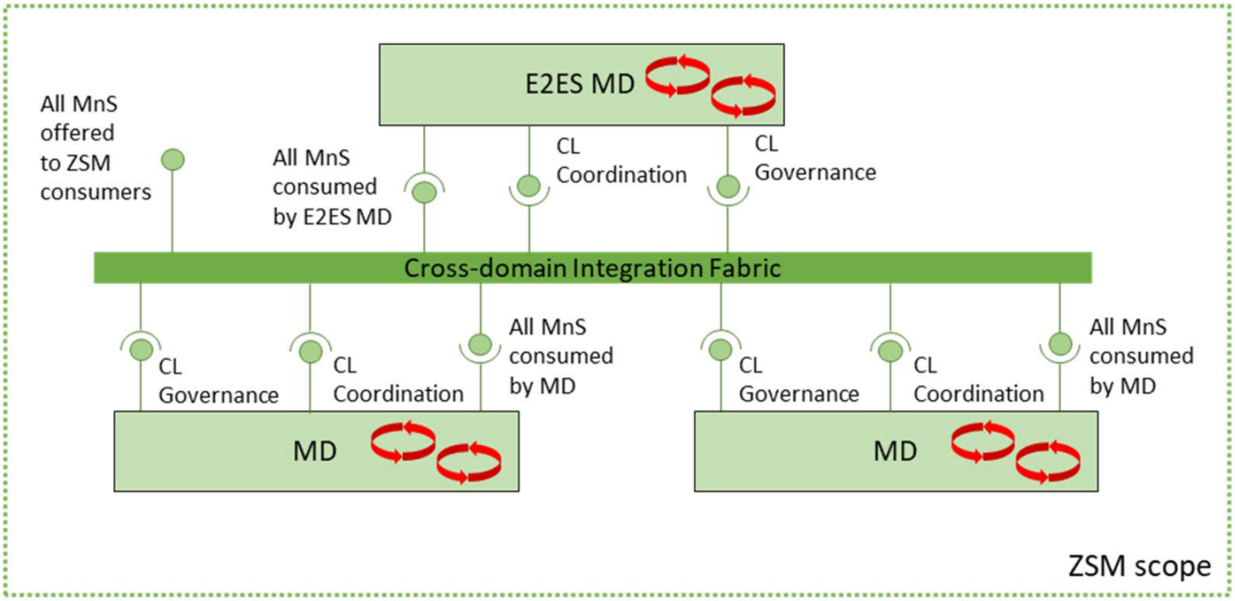

Fig. 3. Closed Loop and related management capabilities [11]

ment, Cognition and Situational Awareness functional blocks. The Knowledge Management functional block is able to facilitate Machine Learning and formal logic by providing formal representation of the ingested data. The Context-Aware Management functional block monitors closely the changes in the state of the environment in order to provide adaptability. The Cognition functional block is responsible for establishing high-level goals in regards to optimizations and Service Level Agreements, which are either learned or provided by the operator. Finally, the Situational Awareness functional block examines the events taking place and evaluates the potential impact of various available actions on the future state of the Assisted System. The “Decide” phase of the OODA process relies on the Model-Driven Engineering and Policy Management functional blocks. The former takes a set of suggested changes and translates them into a set of actions. The latter turns these actions into reusable policies. The “Act” phase is carried out by utilizing the Denormalization and Output Generation functional blocks. These blocks are in charge of producing a denormalized output which can be digested by the various Assisted Systems. In some cases, the API broker may be required to perform additional translations before the output gets transferred to the appropriate Assisted System.

In the case of XR applications, it is of major importance to be able to provide carrier grade assurance capabilities in order to ensure that the QoS requirements will be met. Capabilities such as these heavily rely on the need to conduct optimal resource allocation between competing network slices. The QoS requirements are provided by the customers in the form of Service Level Agreements. The various SLAs are then transferred through the OSS/BSS Assisted System to the Orchestration and Management Assisted System and the ENI system. The SLA requirements are then processed by the Data Ingestion and Normalization functional blocks. During the same time-frame, the Infrastructure Assisted System establishes and configures the corresponding network slices according to the blueprints created by the OSS/BSS Assisted System. This process leads to the normal operational phase of the ENI System, during which the Data Ingestion and Normalization functional blocks gathers raw data from the environment. The normal operational phase is interrupted when an abnormality in regards to the expected resource consumption occurs. The Situational Awareness functional block is responsible for storing the associated configurations up to the point that the abnormality occurred. Then, it is up to the Situational Awareness and the Model-Driven Engineering functional blocks to operate together in order to prevent any potential violations of the SLAs from taking place. The Situational Awareness functional block is in charge of deciding which one of the suggested actions will be taken. After the appropriate plan has been decided, it is up to the Model-Driven Engineering functional block to translate it into a set of commands that can be interpreted by the various network components. The role of the Policy Management functional block is to translate the action plan into a set of policies, which are then forwarded to the Denormalization and Output Generation functional blocks. These blocks translate the policies into a format which can be understood by the Assisted Systems.

Given the increasing spread of XR services, it is becoming more and more pressing for these services to adopt the cloud paradigm from one side and the microservices paradigm from the other side. First, and in order to adopt these two paradigms, XR services need to embrace the NFV framework. This is why the principles of ETSI MANO are very important in the development of new XR services. With the recent support of containerized and hybrid workloads in ETSI MANO, XR services are the natural choice to take advantage of this new capability. Indeed, this type of service relies heavily on both of heavy legacy applications running on VMs and new applications following microservices concepts. Secondly, following the microservice concepts would result in an unprecedented load of management and orchestration in XR applications. Therefore, following ZSM and ENI principles, they can alleviate the load on human operators by shifting it towards autonomous systems. While ZSM is essential in providing closed loops automation at the service level, the ENI would introduce AI and enforce closed-loop automation at the infrastructure level.

III. Edge Computing & Storage

Recently, the Edge computing paradigm has been considered as a key enabler for addressing the increasingly strict requirements of next-generation applications [15]. Contrary to (traditionally centralized) Cloud Computing, in Edge Computing, the computational resources are placed closer to the end-users into the so-called edge. Amongst many others, this has the benefit of reducing the latency times. Moreover, Edge Computing significantly reduces the amount of data in transit towards remote clouds and enable data processing near the data sources. Ultimately, expanding the possibilities for more delay-sensitive and high-bandwidth applications that would not be feasible using cloud and far remote processing alone.

A. *ETSI Multi Access Edge Computing*

Multi-access Edge Computing (MEC) is a standardization initiative from the European Telecommunications Standards Institute (ETSI) and telecommunication industry to specify how Radio-Access Network (RAN) of telecommunication operators can be leveraged to realize the principles of Edge computing. MEC intends to provide a consistent path by specifying how multiple third parties (e.g., service providers) can use the last mile of telecommunication operator network and infrastructure to deploy their services and applications. Such edge deployments unfold the possibilities for the next-generation immersive XR services and open a variety of new business opportunities for all parties. Telecommunication providers have more ways to capitalize on their infrastructure. Traditional cloud providers can expand their offer to a new range of services leveraging the computation resources of telecommunication operators. Whereas, next-generation application developers can create innovative applications tailored for such ultra-low latency environments (i.e., the MEC applications).

The MEC Industry Specification Group (ISG) organized the MEC framework into three levels [16]: system, host, and network levels. Figure 4 provides the generic reference architecture of a MEC system, its functional elements, and the reference points between them.

The host-level includes the MEC host itself, comprised of the MEC applications, the Virtualization Infrastructure and the MEC platform, and their management counterparts (i.e., the Virtualization Infrastructure Manager (VIM) and the MEC platform Manager). The MEC platform includes all the functional blocks allowing for consuming and providing services (e.g., service registry, DNS handling and traffic control in general).

The system level is composed of a Multi-access edge Orchestrator (MEO) used to maintain an overall view of the MEC system (i.e., hosts and applications) and handle the life-cycle of each application. For instance, MEO selects the appropriate MEC hosts for deploying an application considering environment constraints such as latency or resource availability. The system level is also composed of the Operations Support System (OSS) (of the telecommunication operator) and the respective Customer-Facing Service (CFS), which together are responsible for instantiating and forwarding application requests to the MEO.

Finally, the network level refers to the external and network-related entities (e.g., 3GPP Network, Local Network, and External Network).

MEC specification [16] also describes how the MEC framework maps to the generic NFV reference architecture. In such a specification, both MEC applications and the MEC platform are treated as Virtual Network Function (VNF)s. The MEC Virtualization Infrastructure maps to the NFV Infrastructure (NFVI). Whereas, the MEC platform manager and MEC Orchestrator are replaced by the VNF managers and the NFV Orchestrator, respectively.

The manifold MEC possibilities, applications, scenarios, and the role of MEC in Cloud-Based XR services are further discussed in [17], [18] and [19]. Whilst in [20] the authors focus on video streaming scenarios, an increasing relevant application for XR services. The authors discuss the key technologies, resource allocation problems and optimization criteria. MEC specification [21] considers three main categories of use cases and applications that can benefit from the MEC paradigm: Consumer-oriented services (e.g., gaming, remote desktop applications, augmented and assisted reality, and cognitive assistance), Operator and third party services (e.g., active device location tracking, Big Data, security and safety, and enterprise services) and last but not least Network performance and QoE improvements (e.g., content/DNS caching, performance optimization or video optimization).

Indeed, an extensive list of 35 scenarios was already identified by MEC specification can be found in [21]. Two of them are here discussed as a reference to the possibilities of MEC: AR Caching Service and Video Analytics Application. They provide guidance on how the next generation of Cloud-based XR services can fully explore the advantages of the MEC paradigm.

The first scenario is the usage of MEC to support an AR service to cache and process the user location or camera view. This scenario has two main advantages: the offloading of functionalities from the limited end-user devices and avoiding deploying them into a remote cloud location, which would otherwise increase the latency and require the transfer of large amounts of data across the network. Here, MEC is especially important in densely populated locations (e.g., thousands of users in a stadium) where a high number of consumers will consume a given content. Moreover, such data processing at the edge allows the collection of additional related KPIs, which can then support service optimizations and improve the overall service QoE. Such (intermediate) edge processing, plays a pivotal role not only for caching but in numerous Cloud-

