ISG ZSM PoC Proposal: Building Cloud AR/VR Service on CAMARA by Using ZSM Framework

1 PoC Project Details

1.1 PoC Project

PoC Number: *(assigned by ETSI)*	#8
PoC Project Name:	Building Cloud AR/VR Service on CAMARA by Using ZSM Framework
PoC Project Host:	Telefonica
Short Description:	This PoC demonstrates the possibility of building a cloud-based AR/VR service on CAMARA platform using the ZSM framework. The use-case contributes to the NaaS paradigm by providing both the networking requirements and the edge cloud PaaS requirements of such applications. ZSM management services (defined in ZSM 002) and cross-domain orchestration (ZSM 008) will be demonstrated through the realization of the ETSI F5G 008's Cloud VR Gaming Use-case. The solution of this PoC will leverage and extend open source projects including ETSI TeraflowSDN (TFS), ETSI OSM, and IETF Kafka-YANG data mesh.

1.2 PoC Team Members

	Organisation name	ISG ZSM participant (yes/no)	Contact (Email)	PoC Point of Contact (*)	Role (**)	PoC Components
1	Telefonica	Yes	joseantonio.ordonezlucena@telefonica.com diego.r.lopez@telefonica.com	X	Network operator	• Contributes to technical details and the architecture of the PoC • ZSM Consumer implementation on CAMARA
2	Huawei Technologies	Yes	henry.yu1@huawei.com hesam.rahimi@huawei.com	X	Network supplier	• Contributes to technical details and the architecture of the PoC • E2E MD implementation using OSM • Cloud MD implementation • Provide domain controller(s) and network equipment for the PoC
3	Centre Tecnològic de Telecomunicacions de Catalunya (CTTC)	Yes	lluis.gifre@cttc.es ricard.vilalta@cttc.es raul.munoz@cttc.es	X	Research Center	• Contributes to technical details and the architecture of the PoC • Transport MD implementation using TFS
(*) Identify the PoC Point of Contact with an X. (**) The Role will be network/service provider, supplier, or other (universities, research centers, test labs, Open Source projects, integrators, etc…).						

All the PoC Team members listed above declare that the information in this proposal is conformant to their plans at this date and commit to inform ETSI timely in case of changes in the PoC Team, scope or timeline.

1.3 PoC Project Scope

1.3.1 PoC Topics

PoC Topics identified in this clause need to be taken for the PoC Topic List identified by ISG ZSM and publicly available in the ZSM WIKI. PoC Teams addressing these topics commit to submit the expected contributions in a timely manner.

PoC Topic Code	PoC Topic Description	Related WI	Expected Contribution	Target Date
Topic 4 (Cross-domain user-driven E2E services)	Demonstration of the cross-domain orchestration and automation between E2E MD, Transport MD, and Cloud MD	ZSM002 and ZSM008	Demos	Feb 2024

1.3.2 Other topics in scope

List here any additional topic for which the PoC plans to provide input/feedback to the ISG ZSM.

PoC Topic Code	PoC Topic Description	Related WG/WI	Expected Contribution	Target Date
A				
B				
C				
D				

1.4 PoC Project Milestones

PoC Milestone	Milestone description	Target Date	Additional Info
P.S	PoC Project Start	Oct 2023	
P.D1	PoC Demo	Feb 2024	Demonstration of Cloud AR/VR Gaming service creation, deployment, orchestration, and termination via CAMARA service APIs
P.C1	Feedback to ISG ZSM on potential improvements of the ZSM standards tested by the PoC	Mar 2024	This PoC relates to ZSM 002 and ZSM 008,. Potential improvements of these standards, if any, might be discovered by the PoC, and will be documented in the PoC Report.
P.R	PoC Report	Apr 2024	
P.E	PoC Project End	Apr 2024	

NOTE: Milestones need to be entered in chronological order.

1.5 PoC Demonstration Plan

The **CAMARA Initiative** was introduced and discussed in ZSM#19 last year, calling for participation and collaboration. Through this PoC, we would like to contribute to the **Network-as-a-Service (NaaS)** paradigm on CAMARA by implementing a solution using the ZSM framework. More precisely, we propose to build a **Cloud AR/VR Gaming use-case** on CAMARA to demonstrate the services and capabilities of NaaS. The Cloud AR/VR gaming use-case is chosen for the PoC because it is comprehensive enough to illustrate the essential aspects of NaaS and, at the same time, is also simple enough for us to develop a solution for demonstration within a reasonable amount of time.