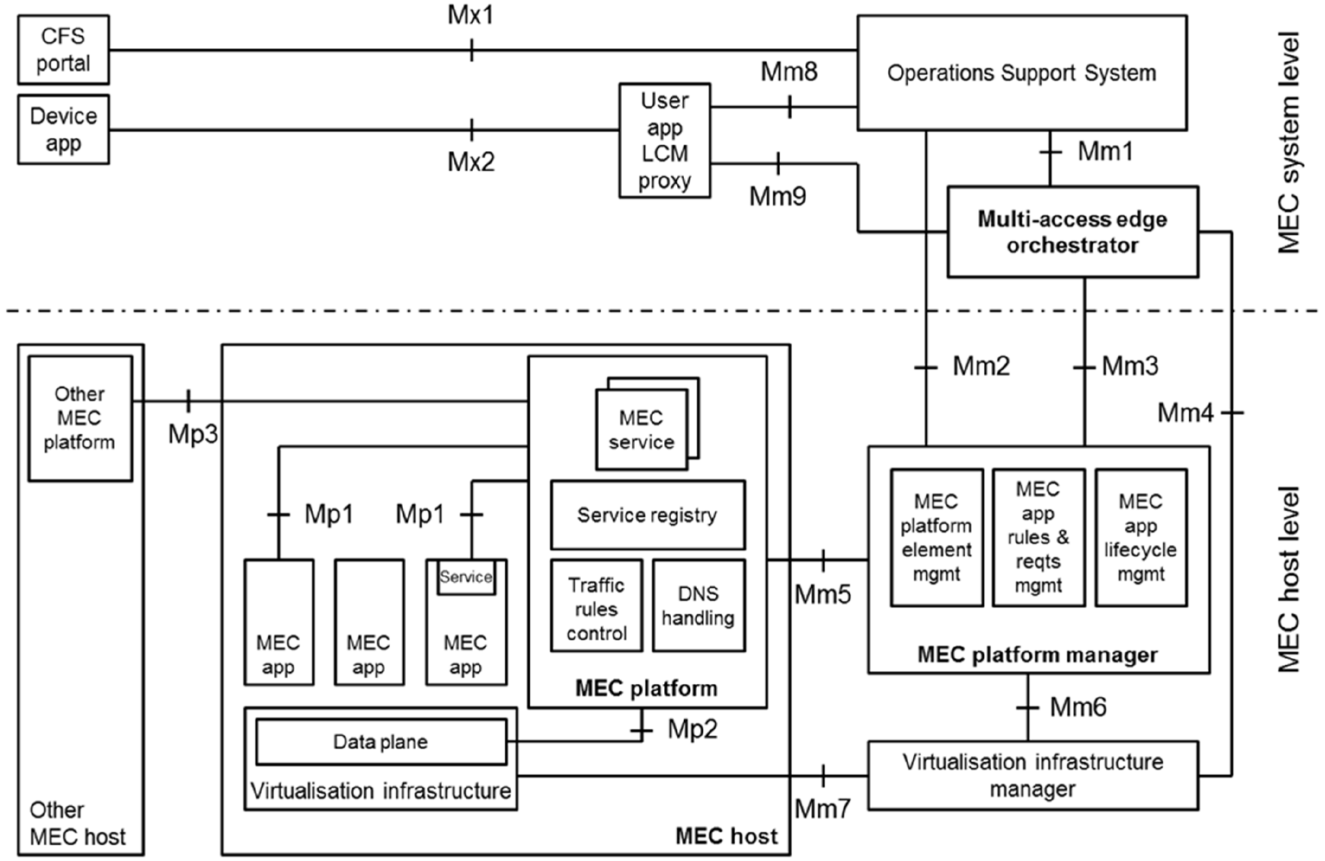

Fig. 4. Multi-access edge system reference architecture [16]

based XR scenarios. The second scenario refers to the usage of a MEC server to deploy a video analytics application, reporting the estimated network conditions (e.g., throughput available) back to the video server. With such information, the video server can then adjust the transmission characteristics to maximize the network capacity and improve the QoE.

The deployment of XR services across multiple technological and administrative domains is a open challenge. Without such integration of different MEC systems, use cases are limited to the users connected to the same MNO (e.g., AR games restricted to a specific geographical area). Contreras et al. [22] provided an overview of the integration options for MEC deployments across different administrative domains, including integration at the infrastructure level, at the platform level, at the service level and the interconnection between two MEC systems. Such comparison work helps to understand how multiple services (MEC applications in their case) could be integrated across such different deployments. The authors referred to the integration at the service level as the most straightforward option, highlighting that technical and interoperability issues can difficult such multi-domain deployments regardless of the chosen integration level. They considered that several open challenges need further investigation, including security, scalability, monitoring, accounting, or discovery automation aspects.

Likewise, one of the latest ETSI MEC Group Reports [23], discusses the various integration patterns for connecting multiple MEC systems. The report includes scenarios involving MEC systems from different Mobile Network Operators (MNO) and the increasing relevant interconnection between MEC and cloud systems. The report also addresses the high-level requirements of discovering other MEC platforms, how MEC platforms and applications can securely communicate, or the application relocation across different MEC systems. Moreover, they discuss a MEC federation scenario for a location-based immersive AR game, proposing two solutions. A first option where each MEC application is responsible for creating an AR gaming room and all are responsible for synchronizing gaming-related components (e.g., users' gameplay actions, players' positions). And a second option is where one of the MEC applications is used to instantiate the gaming room and is responsible for serving the different users connected to different MNOs, redirecting the traffic accordingly and transferring game-related details to the other applications instances.

As presented before, numerous use cases can take advantage of edge to, for instance, reduce the latency in network communications. Nevertheless, to explore the benefits of edge computing, several challenges need to be addressed, including decoupling traditional monolith applications into Cloud-based XR services and managing such complex Cloud-based XR scenarios at scale (e.g. a multi-domain environment).

B. Edge Storage

In edge computing, a large amount of data is generated and consumed by various edge applications. One of the key challenges in the development of applications at the edge, is the efficient data sharing among the multiple edge clients. Data sharing can be realized within individual application frameworks or through an external storage service. In general, edge computing moves the computational load to the edge of the network in order to exploit the computational capabilities of edge nodes. Moving data and computation closer to the clients results in latency minimization and also improves network bandwidth. Thus, edge storage can greatly improve data access which in turn enables latency-sensitive applications. Despite the recent advancements in providing an edge storage solution, there are still issues left to be dealt with. Some issues related to the non-functional requirements of cloud-based application. In addition, edge nodes generally have limited computation, storage, network, or power resources and the distributed, dynamic and heterogeneous environment in the edge along with the diverse application's requirements poses several challenges such as:

- coordination of unreliable devices and network;
- hardware and software incompatibilities that arise due to the plethora of different devices
- mobility of the devices and the users
- integration of different data storage formats and data types
- security and privacy concerns
- QoS & QoE insurance

To tackle these issues, one must leverage the core infrastructure and extend or integrate some of the most prominent software solutions such as MinIO[1], OpenStack Swift[2] or CEPH[3] with cloud-based storage services. Confais et al. [24] suggest that each edge storage system should employ the following properties: low access time, network containment between "edge clouds" (cloudlets), availability through partitioning and mobility. Object storage software systems such as those mentioned above, leverage Peer-To-Peer (P2P) mechanisms and are able to cope better with the proposed edge storage requirements. Clarke et al. [25] propose an epidemic approach for decentralized storage systems which offer data publication, replication and retrieval. However, the proposed approach presents limitations regarding resource discovery. Another decentralized storage system for edge computing is presented by Gheorghe et al. [26]. The tasks and the data are submitted first on gateway nodes, thus leading to network latency minimization. For solving the network discovery problem, a network device discovery is utilized which is able to find all the available devices for a certain application through the LAN router and forms a device tree for communication.

However, it takes a more elaborate solution so as to deal with the inherent unreliability of the edge devices. Research for the efficient data placement takes a prominent role in developing a reliable edge storage solution with security coming in as a follow-up concern when heterogeneous storage systems in edge and cloud nodes need to exchange data. In addition, regarding resource management, several challenges concerning the adaptation to the dynamic environments and the large-scale optimization for the collaboration of multiple edge servers must be addressed.

The literature presents multiple options regarding these topics. Near real-time decisions can be efficiently improved by moving the analytics ”close” to the data. As a result, edge architectures can reduce the amount of data traversing the network, thus minimizing latency and overall costs. Among the most relevant work, there are a layered approach for data storage management and an adaptive algorithm which dynamically finds the trade-off between the quality and the amount of data stored at the edge and the cloud [27]. In order to address the problem of limited storage space in edge computing and to reduce data loss caused by unstable networks, Xing et al. [28] proposed a distributed multi-level storage system model which is based on a multiple-factors LFU (Least Frequently Used) replacement (mLFU) algorithm. In general, replacement algorithms select parts of data to be removed from the current storage node when the storage space overflows. While direct algorithms like LFU are effective on the edge with restricted computational capabilities, they only consider access frequency of data. On the other hand, mLFU also considers the importance of data.

Another important factor which can greatly improve data availability, retrieval robustness and delivery latency at the edge is caching [29]. The performance of an edge caching algorithm is strongly related to the knowledge of content popularity among a number of users. Content popularity is the probability that a specific content item is requested [30]. However, the temporary content popularity changes dynamically over time. Therefore, machine-learning-based techniques are employed, in order to design efficient proactive caching algorithms. However, these techniques are facing new challenges at the edge which are related to data processing, limited computational resources, prohibitively expensive computational learning processes as well as privacy and security concerns. Thus, efficient learning schemes for massive high-dimensional data are of utmost importance, in order to provide accurate prediction of the cached data at the network edge [31]. For example, Zhang et al. [32] propose a learning offloading approach, which is able to decouple the high-complexity of AI-learning tasks and assign them to distribute edge computing resources. Edge-to-edge cooperation is achieved, thus increasing the efficiency of learning due to its high QoS and utilization of idle learning resources. In addition, prefetching mechanisms are employed as an alternative solution to caching through intelligent admission control mechanisms, which are able to identify the correct time frame that data should be prefetched to the edge. When considering the "correct time" network bandwidth and activity as well as the resource availability of the involved edge devices should be taken into consideration [33]. Prefetching is tackled using machine learning and predictive analytics, aiming at having a more concrete prediction model [34], optimizing the off-loading process by preventing bottlenecks and violations on QoS and QoE expectations of

[1] https://min.io/

[2] https://docs.openstack.org/swift/latest/

[3] https://ceph.io/

the platform.

An important research issue that need to explored further is the optimization of small data packets that tend to add considerable overheads to almost all edge storage solutions [35], [36], [37]. It can be achieved with simple statistical tools that reveal relations between the small data packets, allowing us to bundle them together, similar to the shopping carts logic. Furthermore, machine learning algorithms can be utilized in order to predict which packets are more likely to be used at the next time frame. These machine learning mechanisms can take into consideration more complex parameters like user or application profiling.

Moreover, the edge of the network faces some challenges related to its decentralized management and security. By the integration of blockchain and edge computing many benefits can be obtained as stated in [38], such as reliability of the network and distribution of storage and computation over a large number of distributed edge nodes in a secure manner. Nonetheless, edge computing has several security issues related to control, data storage, computation, scalability and network [39]. The migration of services across heterogeneous devices as well as edge servers can be proven risky. In addition, data integrity cannot be guaranteed, as many parts of the data are stored across different locations which would potentially lead to packet loss. Traditional methods for data detection result in heavy storage overheads. Maintaining security and privacy in computational tasks across a large number of computational nodes remains an open challenge. Thus, new solutions are required, capable of adapting to the decentralization, coordination, heterogeneity and mobility of edge computing. On the other hand, blockchain itself faces several challenges related to the massive storage required for transactions, the centralized risk of a large blockchain and throughput constraints. Although blockchain can guarantee a degree of security and privacy, outsourcing services at the edge are prone to transaction loss. In addition, regarding resource management several challenges concerning the adaptation to the dynamic environments and the large-scale optimization for the collaboration of multiple edge servers must be addressed.

IV. 3GPP

3GPP (3rd Generation Partnership Project) is a worldwide consortium of organizations focused on development and maintenance of mobile standards. Since 1998, they have been responsible of GSM, UMTS, LTE and many other specifications. 3GPP organizes its work in periodical releases. Release 15 was the last complete document published. One of the most important topics is 5G End-to-End Network Slicing. This property of the 5G Systems allows to use multiple types of services and apply them to different network requirements (latency, priority, type of users, etc.) at the same time. This feature is important for the interoperability between operators and service providers, that can tailor their communications according to the needs of the networks and limit the resources per slice. Release 16 is complete but still in production due the worldwide difficulties caused by the COVID-19 pandemic. It will go deeper in the 5G system thanks to different use cases and needs born through the Release 15 investigation, applying the mobile standard to all the spectrum of mobility, from the space to the sea. The studies of Release 17 began in January 2020 and are also affected by the pandemic. Release 17 is dedicated to beyond-5G features that are closely intertwined with various networking technologies[40]. It, also, covers highly specific topics and the Extended Reality has found its place in the agenda as in the industry. Their first step is evaluating the performance of XR in terms of power consumption, capacity, mobility and coverage, in partnership with all the top companies of the mobile industry. They have distinguished three main fields of use cases: virtual reality, extended reality and cloud gaming. The specificity of the study is not only in the physical devices and their resources, but also in the spaces where the applications tend to take place. The numerous companies attached to the study, from Facebook to Xiaomi, confirm the high interest of the industry in the possibilities of the Extended Reality.