The solution of the PoC will be carried out as a ZSM framework which consists of three MDs: an E2E MD, a Transport MD, and a Cloud MD. This implementation will demonstrate the essential features of the ZSM architecture specified in ZSM 002 and 008. Furthermore, CAMARA, which exposes the NaaS Service APIs to AR/VR gaming developers, plays the ZSM service consumer role, consuming the services offered by the E2E MD. The PoC will demonstrate that gaming developers may call CAMARA Service APIs to perform tasks, such as cloud AR/VR gaming service creation, deployment, orchestration, and termination.

The solution will be standards-based and open. It will be mainly implemented on open source software, including CAMARA, ETSI OSM, ETSI TSF, IETF Kafka-YANG, Openstack, and Kubernetes. Huawei, as a network supplier, will provide computing hardware and network equipment for the PoC. The hardware will be setup in Huawei's lab in Wuhan, China. The lab will be remotely accessible via VPN connections.

The target development completion date for the PoC is February 2024. And Given the PoC's relevance to CAMARA and GSMA, we propose to perform the demonstration at the MWC in 2024.

1.6 Additional Details

Information about CAMARA, OSM, TFS, and IETF Kafka-YANG is available at the following links respectively:

- https://camaraproject.org/
- https://osm.etsi.org/
- https://tfs.etsi.org/
- https://github.com/network-analytics/draft-daisy-kafka-yang-integration/blob/main/draft-daisy-kafka-yang-integration-04.md

2 PoC Technical Details

2.1 PoC Overview

2.1.1 Use-case Description

Cloud AR/VR gaming, also known as cloud-based AR/VR gaming, refers to a form of gaming where the processing power and rendering capabilities required for AR/VR experiences are offloaded to cloud servers instead of being handled locally on the user's device. In this model, the AR/VR content is streamed to the user's device over high-speed network (which meets low latency requirements), and the device acts as a display and input interface, transmitting user commands back to the cloud server.

Cloud AR/VR gaming offers several advantages and overcomes certain limitations of traditional AR/VR gaming setups: that the user

1- Device independence: With cloud AR/VR gaming, the user's device requirements are reduced as the heavy computational tasks are performed on the cloud server. This means that even devices with limited processing power can access and enjoy high-quality AR/VR experiences. Users can potentially access AR/VR games on a wide range of devices. Including smartphones, tablets, low-spec PCs, or dedicated lightweight VR headsets.

2- Lower Hardware Costs: Since the processing requirements are handled in the cloud, users do not necessarily need expensive hardware to run resource-intensive AR/VR applications. This can lower the barrier to entry for experiencing high-quality AR/VR gaming, as users can rely on cloud infrastructure instead of investing in costly hardware upgrades. That being said, however, cloud AR/VR gaming also introduces a user cost shift from hardware (device) centric to "as-a-service" and subscription-based service model. That is, instead of paying for expensive devices, the user will pay for the service. A subscription model brings cost benefit to end user, because the continuous cost reduction of computing hardware (as predicted by the Moore's law) may lead to the reduction of cloud game server cost and subscription charge.

3- Scalability: Cloud-based infrastructures allows for greater scalability in terms of user capacity. As the processing power is centralized in the cloud servers, the service can accommodate a larger number of users simultaneously without requiring the individual devices to handle the computational load. This scalability is particularly beneficial for multiplayer AR/VR gaming experiences.

4- Content Updates and Maintenance: With cloud-based systems, game updates, patches, and maintenance can be managed centrally on the cloud servers, this eliminates the need for users to manually download and install updates on their devices, ensuring they always have access to the latest versions of games without disruptions.

GSMA [Cloud AR/VR whitepaper]provides an overview of Cloud AR/VR, and it gives the following takeaways:

- AR/VR has variety of applications which are deployed in enterprise and consumer domains (Fig. 1).

Figure 1: Examples of AR/VR apps in enterprise and consumer domain

- The whitepaper identifies a few key challenges exiting in today's local-PC based AR/VR service, such as high computing device cost and low visual quality. It proposes that, to resolve these challenges, we should migrate to the Cloud AR/VR solution (Fig. 2).

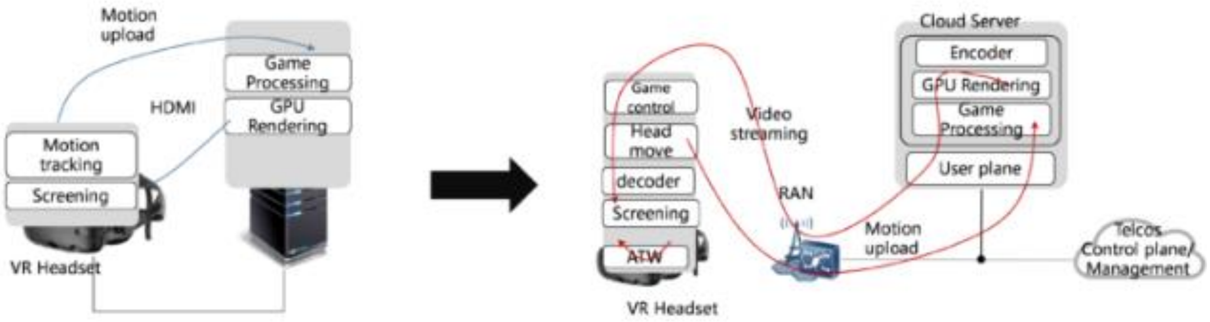

Figure 2: Migration from local-PC based VR to Cloud VR

- The whitepaper proposes an Edge Cloud PaaS architecture. However, NaaS can also be achieved by providing both networking and cloud service (Fig. 3).

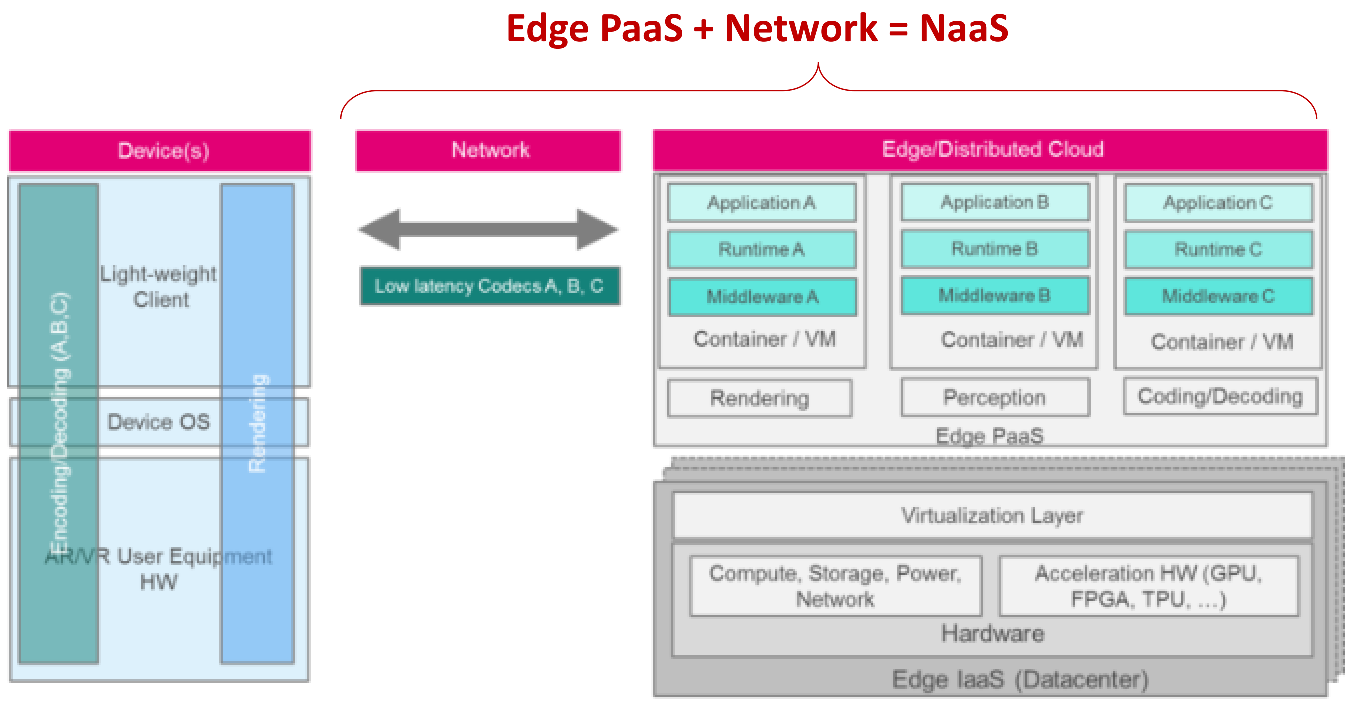

Figure 3: Cloud AR/VR reference architecture. Note that NaaS can be achieved by incorporating the network services into the architecture.