A. 3GPP MEDIA STREAMING

The focus of mobile networks has long shifted from analogue to digital and from voice to data. The predominant challenge of mobile networks is how to continuously move more data and move it faster than before. The official entrance of 5G to the mobile stage began with Release 15 and included many advances and innovative designs to achieve bigger bandwidths, lower latencies and more connected devices than ever before. The breadth and ambition of 5G is enormous - from the re-architecting of the core as a Service Based Architecture, to Multi-Access Edge computing, from Networking Slicing to a huge expansion of the radio access network. Of particular interest is the specification work carried out by 3GPP in the domain of media streaming over mobile networks. XR has often been cited as one of the drivers for 5G requirements and as a highly challenging vertical that demands both bandwidth increases and latency decreases. The 3GPP work focuses on streaming media to a user who has no interactive role in composing that media. Its model centres on a one-way download and thus envisages frame buffering (on the client and potentially at the edge) as a core component. There are, however, a number of interesting characteristics and design attributes in the 3GPP architecture which are presented below.

1) Control and Data Separation

Throughout the 3GPP 5G specifications from the radio access network to the core, a clear separation between the control and data planes is observed which is again evident in the 5G Media Streaming Architecture. In Figure 5 (adapted from [4]), a high-level view of the 3GPP architecture for downlink media streaming is presented. For informational purposes, the typical deployment sites for the various components is utilized. The 5GMS client typically runs on the user device and is used by third party applications running on the device to interact with the operator media streaming infrastructure.

[4] Qualcomm Technologies, Inc, VR and AR pushing connectivity limits. October 2018, https://www.qualcomm.com/media/documents/files/vr-and-ar-pushing-connectivity-limits.pdf

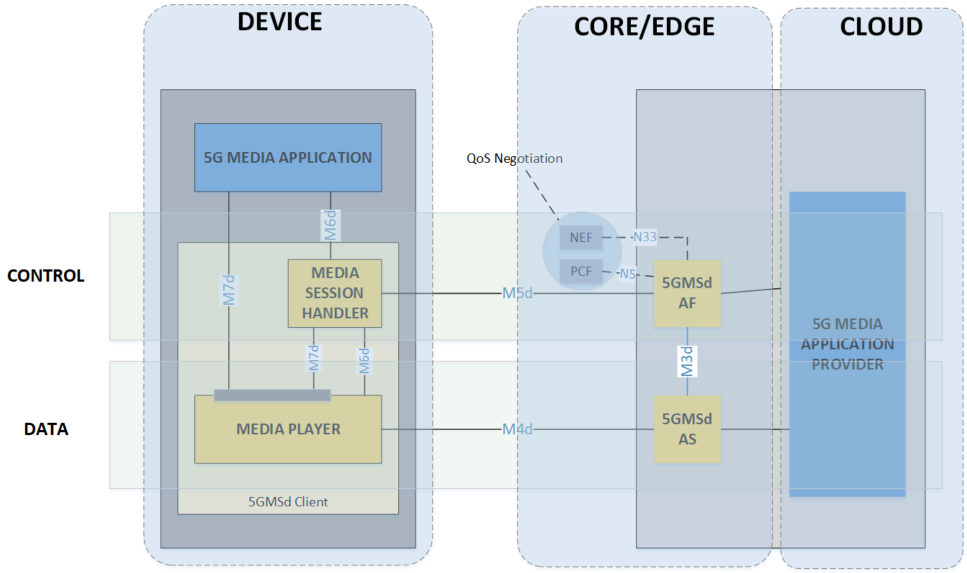

Fig. 5. Control and Data Plane separation in 5G Media Streaming

The Media Application Provider typically installs their server side application on a public cloud and connects it to the 5G data and control planes through standardised APIs to stream their media to 5GMSd-Aware applications. The Media Server takes responsibility of the data plane – decoding, decryption, presentation, monitoring – while the Control Plane is managed by the Session Handler whose purpose is to establish, control and support the delivery of a media session.

2) *QoE Support and Monitoring*

Depending on the level of trust agreement with the operator, server side applications will have access (through Network Exposure Function -NEF- of 5G Core) to functions within the operator network to assist in QoE targets. Release 16 witnessed the introduction of Downlink Network Assistance to the media streaming architecture. This enables a user device that is receiving a downlink media stream to improve the session QoE. As well as clients being able to measure the downlink traffic rate, they can now ask the network what the most appropriate bit rate currently is or what it will most likely be for in the next nominal period. In addition, if a client is running out of buffered content, then it can request a temporary delivery boost from the network. This feature may also be used at the beginning of a session for a faster bootstrap experience. The instrumentation, gathering and reporting of metrics and consumption figures is an area that 3GPP invested some considerable attention in and the design they arrived at is quite powerful. While the media server component is heavily instrumented to be able to gather useful metrics, it does so only at the behest of the session handling infrastructure. Metrics configuration is done on the network level, for instance defining which geographical areas that shall have metrics collection active, which metrics to collect, and how metrics shall be reported. Metrics are periodically reported to AF which periodically reports to the central server - possibly after carrying out some aggregation and filtering.

B. *3GPP Edge Computing*

Section III has already presented a general introduction to edge computing and discussed the reference architecture of Multi-access edge computing (MEC), a standardization initiative by ETSI ISG. In this section, we will discuss the relevance of edge computing to 3GPP (Third Generation Partnership Project). Similar to ETSI ISG MEC work, 5G networks based on 3GPP specifications leverage the network function interactions to align network virtualisation and SDN paradigms. In fact, ETSI has published a white paper [41] on how MEC can benefit the 3GPP ecosystem to enable application and service delivery at the edge of mobile networks. Figure 6 depicts the integrated MEC deployment by a 5G network. The MEC system comprises of MEC hosts and a MEC orchestrator, i.e., centralized system-level functional entity that can interact with Network Exposure Function or directly with a 5G NF. The distributed host functions are expected to interact with the 5G NF and are often deployed in the data network of the 5G system. MEC is deployed on the N6 reference point (as shown in Figure 6), in a data network external to the 5G system.

The distributed MEC host is intended to accommodate MEC apps, a message broker and a service to steer traffic to local accelerators. The ETSI white paper mentioned above also describes various deployment scenarios where MEC hosts can be deployed in the edge or centrally and the UPF takes care of traffic steering to MEC applications. The four deployment scenarios include – (i) MEC and the UPF collocated with the Base Station, (ii) MEC collocated with a transmission node, possibly with a local UPF, (iii) MEC and the local UPF collocated with a network aggregation point, (iv) MEC collocated with the Core Network functions (in the same data center), thus MECs can be flexibly deployed across any network components, from edge to a central data network.

As edge can be a huge benefit to the 5G, especially for latency sensitive applications, 3GPP has introduced edge

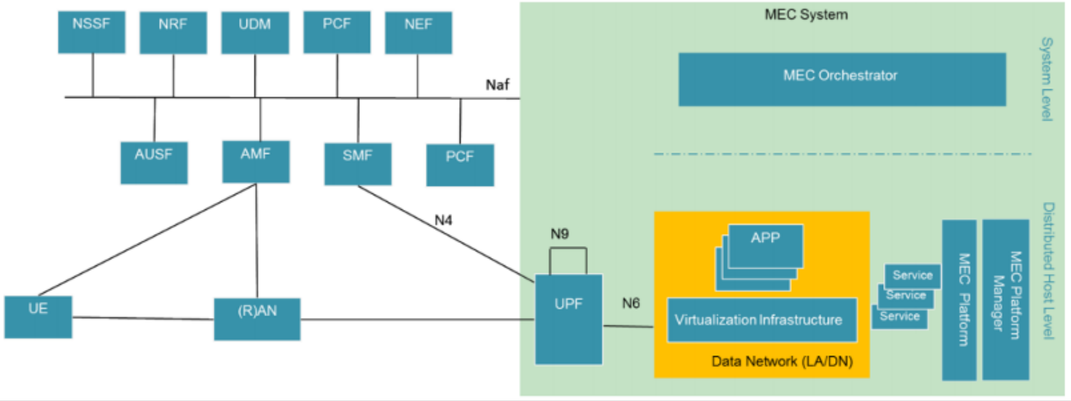

Fig. 6. MEC deployment in 5G Network

computing as a priority area in Release-17[5] aiming to provide native support for edge computing in 3GPP networks. The efforts cover application layer architecture, mobile core network enhancement, security aspects of edge computing in 5G core, multimedia streaming extensions and management of Edge computing ecosystem. Multiple stakeholders are involved in such edge deployments including:

- Edge computing service provider (ECSP) – to build the infrastructure
- Mobile Network Operator (MNO) – to provide access to the infrastructure
- Application Service Providers (ASP) – to host application on the edge infrastructure

The architecture for enabling edge services has been presented in [41] (shown in Figure 7). The main objective of the work is to enable the communications between the applications running on the mobile device (Application Clients) and Edge servers (EAS) located at the network edge. The communication includes both control (service provisioning, edge discovery, mobility management between EAS) and data signals. The architectural principle includes (i) portability: Application logics of AC and EAS should not be modified while reaching the edge hosting services, (ii) differentiation: MNO should be able to decide service differentiation, and (iii) flexible deployments. The architecture also paves the way to internetwork with existing 3GPP networks using existing capability functions such as NEF and PCF.

Thus, 3GPP's native support for edge computing aims to offer several capabilities which include on-demand service provisioning, better availability of edge resources and resilience, network capability exposure via north bound API and support for edge/cloud continuum. The alignment of this architecture with ETSI to have a synergised Mobile Edge Cloud architecture has also been proposed [42]. Several open-source implementations are available to enable the deployment of edge computing solutions. The main ones include Central Office Re-architected as a Datacenter (CORD)[6], Low Latency Multi-access Edge Computing Platform for Software-Defined Mobile Network (LL-MEC)[7] and LightEdge [43].

V. Networking

The fact that XR applications need to be able to operate in real time in order to provide an immersive experience to the end users, makes them extremely latency-sensitive. Furthermore, Holography-based applications in particular, require huge amounts of bandwidth to be allocated. To that end, the re-examination of certain aspects of the Internet data plane, the involved transfer protocols and the subsequent architecture is required. It is of paramount importance to introduce network mechanisms that are able to accommodate the various latency-sensitive and bandwidth-intensive XR applications. Despite the advances in 5G technologies and 3GPP Release 17 that aims to greatly enhance traffic steering and policy provisioning, it is still quite challenging to guarantee bounded end-to-end latency.

A. *Time-sensitive & Deterministic Networking*

Deterministic services were introduced to solve this issue. Deterministic services are an additional QoS feature that is provided by Best-Effort networks[44]. Deterministic QoS features are essential in order to facilitate flows that are critical to real-time applications. The latency-sensitive nature of XR applications can be expressed in two ways. The first

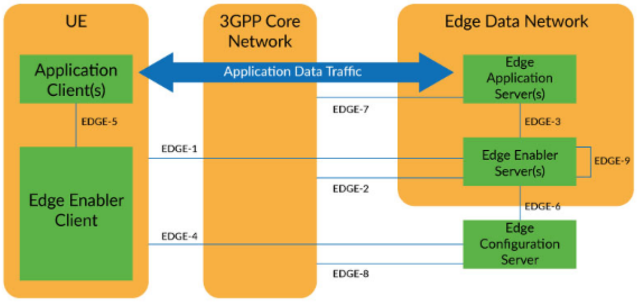

Fig. 7. Enabling applications at the edge on 3GPP networks.