2.1.2 Cloud AR/VR Service: How Is It Relevant to CAMARA

CAMARA works closely with the GSMA Operator Platform Group to define service APIs of NaaS. We can integrate CAMARA with the Cloud AR/VR architecture to provide a full NaaS solution (Fig. 4).

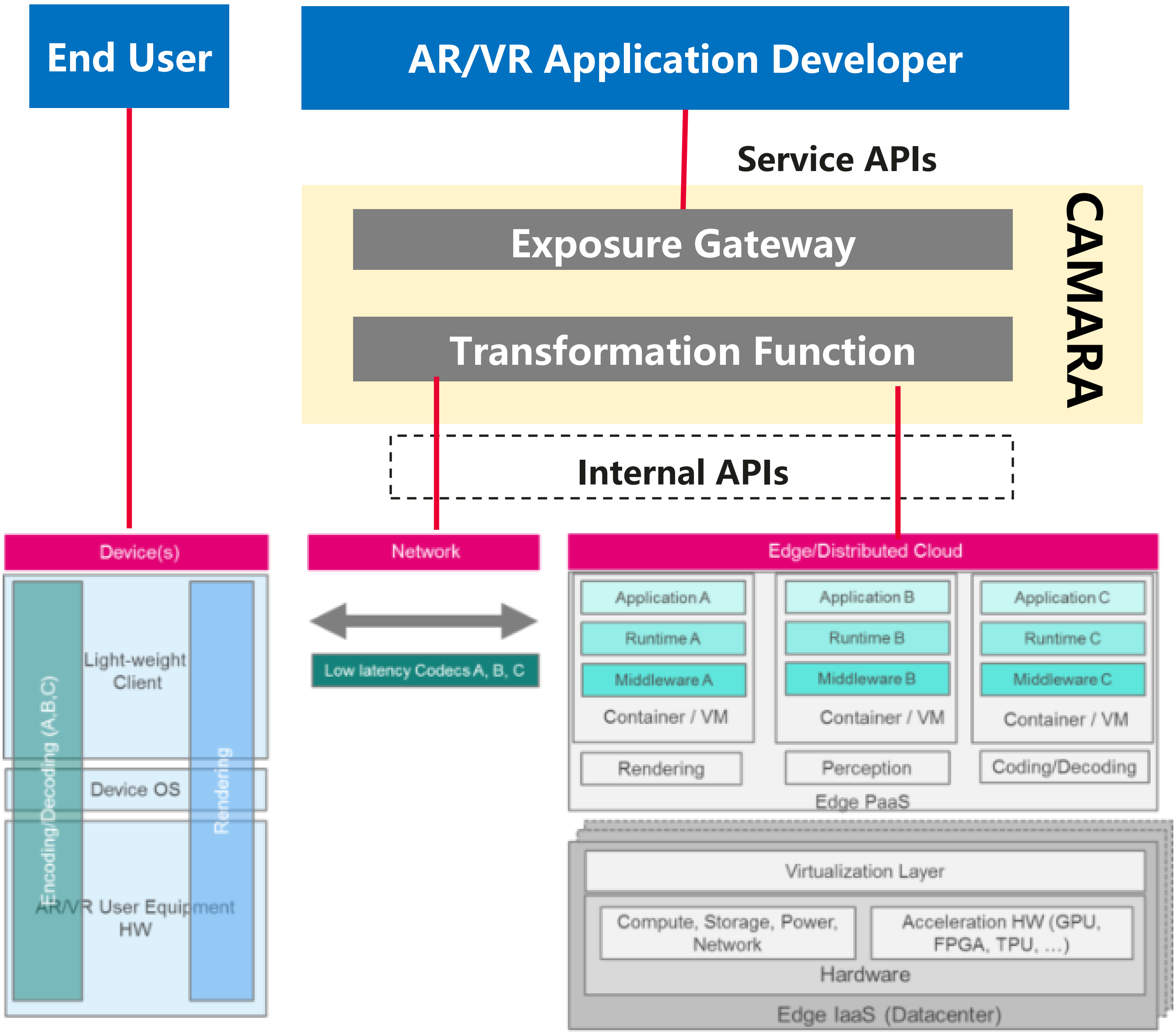

Figure 4: Cloud AR/VR reference architecture integrated with CAMARA

As depicted in Figure 4, CAMARA contains the following capabilities:

- **Service APIs**: APIs which are made available for consumption to AR/AR application developers, e.g., AR gaming developers
- **Internal APIs**: Implemented in telco assets, including network resources and cloud resources. These APIs are typically defined in SDOs or industry fora, and quiet tied to the underlying technology
- **Exposure Gateway**: It provides the service APIs that are needed by the AR/VR application developer to deploy the application, e.g., on-demand bandwidth allocation, cloud computing resource allocation, etc.
- **Transformation function**: Keeps information on correspondences between service APIs and internal APIs. E.g., executing workflows to enforce these mappings.

2.1.3 Cloud AR/VR Service: Leveraging the Underlay Network Technology from ETSI F5G's Cloud VR Gaming Use-Case

ETSI F5G 008 describes a Cloud VR Gaming use-case (Fig. 5). Their physical network technology can be used to support our CAMARA/ZSM PoC.

Huawei is able to provide the physical network (encircled within the blue lines of Fig. 5) for the PoC.

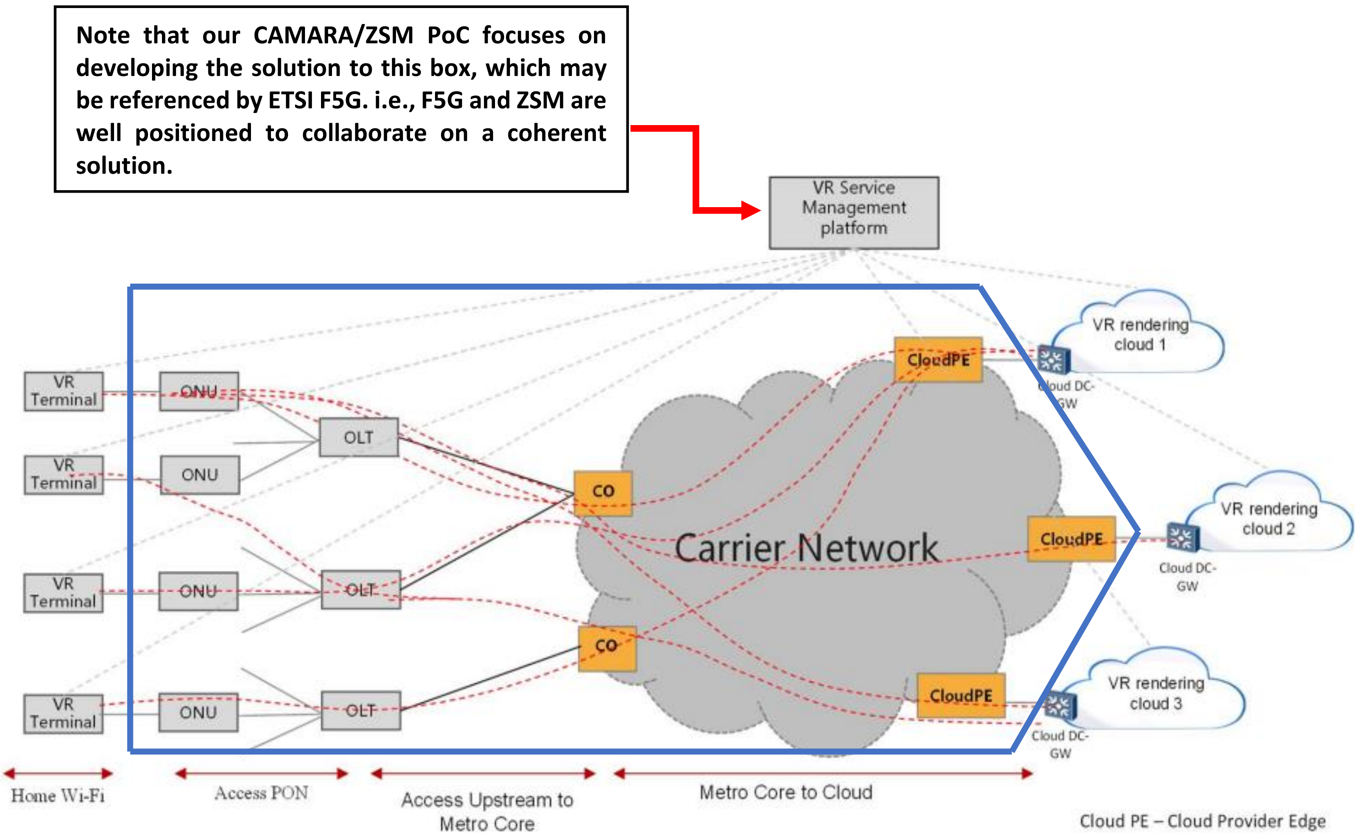

Figure 5: ETSI F5G Cloud VR Gaming use-case

2.1.4 Apply ZSM Framework to a NaaS Implementation

In section 2.1.1 and 2.1.2, we show that a NaaS platform could be achieved by combining the cloud AR/VR reference architecture (which offers integrated networking and edge/distributed cloud services) with the CAMARA service APIs (Figure 4). Furthermore, a cloud VR gaming service can be a typical use-case to illustrate the functionality of such platform. A VR game developer (i.e., customer of the platform), for example, may use/call CAMARA service APIs to deploy and run a game service on the networking and cloud resources provided by the platform. We also mention, in section 2.1.3, that a F5G underlay networking technology can be used to support the use-case (Figure 5).

However, only knowing a conceptual/logical architecture of NaaS (as depicted in Figure 4) and an underlay networking technology (as illustrated in Figure 5) is inadequate to carry out an implementation of a NaaS solution. To implement NaaS, a software architecture which defines design components and their interfaces is required, and the ZSM framework serves that purpose. In this PoC, we choose the ZSM framework as the software architecture of our implementation. The ZSM framework has the following characteristics which makes it able to meet the key requirements of a NaaS solution:

- It is an end-to-end (E2E) service management solution. A NaaS platform is required to provide E2E services to its users. This responsibility can be fulfilled by the E2E Management Domain (MD) in the ZSM framework, with help from other MDs.