[5] https://www.3gpp.org/release-17
[6] https://opennetworking.org/cord/
[7] https://mosaic5g.io/ll-mec

one is the need to facilitate the temporal correlations that may be formed among specific traffic flows. In order to do so, it is essential to implement time synchronization via the use of a synchronized clock that serves as a common time reference. Time-sensitive Networking (TSN) can provide this type of functionality. Furthermore, Time-sensitive networking is able to leverage time synchronization in order to conduct traffic steering and scheduling in a manner that leverages time cycles. The Deterministic Networking[8] Working Group was established by the IETF, in order to expand Time Sensitive Networking to incorporate routers. The inclusion of routing devices enables the various Time Sensitive techniques to be applied in large scale networks. Thus, the isolated TSN networks can be connected via the use of Ethernet Bridges in a manner that guarantees bounded end-to-end latency. Deterministic Networking (DetNet) is able to establish deterministic data paths that enable low latency, loss and jitter. The incorporation of cyclic operations in queue and dequeue functionalities was suggested by the IEEE in 802.1Qch[9] (Cyclic Queuing and Forwarding). Cyclic Queuing and Forwarding (CQF) is based on dividing time into specified intervals. The queue and dequeue processes require two queues for each class. CQF has the drawback that the propagation latency has to be less than the specified time cycle. As one can see, CQF is not an appropriate solution in the context of large scale networks. In [45], Cycle Specified Queuing and Forwarding is proposed in order to tackle this issue. CSQF is similar to CQF except for the incorporation of explicit descriptions of the transmission cycles at every DetNet node that is included in the path.

Another way of defining the latency-sensitive nature of XR applications is based on their need for bounded low end-to-end latency. By classifying the various data flows, it is possible to establish dedicated data paths for the time-sensitive traffic flows. This method enables the distribution of the available transmission mediums between DetNet and non-DetNet flows in an optimal manner. DetNet-enabled devices are designed to incorporate ports, each of which is equipped with queues that are utilized by DiffServ and Best-Effort traffic. Each DetNet-enabled device can be configured to utilize per-class or per-flow queuing. The reservation of specific network assets for latency-sensitive services, enables the establishment of network-layer certainty in terms of information.

One way of optimally leveraging the available bandwidth is via the use of Multipath Routing. Multipath Routing is the simultaneous use of multiple paths for the purpose of transmitting streams of data flows. Each path corresponds to a specific stream. This process enables the creation of multiple transmission queues and consequently achieves better utilization of the available bandwidth. In case the number of streams exceeds the number of available paths, some of them are chosen to share the available paths. On top of better transmission performance, Multipath Routing introduces advanced fault tolerance by providing alternative paths to the streams, in case the designated one fails.

Multipath Routing can be implemented by leveraging routing strategies that operate in combination with already established protocols. Equal-Cost Multipath Routing is a well known Multipath Routing strategy. Its cornerstone is the use of hash functions to distribute the traffic among various equal cost paths. Its main drawback is that it does not consider the events that are bound to affect the status of the network. This rather unfortunate inability leads to sub-optimal load balancing. In[10], the Internet Engineering Task Force introduces Multipath TCP. Multipath TCP is a set of extensions to standard TCP. These extensions enable transport connections to take place by leveraging multiple paths simultaneously. However, all the solutions explored to this point fail to provide low upper bound of end-to-end latency. In [46], Latency-controlled End-to-End Aggregation Protocol (LEAP) is introduced. LEAP is a multipath transport layer protocol that provides probabilistic end-to-end performance guarantees thanks to path multiplexing and inter-flow coding. The utilization of packet-level encoding enables the information of each data flow to be carried through all the available paths. The information is then retrieved at the destination. In [47], a rather similar approach is explored in the context of UDP for video streaming.

B. OpenRAN

The evolution of cellular networks, especially, in the last two decades has facilitated better and seamless connectivity to the Internet. Various technological innovations and milestones have been achieved over various generations of cellular access and the most recent fifth generation of cellular networks has not just seen upgrades in terms of network availability and speed but also in terms of openness, transparency and security. Open radio access network is one such initiative to move towards open radio networks from a conventional closed network, by supporting interoperation between networking devices from various vendors. The three primary elements in the RAN includes (i) Radio Unit (RU), where the radio signals are processed, (ii) Distributed Unit (DU) comprising of real-time baseband functions, and (iii) Centralized Unit (CU) where less time-sensitive packet processing functions reside. The OpenRAN defines standard interfaces to communicate between these RAN elements breaking the siloed nature of traditional RAN, and at the same time, to bring down the costs of deployment for network operators. Moreover, open RAN standards are developed using vRAN principles, offering security, flexibility at a diminished cost. 3GPP plays a major role in facilitating open RAN, especially for 5G, through splitting the gNB (5G radio base station) into a CU and DU, as shown in Figure 8. The CU can be implemented as a Virtual Network Function and DU as a Physical Network Function, with the communication between CU and DU [48].

Several industrial initiatives are extending the standards beyond the 3GPP specifications mainly due to the expansion in industrial IoT devices. More specifically, 3GPP Release 17 describes the implementation of unlicensed spectrum, massive Multiple-Input and Multiple-Output (MIMO) and a lightweight communication for industrial settings. In what it

[8] https://datatracker.ietf.org/wg/detnet/about/

[9] https://1.ieee802.org/tsn/802-1qch/

[10] https://datatracker.ietf.org/wg/mptcp/documents/

follows, the three main such industrial alliances that facilitate to OpenRAN are presented.

1) O-RAN Alliance

O-RAN alliance had been setup to define requirements and help build a supply chain ecosystem to realize openness and intelligence for evolving radio access networks. Openness allows RAN implementations to scale and enable smaller vendors/operators to introduce customized services, building a competitive ecosystem. On the other hand, intelligence would help in automating the deployments and operating the network in an automated fashion. O-RAN alliance was formed by operators and later became a community of vendors, and research & academic institutions. A key principle of the O-RAN architecture is to extend SDN concept of decoupling the Control Plane (CP) from the User Plane (UP) into RAN alongside embedded intelligence. This extends the CP/UP split of Centralized Unit being developed within 3GPP through the E1 interface (Figure 8), and further enhances the traditional radio resource management functions (assigning, reassigning, and release radio resources in single/multi-cell environments) with embedded intelligence.

The intelligence is introduced through the hierarchical (Non-Real Time -RT- and Near-RT) RAN Intelligent Controller (RIC) with the A1 and E2 interfaces (Figure 8). The Non-RT function (i.e., >1s) include service and policy management, RAN analytics and model training. The trained models and real-time control functions are distributed to the RIC for (near) real-time executions. Ninkam et al. [49] show the benefits of implementing Open RAN through specifications from O-RAN alliance through evaluation of real-world cellular data. The work also points the challenges and open problems in the area.

2) OpenRAN

OpenRAN project group was announced by TIP (Telecom Infra Project), a consortium of hundreds of service and technology providers, to enable open architecture design and flexible deployment of RAN equipment. OpenRAN aims in building disaggregated RAN functionality through open interface specifications, whereas O-RAN alliance is a specification group defining next generation RAN infrastructures. The key philosophy behind OpenRAN includes disaggregation of RAN HW & SW on vendor neutral platforms, open interfaces to enable interoperability within vendors, flexibility in terms of choosing the cellular-generation, hardware agnostic deployments and innovation via adopting smartness – all intended to take a holistic approach towards building next-generation RAN. OpenRAN architecture has been split into four components and two segment subgroups. The component subgroups look into the open/disaggregated hardware implementations of Radio Unit and Distributed/Centralized Unit and as well on the Radio Intelligence and OpenRAN orchestration and life-cycle management. The segment subgroups investigate the outdoor macro deployments and indoor small-cell deployments.

The RAN architecture is split into two parts: (i) the lower layer RAN split between the antenna integrated Radio Unit (RU) and the Distributed Unit (DU) amd (ii) the higher layer RAN split, a 3GPP standard F1 interface between the DU and the Centralized Unit (CU). The F1 interface is expected to comply with 3GPP F1 interface specifications (Figure 9). The management system of the OpenRAN provides interfaces that comply with 3GPP Integration Reference Points (IRP) specifications. The system is expected to provide management, configuration, monitoring, optimization and troubleshooting capabilities. The OpenRAN and O-RAN alliance have also signed a liaison agreement to ensure alignment in developing interoperable RAN solutions[11].

3) Open RAN policy Coalition

Open RAN policy Coalition (ORPC) is a more recent group aimed to advocate for government policies to support the development and adoption of open radio access network. In a sense, the coalition aims to bring a policy-perspective to complement the standardization and technical implementations done by O-RAN and OpenRAN communities. Overall, the goals of ORPC include supporting the O-RAN and OpenRAN communities, calling the government to support for open and interoperable solutions, creating policies that support vendor-diversity, and enabling funding and promoting open radio deployments[12]. Furthermore, operators and vendors have published their views and commitments to open RAN through these alliances (Nokia[13], Telefonica[14], NTT Docomo[15], Ericsson[16]). Several research work and frameworks to enrich the capabilities of RAN have also been made open, following the CU/DU split specifications from the alliances. Some of them include O-RAN[17], COMAC[18] and SD-RAN[19].

VI. XR-Relevant Standards

A. MPEG Mixed & Augmented Reality

The MAR Standard [50] (Mixed and Augmented Reality) is a reference model (MAR-RM) established to define required modules, minimal functionalities and the associated information content and models for applications, components, systems, services that must claim compliance with MAR systems. This reference model, published as ISO/IEC 18039 [51], defines the scope and key concepts of MAR including the relevant terms and their definitions and a generalized system architecture. The MAR-RM is agnostic to platforms, used devices and algorithms. It does not specify how MAR applications should be designed, developed and implemented. The objective of the MAR reference model is to establish the definitions, main concepts and overview of the architecture needed to create mixed and augmented reality systems or applications. The reference model is planned for use by current and future engineers of MAR applications, parts, frameworks, administrations, or particulars to depict, analyse, contrast, and

[11] https://techblog.comsoc.org/2020/02/26/tip-openran-and-o-ran-alliance-liaison-and-collaboration-for-open-radio-access-networks/

[12] https://www.openranpolicy.org/wp-content/uploads/2021/06/ORPC-Open-RAN-NOI-Reply-Comment-Letter-as-filed-May-28-2021-c3.pdf

[13] https://www.nokia.com/networks/radio-access-networks/open-ran/

[14] https://www.telefonica.com/es/wp-content/uploads/sites/4/2021/08/Whitepaper-OpenRAN-Telefonica.pdf

[15] https://www.nttdocomo.co.jp/binary/pdf/corporate/technology/whitepaper_5g_open_ran/OREC_WP.pdf