- It provides a federated solution consisting of multiple management domains (MDs). A NaaS platform is required to provide services by managing and orchestrating heterogenous networking and computing resources, such as wireless networks, fixed networks, and cloud computing and networking. The ZSM framework manages heterogenous networks through the orchestration of MDs. Each MD is responsible for a particular portion of the network which use the same technology and management interface (e.g., wireless MD, Transport MD, Cloud MD, etc.).
- It is scalable and extendable. A NaaS platform is required to be able to evolve and extend. For example, in this PoC, a wireless network (e.g, RAN) is not included (i.e., we only demonstrate playing VR games from home). However, Our ZSM solution can be extended to demonstrate mobile gaming by adding and implementing a wireless MD, without any architectural rework.

2.1.5 Scope and Content of the PoC

The goal of this PoC is to demonstrate the basic functionality of NaaS via the cloud VR gaming use-case. To do so, we demonstrate the scenario in which two VR game developers deploy their games into one cloud via CAMARA service APIs. This implies, therefore, that the solution of this PoC needs to have the network slicing capability, so that it can provide service isolation for the two game developers by slicing the network and cloud resources.

In addition, this PoC demonstrates two end users who are able to access (or play) both games. In other words, the PoC needs to be able to create two network slices (e.g., L3 VPNs), each of which contains both end user devices.

In summary, this PoC demonstrates the creation of two isolated gaming services on a single NaaS platform using a common underlay network and cloud resources. Furthermore, within each isolated service, the gaming service lifecycle (game installation, activation, deactivation, and uninstallation) is demonstrated. Through this demonstration, the ZSM functionalities including management services (defined in ZSM 002), cross-domain orchestration (ZSM 002), and integration fabric (ZSM 008) will be illustrated and studied.

2.1.6 Design Requirements

Although the PoC demonstrates a specific use-case (i.e., Cloud AR/VR service), we try to develop a generic solution, independent from specific use-cases and technologies, to maximize its reusability. The following design requirements, which are highlighted in Figure 6, are identified to achieve this goal:

1- **Architecture**: The top-level design follows the ZSM framework which has well defined management domains. This allows us to develop a flexible and generic federated solution.

2- **Service APIs**: CAMARA Service APIs that are used by, but not limited to, AR/VR application developers. These APIs may include Network quality-on-demand (QoD), Edge cloud, Home device QoD, etc.

3- **Cross-domain integration fabric**: Integration fabric is needed between E2E MD and MDs. We need features such as Transport or Cloud service discovery and registration, and event notification and streaming. For the latter, Kafka-YANG data mesh may be used as an implementation choice.