[16] https://www.ericsson.com/en/openness-innovation/open-ran-explained

[17] https://wiki.o-ran-sc.org/pages/viewpage.action?pageId=20876303

[18] https://opennetworking.org/comac/

[19] https://opennetworking.org/sd-ran/

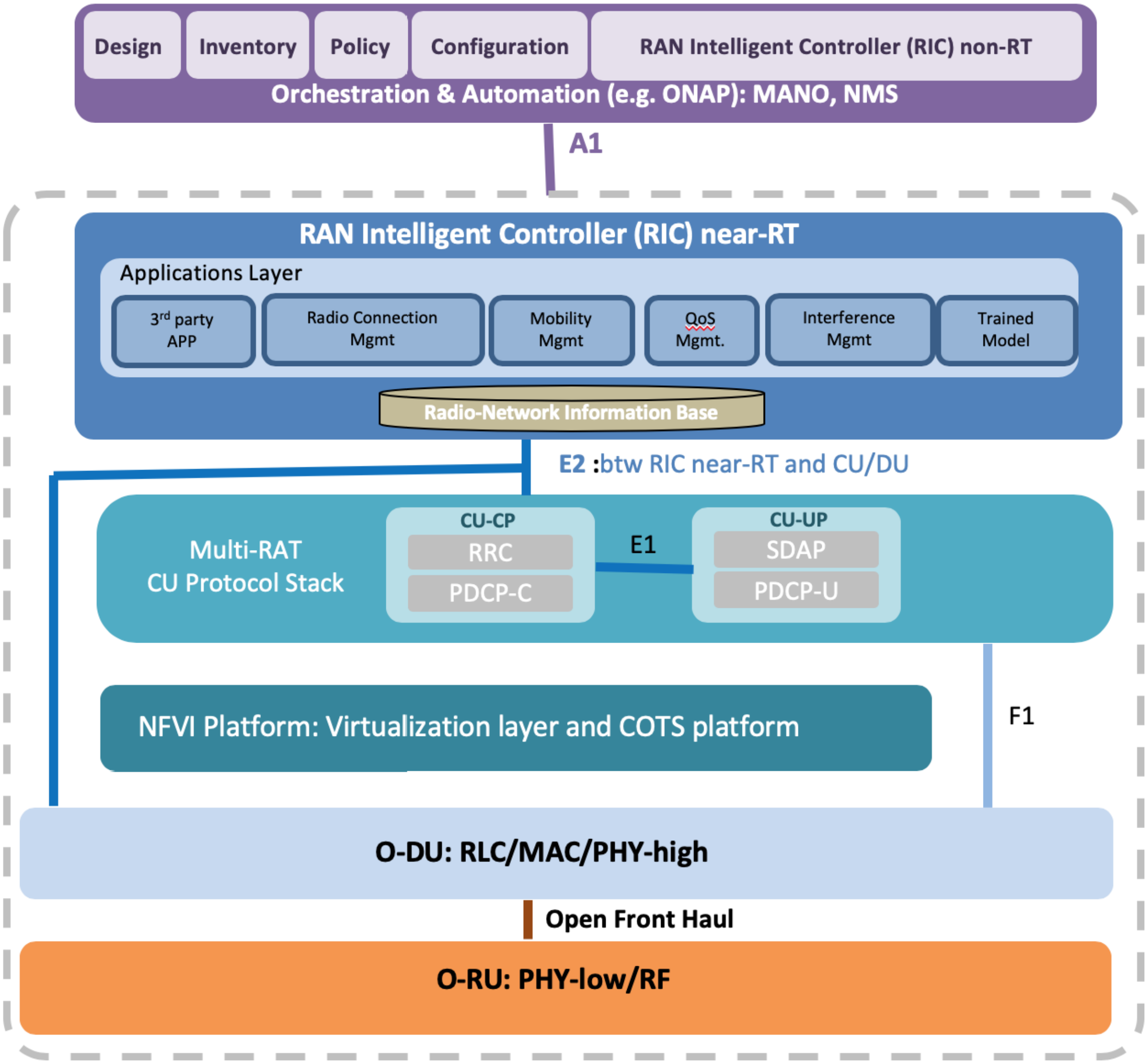
Design
Inventory
Policy
Configuration
RAN Intelligent Controller (RIC) non-RT
Orchestration & Automation (e.g. ONAP): MANO, NMS
A1
RAN Intelligent Controller (RIC) near-RT
Applications Layer
3rd party APP
Radio Connection Mgmt
Mobility Mgmt
QoS Mgmt.
Interference Mgmt
Trained Model
Radio-Network Information Base
E2 :btw RIC near-RT and CU/DU
Multi-RAT CU Protocol Stack
CU-CP
RRC
PDCP-C
E1
CU-UP
SDAP
PDCP-U
NFVI Platform: Virtualization layer and COTS platform
F1
O-DU: RLC/MAC/PHY-high
Open Front Haul
O-RU: PHY-low/RF

Fig. 8. Reference architecture of O-RAN alliance

impart their compositional plan and execution. The sensed information from the physical world is delivered in the MAR execution engine either directly or through the context analyzer. The MAR execution engine may also access media assets and required external services. The engine processes the sensed information, outputs the processing outcome, and manages user interactions with the system. The Enterprise viewpoint verbalizes the perspective of the business elements in the framework that ought to be justifiable by all partners. This intentionally focuses on scope and policies while it also presents the targets of various actors involved. It defines the actors involved in a MARS (Mixed and Augmented Reality System), the associated potential business models and the desirable characteristics at both ends of the value chain. The actors may be categorized in four classes:

- Providers of Authoring/Publishing: MAR Authoring Tools Creator, MAR experience creator, and Content Creator
- Providers of MAR execution engine components: Device manufacturer and Device middleware/component Provider
- Service Providers: MAR Service Provider, Content Aggregator, Telecommunication Operator, and Service Middleware/Component Provider

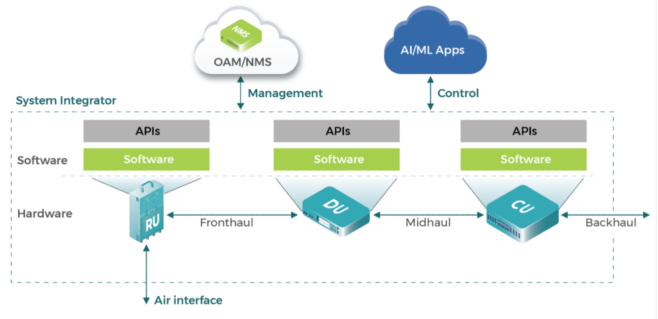

Fig. 9. Reference architecture for TIP-OpenRAN

- MAR User: MAR Consumer/End-User Profile

The Computational viewpoint identifies the main processing components, both hardware and software, and define their roles and interconnectivity [51]. Below, descriptions for the different components of reference system architecture are provided.

- Sensor: A sensor is a device used to detect, recognize, and track the target physical object to be augmented. It captures properties-measurements of the physical world and interprets and converts them into digital signals. Observed data may be utilized by tracker/recognizer to evaluate the context and/or to compose the scene. There is a variety of types of devices (cameras, environmental sensors, etc.) that measure different physical properties
- Context analyser-Recognizer: is a system that analyses signals sensed from the physical world by conducting comparisons with local or remote target signal (i.e., target for augmentation). It produces MAR events and data
- Context analyser-Tracker: as part of the context analyser, aims to detect and measure changes in the target signals physical properties (e.g., pose, orientation, volume, etc.)
- Execution engine: Its main objective is to further recognize and track the target to be augmented by interpreting the sensed data. It imports the object data from the physical world. It also creates computational simulations of the dynamic behaviour of the augmented world. Finally, it integrates the physical and virtual data before rendering within the required modalities (e.g. visuals, aurals, haptics)
- Simulator-Spatial mapper: as part of the simulator within the execution engine, computes spatial relationships (position, orientation, scale, and unit) between the physical world and the MAR scene by applying the required calibrating transformations. Furthermore, it maps each sensor's spatial reference frames and spatial metrics
- Simulator-Event mapper: as part of the simulator within the execution engine, is responsible of associating MAR events, obtained from the Recognizer or the Tracker, with conditions specified by the MAR Content creator in the MAR scene
- Renderer: as the part of the execution engine, produces the output signal for the presentation of the MAR scene simulation. It is responsible of converting the MAR scene into the proper form of output signal for the given display device
- Display-UI: The display device presents actual MAR scene to the end-user in various modalities. Displays and UI include monitors, head-mounted displays, projectors, scent diffusers, haptic devices, and sound speakers

B. Omnidirectional MediA Format (OMAF)

Immersive media technologies such as virtual Reality (VR) combine omnidirectional media and head-set displays to provide users with an immersive experience. Omnidirectional media includes videos that have been captured using 360-degree cameras with a field of view that covers approximately the entire sphere in the horizontal plane. Omnidirectional MediA Format (OMAF) is a virtual reality (VR) system standard developed by the Moving Picture Experts Group (MPEG). OMAF enables omnidirectional media applications - 360° video, images, audio and timed text (text media synchronised with other media). OMAF v2[20] fully supports three degrees of freedom (3DOF), while support for six degrees of freedom (6DOF) is still progressing. This will allow for transitional user movement to prompt the rendering of overlays and for multiple viewpoints. Requirements for OMAF v3 include support for new visual volumetric media types, such as video-based point cloud compression (V-PCC) and immersive video. The MPEG standard for visual volumetric video-based coding and V-PCC has been finalized and can be used to represent captured volumetric objects. The MPEG Immersive Video standard was planned for completion in July 2021 and enables 6DOF within a limited viewing volume. It is expected that the OMAF standardization for integrating these media types will start later this year.

Figure 10 presents the OMAF architecture, showing the three main modules; (1) OMAF content authoring module, (2) delivery access module, and (3) the OMAF player module.

- OMAF content authoring module - media acquisition, omnidirectional video/image preprocessing, media encoding, and media file and segment encapsulation
- OMAF delivery access module - may either use file delivery or streaming delivery for which the content is timewise partitioned into segments
- OMAF player module - media file and segment decapsulation, media decoding, and media rendering

The key underlying technologies for file/segment encapsulation and delivery of OMAF are ISO Base Media File Format (ISOBMFF) and Dynamic Adaptive Streaming over HTTP (DASH). OMAF specifies the following:

- A coordinate system that consists of a unit sphere and three coordinate axes; x (back-to-front), y (side-to-side) and z (up)
- A projection and rectangular region-wise packing method used for conversion of a spherical video sequence into a two dimensional rectangular video sequence. The spherical signal is achieved by stitching video signals captured by multiple cameras
- Storage of omnidirectional media and the associated metadata using ISOBMFF
- Encapsulation, signalling, and streaming of omnidirectional media in MPEG-DASH (Dynamic Adaptive Streaming over HTTP) and MMT (MPEG media transport)
- Media profiles and presentation profiles that interoperability and conformity points for media codecs as well as media coding and encapsulation configurations that may be used for compression, streaming and playback of the omnidirectional content

It is widely agreed that Edge Computing provides more scope for the development of new service platforms. In parallel to this, container-based deployments in the cloud work to

[20] https://github.com/nokiatech/omaf

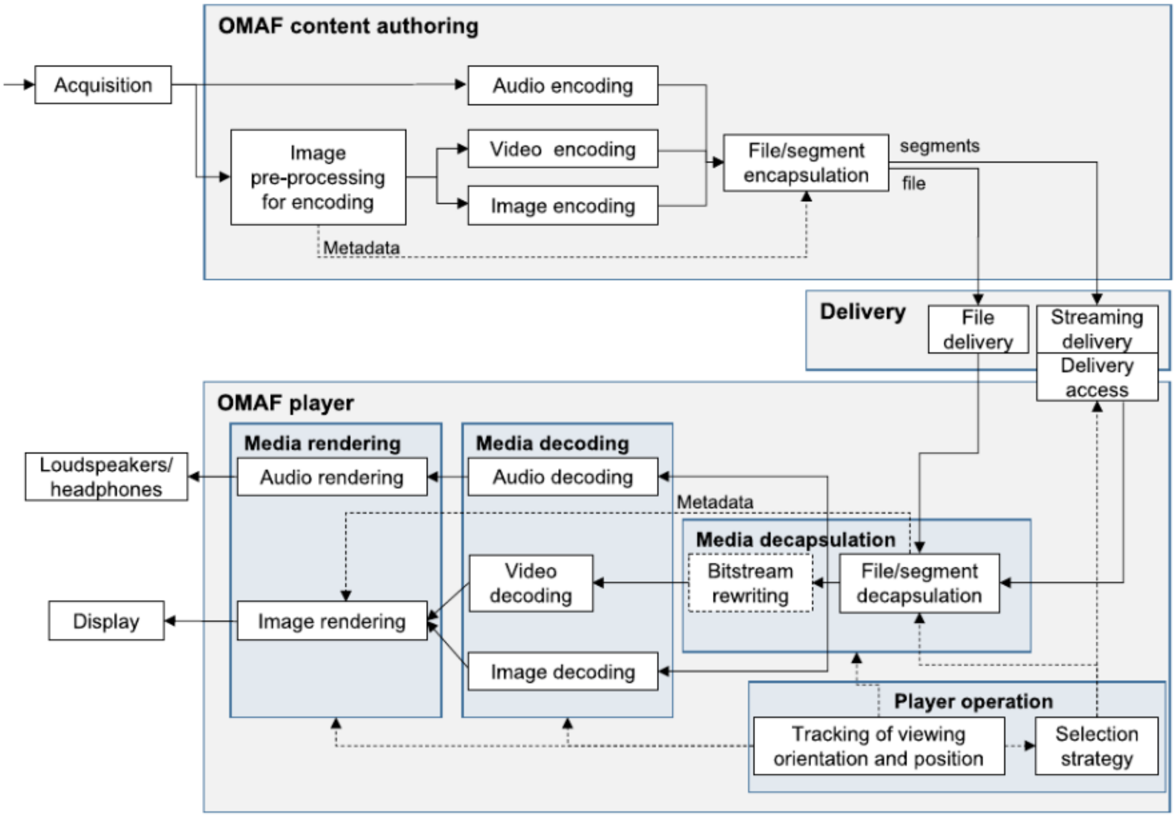

Fig. 10. Reference architecture for OMAF

simplify things further, especially when combined with software defined networking (SDN). In this regard, easily scalable computing capacity is a requirement for such architectures. Furthermore, hosting services closer to consumers at the network edge should greatly reduce latency and bandwidth requirements for real-time omni-directional media processing. The Network-based Media Processing (NBMP) which is a complementary standard (ISO/IEC 23090 Part 8)[21], works to facilitate this by defining interfaces, media formats, and metadata to provide a standardised way to perform media processing on any Edge and Cloud computing architectures. NBMP can also be used to reduce network redundancy at delivery by implementing conversion on-the-fly in the network.

As shown in Figure 11, users describe the media processing operations to be performed by the media processing entities in a network. A workflow is described by composing a set of media processing functions accessible via the NBMP APIs. The Media Processing Entity (MPE) then runs tasks to process the media data and metadata received from the media source (or other processing tasks), to be consumed by a media sink (or other processing task).

C. Supporting standards by Khronos group

1) OpenXR

OpenXR[22] is an open standard by Khronos group to connect XR devices and game engines with each other. The goal is to eliminate the various specific implementations to support an XR-device, X in game engine Y on platform Z by providing a generic, cross platform application interface to any supported XR device within any supported game engine on multiple platforms. Special sensors and interfaces for hand tracking and eye-tracking are also supported. Even cloud-based solutions via 5G have already been discussed. Currently version 1.0 has

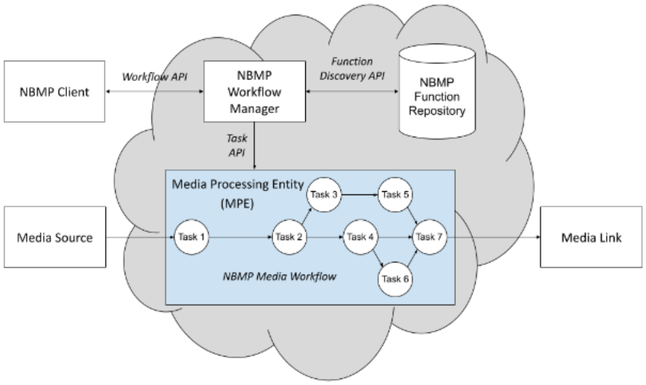

Fig. 11. The workflow of Network-based media processing

[21] https://mpeg.chiariglione.org/standards/mpeg-i/network-based-media-processing

[22] https://www.khronos.org/openxr

been released. A lot of well known companies are members of the consortium. To adapt this standard for a specific end user device like a holographic 3D display, a specific implementation in OpenXR is mandatory to be developed first, so that the various game engines and OpenXR core libraries can access and control the holographic 3D end user device. 3D content in the supported game engines must be appropriately generated to be compatible with holographic 3D. On the other hand, some well-known 2D stereo based HMDs are already supported.

2) glTF

glTF™[23] (GL Transmission Format) is a royalty-free specification for the efficient transmission and loading of 3D scenes and models by engines and applications. glTF minimizes the size of 3D assets, and the runtime processing needed to unpack and use them. glTF defines an extensible, publishing format that streamlines authoring workflows and interactive services by enabling the interoperable use of 3D content across the industry.

D. DRACO

3D graphics are an indispensable part of many applications, including gaming, design and data visualization. Year by year, graphics processors, rendering engines and creation tools improve, resulting in larger and more complex 3D models. Such high-fidelity models have become the industry standard and help fuel new applications in immersive virtual reality (VR) and augmented reality (AR). However, the increased model complexity forces an increase of storage and bandwidth requirements to keep up with the explosion of 3D data. To deal with this rising problem, multiple compression algorithms are being designed and implemented. One of the current state-of-the-art industry-ready methods is Draco, an opensource compression library designed by Google's Chrome Media team[24]. Draco aims to improve the storage and transmission of 3D graphics by compressing meshes and point-cloud data. The huge impact of Draco is depicted in current bibliography, as it influences and drives the design of alternative compression/decompression techniques [52], [53], [54], [55], [56], [57], [58], [59], [60], [61]. The framework uses a kd-tree to efficiently store data corresponding to points, connectivity information, texture coordinates, color information, normals and any other generic attributes associated with geometry. Therefore, it can be used to compress and decompress geometric meshes as well as point clouds, making it ideal for AR and VR applications. Draco compress *.obj models into *.drc equivalent models and vice versa. Despite the high-compression rate (e.g., 96.7% for the Stanford Dragon and 72.6% for the Stanford Bunny), a model that is compressed and then decompressed preserves the high-fidelity of the original model. The running times of these processes depend on whether C++ or Javascript encoders/decoders are used (e.g., decoding using C++ is significantly faster).

E. ETSI Augmented Reality Framework

ETSI, as part of its standardization efforts, is currently specifying an Augmented Reality Framework (ARF) whose objective is to provide a transparent architecture for interoperation in the highly heterogeneous ecosystem of providers and technologies that stimulate developers. ARF building blocks were defined across three layers as depicted in Figure 12: Hardware, Software and Data Layers [62]. The Hardware Layer comprises the Tracking Sensors (e.g., for positioning and orientation), the Processing Units (i.e., dedicated embedded processing components such as GPUs), the Rendering Interfaces (i.e., the components on which the AR content is rendered) and the Interaction Interfaces used to interact with the system. The Software Layer encompasses the Vision Engine, leveraging the output of tracking sensors and processing units to understand the real-world environment, and the 3D Rendering Engine used to maintain a 3D view from the world. Finally, the Data Layer is divided into a World Knowledge component, which maintains a digital representation of the real world, and the Interactive Contents (i.e., the representation of virtual elements).

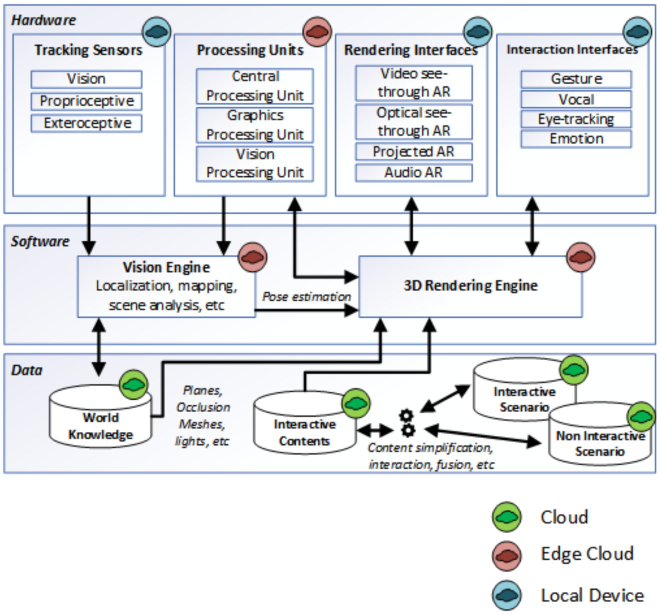

Fig. 12. ETSI Augmented Reality Framework (ARF), adapted from [62]

ISG AR group specified a reference functional architecture for both fully embedded AR systems and implementations spread over IP networks (e.g. edge/cloud environments). The AR functional architecture was organized into ten high-level functions.

- World Capture (hardware layer): analysis of the environment using different types of sensors (e.g., RGB-D cameras, event-based cameras) to provide different streaming types (e.g., audio, video, events) to be consumed by additional components and sub-functions. It collects position and orientation and records the space and sounds
- World Analysis (software layer): processing of the information captured by the sensors. It defines how the

[23] https://www.khronos.org/gltf/
[24] https://google.github.io/draco

AR system shall deliver functions such as Object Recognition and Identification (e.g., using Machine Learning techniques) or Object Tracking capabilities (position and orientation estimation) based on previously captured data. All these parameters allow the creation of the 3D representation

- World Storage (data layer): reception and delivery of data used by other functions. It keeps an updated representation of the real world that can be shared between different devices. Also, World Storage can extract the information needed from the representation to update in real-time without bandwidth waste. This allows updating the representation faster only using the data that changes when the device is relocating
- Asset Preparation (data layer): it provides multimedia objects to add to the AR scene and allows to interact with real objects in the scene. Object behaviour is the sub-function that stores predefined behaviours, based on AI algorithms
- External Application Support (hardware layer): real-time communication and data exchange
- AR Authoring (data layer): information and format optimization before sending it to the rendering functions. It also adds objects to the scene and their behaviour
- User Interactions (hardware layer): it delivers to the system information from other types of interaction devices, as tactile surfaces, biometric sensors, etc
- Scene Management (Software Layer): it is the core function, maintaining scenes at run-time. It receives optimized information from AR Authoring and interactions captured by sensors and answers with the new interactions that must be shown in the scene
- 3D Rendering (Software Layer): real-time rendering of the scene with the information provided by Scene Management. It generates audio, video and haptic responses of the interactive objects
- Rendering Adaptation (Hardware Layer): it updates the scene with the new rendered information

Transmission related sub-functions such as Security (i.e., the need for encrypted communications and access control mechanisms), Communication, including the encoding/decoding schemes to cope with large representation formats (e.g., 3D Point Cloud) or even Service Conditions are briefly discussed. Namely, the Service Conditions sub-function, already comprises the idea that service-related parameters such as network conditions should be taken into consideration to adapt the behaviour of the AR system and be used to optimize the overall Quality of Experience (QoE). Moreover, the presented AR functional framework does also acknowledge the manifold benefits of real-time connectivity with external systems. Nevertheless, the details of such communications are left outside of the ETSI AR framework specification.