4- **Transport MD NBI**: Needs to be intent-like and standards-based

5- **Cloud MD NBI**: Needs to be intent-like and standards-based

Note: Another design requirement could be the closed-loop automation of the MDs (which is defined in ZSM 009). Although this requirement may be necessary for a robust and complete solution, due to the time constraint of this PoC, we may need to defer it to the next PoC development phase. i.e., We plan to develop and evolve the solution in multiple PoC phases.

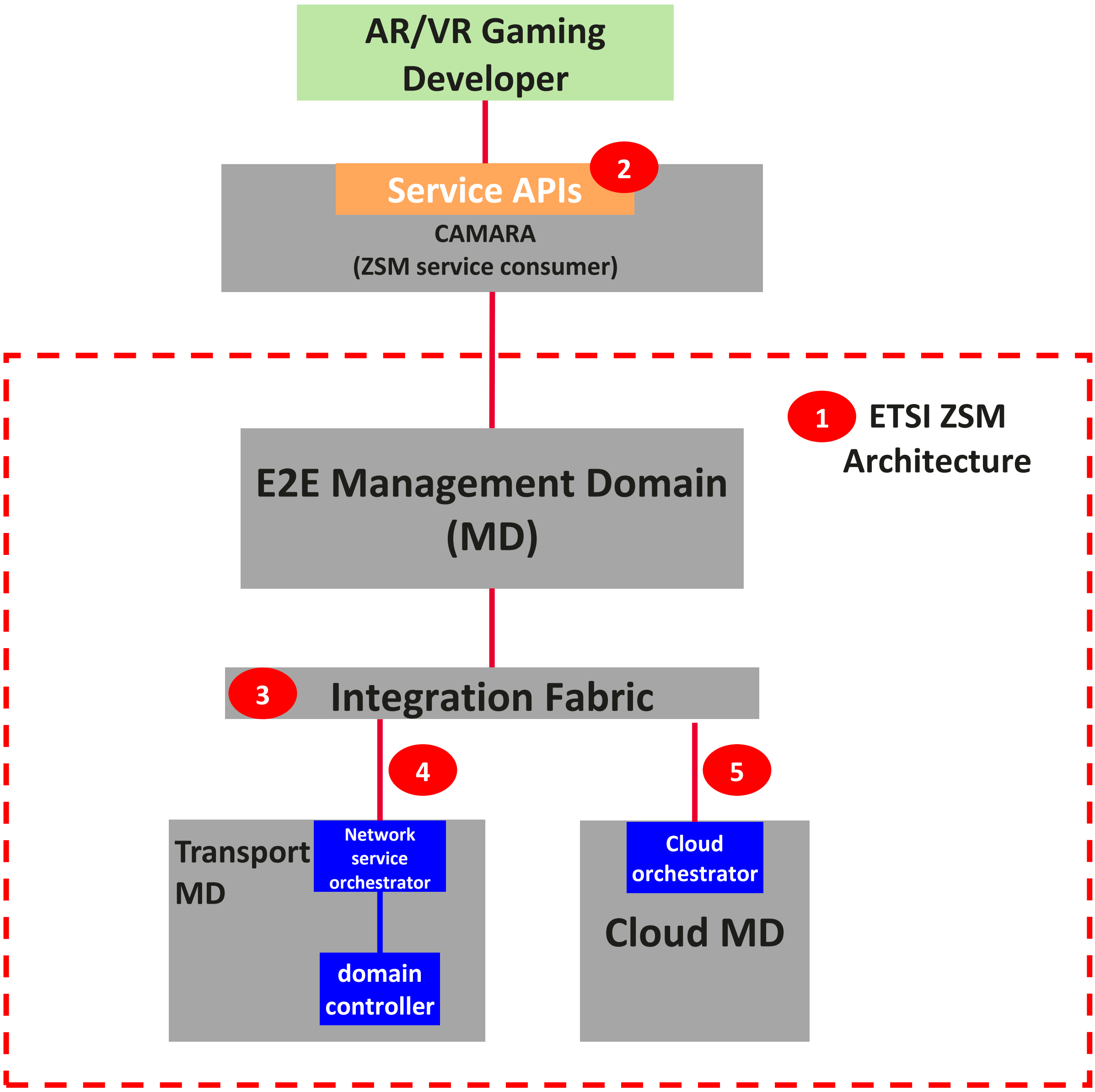

Figure 6: Major design requirements for the PoC

2.1.7 Top Level Design

This PoC demonstrates the service automation of the cloud-based VR gaming experience. The design architecture of the use-case follows the ZSM framework (ZSM 002) and other relevant ZSM standards (ZSM 008). Figure 7 depicts the use-case realization using the ZSM framework and CAMARA.