In the Hardware Layer, most of the components are local. Nevertheless, the dedicated Processing Units were conceptually pushed to a low latency edge cloud. Likewise, the engines present in the Software Layer were also designed as part of the edge cloud. Whereas the remaining components, at the data layer, were designed as part of the cloud. ETSI specifies relevant components and interfaces required for an AR solution through a set of six use cases centred in industry, whose objective is reducing costs using virtual prototypes and decreasing timeframes of procedures:

- Try before buying with AR, an app to visualize furniture in a certain space
- Maintenance Support, a tool that improves communication between engineers and technicians while updating circuits in street cabinets
- Manufacturing procedure, an AR-based appliance for workstations with virtual tools to guide the user through manufacturing steps
- Collaborative design review, an app with two headsets that allows seeing the same parts of the product in review
- Factory inspection based on an ARCloud, it detects sensors located in a plant and shows the information to the inspector through a headset
- Usability Evaluation of Virtual Prototypes, to test early versions before production by observing the virtual interaction that a potential user does

Each use case contains a description of how their functional steps map into the proposed AR architecture. For some of them, again, it was stressed the benefits of having different functions running on local, edge or cloud, which ultimately allows better resource exploitation. Furthermore, such specification was mainly focused on AR technology and scenarios, yet it could also be interesting to see how this work compares with the Virtual Reality scenarios or whether a unified approach for Mixed Reality Scenarios can be devised.

F. 3D Point cloud data formats

Convenient ways to provide visual information are based on image- or video data. This is due to the fact that 2D displays or 2D projection systems are a common way to present visual information. They use pixelated image-data as input, with a certain update rate we get video information. Various methods exist to compress image or video data. The basis for image compression is on the one side the fact that the human eye is somewhat tolerant against some variation in information. On the other side there is often a lot of redundant information in images and video material. So this type of data can be compressed very efficiently. But with upcoming new display technology in the area of volumetric or holographic displays, the demand for new data formats beyond only flat 2D surfaces comes up. Also, the demand for variable perspectives within content arises - e.g. for sports events recordings. A format providing not only pixels for representation on a 2D surface but providing a volumetric representation based on 3D points to allow various perspectives would be an excellent basis for such scenarios - a 3D point cloud. Another known term for such 3D points is called Voxels.

There are various approaches to implement such a format. From the view of official standardization, good progress was made by the MPEG Point Cloud Compression project. It was initiated in 2014. A call for proposals in 2017 resulted in a first draft of the standard by the end of 2018. Until today, the standard is under development, there is an actively maintained reference implementation. Basically, the standard proposes two types of 3D point cloud compression - video based (V-PCC ISO/IEC 23090-5) and geometry based (G-PCC ISO/IEC 23090-9) [63].

The V-PCC variant uses classic image based processing (color + depth + occupancy maps). By applying common image based compression methods (HEVC in the reference implementation), quite good compression rates have been achieved. The method is based on projection of the 3D source scene or point cloud on multiple 2D maps from different perspectives. These projections or patches are then mapped into the frame - the "atlas" - to be encoded / decoded by means of video compression. Here multiple maps are generated, attribute maps (can be RGB color but also something else), depth maps (representing the distance from the according perspective) and an occupancy map (representing valid pixels). Within a (lossless encoded) meta data channel, information about how to reconstruct these patches back into the 3D point cloud are provided within the multiplexed data stream. Within the process of generating the patches and atlas, some improvements on the data are done, e.g. detection and removal of duplicate 3D points or improvement of quality especially on the regions between patches (seams). As a result, very good compression rates have been achieved. The MPEG PCC research group defined some reference data sets, where the rates and quality of different algorithm versions and parameter variants could be measured and compared. For example, a scene with 100k points @30fps corresponds to 360 Mbit/s uncompressed data rate. With V-PCC a compression to about 1 MBit/s can be achieved using version TMC2v8.0 while achieving good quality.

The G-PCC variant is based on compressing the 3D points directly one by one. Here the 3D points structure (point locations) is encoded lossless by using an octree approach. Here a cube is dived into 8 cubes, iteratively, from top to bottom, until finally the point level is reached. At each level it is noted if there are some valid points inside the cube (appropriate bit is 1) or not (bit is 0). For one cube, a 8 bit word represents the 8 cubes assigned in the next level of the hierarchy. In contrast, for encoding point attributes (i.e. RGB color), three compression methods have been developed. These methods basically make use of similarities / redundancy between colors down the octree graph. The algorithm also allows for different levels of details - usable e.g. to adapt for variations in available data rate or to adapt for current detail requirement in rendering process. Currently, the algorithm does not use temporal compression approaches to enable lower data rates in situations where the 3D scene does not change much from frame to frame - compared to video compression where this approach is extremely effective. But some work into this direction was supposed for the next version of the standard. For an example scene with 100k points at 10 fps corresponding to 110 MBit/s uncompressed data rate, a compression down to about 24 MBit/s could be achieved with good quality.

G. 3D Mesh generation from 3D Point cloud data

Mesh generation based on the 3D point cloud data is a very complex process requiring agile algorithms and quite a lot of computing power. Here we focus on approaches dealing with the problem of fast and robust reconstruction of shapes and surfaces rather than very high precision and quality mesh generation. In the literature, there exists a number of methods for effective 3D point cloud meshing. Some methods utilize a feature detection process and first extracts from the point cloud a set of sharp features. Then the reconstruction process must be incorporated in order to provide implicit surfaces and generate a mesh approximating the surfaces and extracted edges. Such a mesh provides a trade-off between accuracy and mesh complexity but also a trade-off between speed and accuracy. The whole process should be as robust as possible to noise contained in the 3D point cloud. Alpha shapes is another interesting, well documented and widely used method for mesh regeneration. It was introduced by H. Edelsbrunner et al. in 1983 [64]. As a generalization of a convex hull it is quite an intuitive and easy to understand method. It is based on the gradual removal of those parts of the space surrounding the point cloud that do not contain any points. The method can be used iteratively to increase the accuracy of the resulting mesh. More detailed description of the method can be found in [25]. Ball pivoting by F. Bernardini et al. [65] and Poisson surface reconstruction by M. Kazhdan et al. [66] are other meshing algorithms that need to be mentioned here. They provide different characteristics to the resulting mesh. There exists an open-source library called Open3D that supports rapid development of software that deals with 3D data - among others it supports mesh generation from 3D point cloud data. More about this library can be found in[26].

Unfortunately, the very process of creating a point cloud from the real space around us is extremely complex and is potentially burdened with many imperfections. Obtaining data from the image generated by RGB cameras is usually insufficient for rapid meshing. A much better and more precise method is to use dedicated devices and technologies such as LiDAR. Apple built-in LiDAR technology into their newest mobile devices. They use their own proprietary libraries/APIs to perform fast and robust mesh generation. The documentation of the methods and algorithms they use is unfortunately not publicly available. Nevertheless it is currently the fastest and most precise technology available for end-users. It provides a very effective solution to both problems: generating 3D point cloud data from the real surroundings and mesh generation. The LiDAR Scanner measures the distance to surrounding objects up to 5 meters away, works both indoors and outdoors, and operates at the photon level at nano-second speeds. New depth frameworks in iPadOS combine depth points measured by the LiDAR Scanner, data from both cameras and motion sensors, and is enhanced by Apple proprietary computer vision algorithms for a more detailed understanding of a scene.

[25] Introduction to Alpha Shapes, https://graphics.stanford.edu/courses/cs268-11-spring/handouts/AlphaShapes/as_fisher.pdf

[26] http://www.open3d.org/docs/latest/index.html

VII. Conclusions

In this paper we have discussed the requirements on XR applications and presented a survey of selected technologies that we see as important enablers for the successful deployment of XR applications in beyond-5G mobile communication networks and for making them available to a wide and diverse user base.

The requirements of XR applications on communications networks that will provide these services are quite demanding, if satisfying levels of QoS and QoE are to be achieved, e.g. regarding latency times and transmission speeds that mainly due to the large volume of 3D image data go way beyond the typical requirements of present applications.

Three categories of XR application use cases are considered having a very high potential for market introduction: Real-time Holographic, Immersive Virtual Training, and Mixed Reality Interactive applications. Examples for specific applications within these categories include e.g. holographic concerts or holographic meetings, VR trainings e.g. in medical education or aeronautical operations, and various types of mixed reality gaming. While some of these types of applications already exist today in local or corporate environments, the challenge is to deploy them over beyond-5G mobile networks in an efficient way, making them available to a wide user base, being offered as a service that users can book and start using as easy as it currently is to download an app on their mobile device.

In the survey we presented enabling technologies regarding both, the overall (mobile) communication network, as well as several specific to XR applications.

Regarding the communication network, technological developments that appear to be very beneficial for enabling XR applications include e.g. the ETSI Multi-Access Edge Computing (MEC), Edge Storage, ETSI Management and Orchestration (MANO), ETSI Zero touch network & Service Management (ZSM), Deterministic Networking, etc. These enable lower latency times and higher throughput with a certain degree of QoS, but also facilitate dynamic and automated instantiation of required network resources, and provide means towards their efficient usage.

Regarding XR-service specific technologies various works and activities are of interest, including 3GPP Media Streaming, MPEG's Mixed and Augmented Reality standard, the Omnidirectional MediA Format (OMAF), ETSI's Augmented Reality Framework and works on 3D-Point Clouds, to again mention a few. These provide architectural frameworks, 3D data models and media types, streaming mechanisms, the required compression algorithms etc.

Extended Reality applications stand a good chance to become one of the target use cases in 6G. They have found their place in the agenda of 3GPP Release 17 as well as in the considerations of industry. Given their challenging requirements on communication networks they can also be a suitable benchmark and serve well for KPI validation. Making XR applications usable for a wide user base will require significant progress on enabling technologies both in the network and terminal devices, and forming a standard that consolidates the various options into a workable solution that is open to attract many stakeholders in an XR services market.

Acknowledgment

This work is part of the CHARITY project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101016509.

References

[1] A. Makris, A. Boudi, M. Coppola, L. Cordeiro, M. Corsini, P. Dazzi, F. D. Andilla, Y. González Rozas, M. Kamarianakis, M. Pateraki, T. L. Pham, A. Protopsaltis, A. Raman, A. Romussi, L. Rosa, E. Spatafora, T. Taleb, T. Theodoropoulos, K. Tserpes, E. Zschau, and U. Herzog, "Cloud for holography and augmented reality," in *2021 IEEE 10th International Conference on Cloud Networking (CloudNet)*, 2021, pp. 118–126.

[2] *Network Functions Virtualisation (NFV) Release 4 Management and Orchestration Requirements for service interfaces and object model for OS container management and orchestration specification, ETSI GS NFV-IFA 040 - V4.2.1*, ETSI, 5 2021.

[3] *Network Functions Virtualisation (NFV), Infrastructure Overview, ETSI GS NFV-INF 001 - V1.1.1*, ETSI, 1 2015.

[4] *Network Functions Virtualisation (NFV) Architectural Framework, ETSI GS NFV 002 - V1.2.1*, ETSI, 12 2014.

[5] *Network Functions Virtualisation (NFV) Infrastructure Compute Domain, ETSI GS NFV-INF 003 - V1.1.1*, ETSI, 12 2021.

[6] *Network Functions Virtualisation (NFV) Infrastructure Hypervisor Domain, ETSI GS NFV-INF 004 - V1.1.1*, ETSI, 1 2015.

[7] *Network Functions Virtualisation (NFV) Infrastructure Network Domain, ETSI GS NFV-INF 005 - V1.1.1*, ETSI, 12 2014.

[8] *Network Functions Virtualisation (NFV) Management and Orchestration, ETSI GS NFV-MAN 001 - V1.1.1*, ETSI, 12 2014.

[9] *Zero-touch network and Service Management (ZSM) Reference Architecture, ETSI GS ZSM 002 - V1.1.1*, ETSI, 8 2019.

[10] *Zero-touch network and Service Management (ZSM) Requirements based on documented scenarios, ETSI GS ZSM 001 - V1.1.1*, ETSI, 10 2019.

[11] *Zero-touch network and Service Management (ZSM) Closed-Loop Automation Part 1: Enablers, ETSI GS ZSM 009-1 - V1.1.1*, ETSI, 6 2021.

[12] Y. Wang, R. Forbes, U. Elzur, J. Strassner, A. Gamelas, H. Wang, S. Liu, L. Pesando, X. Yuan, and S. Cai, "From design to practice: Etsi eni reference architecture and instantiation for network management and orchestration using artificial intelligence," *IEEE Communications Standards Magazine*, vol. 4, no. 3, pp. 38–45, 2020.

[13] Y. Wang, R. Forbes, C. Cavigioli, H. Wang, A. Gamelas, A. Wade, J. Strassner, S. Cai, and S. Liu, "Network management and orchestration using artificial intelligence: Overview of etsi eni," *IEEE communications standards magazine*, vol. 2, no. 4, pp. 58–65, 2018.

[14] D. M. Gutierrez-Estevez, M. Gramaglia, A. De Domenico, G. Dandachi, S. Khatibi, D. Tsolkas, I. Balan, A. Garcia-Saavedra, U. Elzur, and Y. Wang, "Artificial intelligence for elastic management and orchestration of 5g networks," *IEEE Wireless Communications*, vol. 26, no. 5, pp. 134–141, 2019.

[15] D. Sabella, A. Alleman, E. Liao, M. Filippou, Z. Ding, L. G. Baltar, S. Srikanteswara, K. Bhuyan, O. Oyman, G. Schatzberg *et al.*, "Edge computing: from standard to actual infrastructure deployment and software development," *ETSI White paper*, pp. 1–41, 2019.

[16] *Multi-access edge computing (MEC) framework and reference architecture, ETSI GS MEC 003 - V2.1.1*, ETSI, 8 2019.

[17] Y. Sun, Z. Chen, M. Tao, and H. Liu, "Communications, caching, and computing for mobile virtual reality: Modeling and tradeoff," *IEEE Transactions on Communications*, vol. 67, no. 11, pp. 7573–7586, 2019.

[18] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5g network edge cloud architecture and orchestration," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1657–1681, 2017.

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

[20] X. Jiang, F. R. Yu, T. Song, and V. C. Leung, "A survey on multi-access edge computing applied to video streaming: Some research issues and challenges," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 871–903, 2021.

[21] *Multi-access Edge Computing (MEC) Framework and Reference Architecture, ETSI GS MEC 003 - V2.1.1*, ETSI, 1 2019.

[22] L. M. Contreras and C. J. Bernardos, "Overview of architectural alternatives for the integration of etsi mec environments from different administrative domains," *Electronics*, vol. 9, no. 9, p. 1392, 2020.

[23] *Multi-access Edge Computing (MEC) Study on Inter-MEC systems and MEC-Cloud system coordination, ETSI GR MEC 035 - V3.1.1*, ETSI, 6 2021.

[24] B. Confais, A. Lebre, and B. Parrein, "Performance analysis of object store systems in a fog and edge computing infrastructure," in *Transactions on Large-Scale Data-and Knowledge-Centered Systems XXXIII*. Springer, 2017, pp. 40–79.

[25] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, "Freenet: A distributed anonymous information storage and retrieval system," in *Designing privacy enhancing technologies*. Springer, 2001, pp. 46–66.

[26] A.-G. Gheorghe, C.-C. Crecana, C. Negru, F. Pop, and C. Dobre, "Decentralized storage system for edge computing," in *2019 18th International Symposium on Parallel and Distributed Computing (ISPDC)*. IEEE, 2019, pp. 41–49.

[27] I. Lujic, V. De Maio, and I. Brandic, "Efficient edge storage management based on near real-time forecasts," in *2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC)*. IEEE, 2017, pp. 21–30.

[28] J. Xing, H. Dai, and Z. Yu, "A distributed multi-level model with dynamic replacement for the storage of smart edge computing," *Journal of Systems Architecture*, vol. 83, pp. 1–11, 2018.

[29] Y. Huang, X. Song, F. Ye, Y. Yang, and X. Li, "Fair and efficient caching algorithms and strategies for peer data sharing in pervasive edge computing environments," *IEEE Transactions on Mobile Computing*, vol. 19, no. 4, pp. 852–864, 2019.

[30] T. Hou, G. Feng, S. Qin, and W. Jiang, "Proactive content caching by exploiting transfer learning for mobile edge computing," *International Journal of Communication Systems*, vol. 31, no. 11, p. e3706, 2018.

[31] Z. Chang, L. Lei, Z. Zhou, S. Mao, and T. Ristaniemi, "Learn to cache: Machine learning for network edge caching in the big data era," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 28–35, 2018.

[32] L. Zhang, J. Wu, S. Mumtaz, J. Li, H. Gacanin, and J. J. Rodrigues, "Edge-to-edge cooperative artificial intelligence in smart cities with on-demand learning offloading," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[33] S. Sondur, K. Kant, S. Vucetic, and B. Byers, "Storage on the edge: Evaluating cloud backed edge storage in cyberphysical systems," in *2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, 2019, pp. 362–370.

[34] T. Theodoropoulos, A.-C. Maroudis, J. Violos, and K. Tserpes, "An encoder-decoder deep learning approach for multistep service traffic prediction," in *2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService)*, 2021, pp. 33–40.

[35] M. Caprolu, R. Di Pietro, F. Lombardi, and S. Raponi, "Edge computing perspectives: architectures, technologies, and open security issues," in *2019 IEEE International Conference on Edge Computing (EDGE)*. IEEE, 2019, pp. 116–123.

[36] S. Shahzadi, M. Iqbal, T. Dagiuklas, and Z. U. Qayyum, "Multi-access edge computing: open issues, challenges and future perspectives," *Journal of Cloud Computing*, vol. 6, no. 1, pp. 1–13, 2017.

[37] B. Varghese, N. Wang, S. Barbhuiya, P. Kilpatrick, and D. S. Nikolopoulos, "Challenges and opportunities in edge computing," in *2016 IEEE International Conference on Smart Cloud (SmartCloud)*. IEEE, 2016, pp. 20–26.

[38] R. Yang, F. R. Yu, P. Si, Z. Yang, and Y. Zhang, "Integrated blockchain and edge computing systems: A survey, some research issues and challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1508–1532, 2019.

[39] C. Luo, L. Xu, D. Li, and W. Wu, "Edge computing integrated with blockchain technologies," in *Complexity and Approximation*. Springer, 2020, pp. 268–288.

[40] K. Samdanis and T. Taleb, "The road beyond 5g: A vision and insight of the key technologies," *IEEE Network*, vol. 34, no. 2, pp. 135–141, 2020.

[41] S. Kekki, W. Featherstone, Y. Fang, P. Kuure, A. Li, A. Ranjan, D. Purkayastha, F. Jiangping, D. Frydman, G. Verin *et al.*, "Mec in 5g networks," *ETSI white paper*, vol. 28, pp. 1–28, 2018.

[42] N. Sprecher *et al.*, "Harmonizing standards for edge computing-a synergized architecture leveraging etsi isg mec and 3gpp specifications," *ETSI White paper No. 36, no. 979-10-92620-35-5*, 2020.

[43] E. Coronado, Z. Yousaf, and R. Riggio, "Lightedge: mapping the evolution of multi-access edge computing in cellular networks," *IEEE Communications Magazine*, vol. 58, no. 4, pp. 24–30, 2020.

[44] N. Finn, "Time-sensitive and deterministic networking whitepaper," 2017.

[45] M. Chen, X. Geng, and Z. Li, "Segment routing (sr) based bounded latency," *Internet Engineering Task Force, Internet-Draft draft-chendetnet-sr-based-bounded-latency-00*, 2018.

[46] F. Chiariotti, S. Kucera, A. Zanella, and H. Claussen, "Leap: A latency control protocol for multi-path data delivery with pre-defined qos guarantees," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2018, pp. 166–171.

[47] S. Ha, I. Rhee, and L. Xu, "Cubic: A new tcp-friendly high-speed tcp variant," *SIGOPS Oper. Syst. Rev.*, vol. 42, no. 5, p. 64–74, Jul. 2008. [Online]. Available: https://doi.org/10.1145/1400097.1400105

[48] *5G, NG-RAN, Architecture description, ETSI TS 138 401 - V15.5.0*, ETSI, 5 2019.

[49] S. Niknam, A. Roy, H. S. Dhillon, S. Singh, R. Banerji, J. H. Reed, N. Saxena, and S. Yoon, "Intelligent o-ran for beyond 5g and 6g wireless networks," *arXiv preprint arXiv:2005.08374*, 2020.

[50] "Information technology — Computer graphics, image processing and environmental data representation — Mixed and augmented reality (MAR) reference model," International Organization for Standardization, Standard, Feb. 2019.

[51] J. Lee, Y. Lee, S. Lee, and G. J. Kim, "Standardization for augmented reality: introduction of activities at iso-iec sc 24 wg 9," in *Proceedings of the 12th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, 2013, pp. 279–280.

[52] T. Huang and Y. Liu, "3d point cloud geometry compression on deep learning," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 890–898.

[53] D. C. Garcia and R. L. de Queiroz, "Intra-frame context-based octree coding for point-cloud geometry," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 1807–1811.

[54] T. Wiemann, F. Igelbrink, S. Pütz, M. K. Piening, S. Schupp, S. Hinderink, J. Vana, and J. Hertzberg, "Compressing ros sensor and geometry messages with draco," in *2019 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2019, pp. 243–248.

[55] A. Varischio, F. Mandruzzato, M. Bullo, M. Giordani, P. Testolina, and M. Zorzi, "Hybrid point cloud semantic compression for automotive sensors: A performance evaluation," *arXiv preprint arXiv:2103.03819*, 2021.

[56] M. Hosseini and C. Timmerer, "Dynamic adaptive point cloud streaming," in *Proceedings of the 23rd Packet Video Workshop*, 2018, pp. 25–30.

[57] X. Sun, S. Wang, M. Wang, S. S. Cheng, and M. Liu, "An advanced lidar point cloud sequence coding scheme for autonomous driving," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 2793–2801.

[58] L. Wiesmann, A. Milioto, X. Chen, C. Stachniss, and J. Behley, "Deep compression for dense point cloud maps," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2060–2067, 2021.

[59] B. Han, Y. Liu, and F. Qian, "Vivo: Visibility-aware mobile volumetric video streaming," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, 2020, pp. 1–13.

[60] C. Portaneri, P. Alliez, M. Hemmer, L. Birklein, and E. Schoemer, "Cost-driven framework for progressive compression of textured meshes," in *Proceedings of the 10th ACM Multimedia Systems Conference*, 2019, pp. 175–188.

[61] K. Christaki, E. Christakis, P. Drakoulis, A. Doumanoglou, N. Zioulis, D. Zarpalas, and P. Daras, "Subjective visual quality assessment of immersive 3d media compressed by open-source static 3d mesh codecs," in *International Conference on Multimedia Modeling*. Springer, 2019, pp. 80–91.

[62] *Augmented Reality Framework (ARF); AR framework architecture, ETSI GS ARF 004-2 - V1.1.1*, ETSI, 8 2021.

[63] D. Graziosi, O. Nakagami, S. Kuma, A. Zaghetto, T. Suzuki, and A. Tabatabai, "An overview of ongoing point cloud compression standardization activities: video-based (v-pcc) and geometry-based (g-pcc)," *APSIPA Transactions on Signal and Information Processing*, vol. 9, p. e13, 2020.

[64] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel, "On the shape of a set of points in the plane," *IEEE Transactions on information theory*, vol. 29, no. 4, pp. 551–559, 1983.

[65] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," *IEEE transactions on visualization and computer graphics*, vol. 5, no. 4, pp. 349–359, 1999.

[66] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proceedings of the fourth Eurographics symposium on Geometry processing*, vol. 7, 2006.