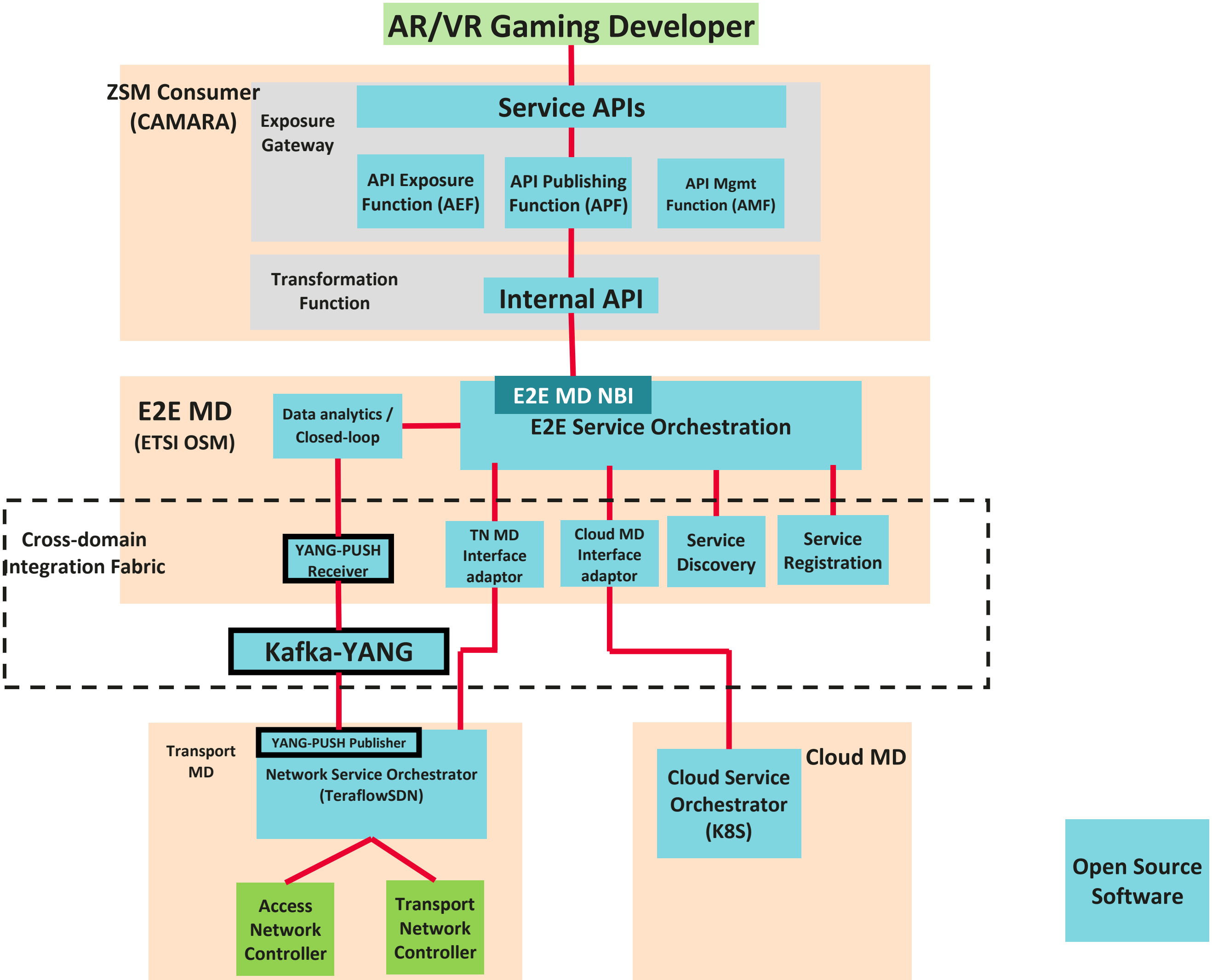

Figure 7: PoC High-Level Design (Based on the design requirements)

CAMARA is responsible for offering NaaS capabilities, via its service APIs, to VR game developers. More specifically, the following two service APIs are used in this PoC:

- *Edge Cloud*. This API contains several sub-APIs, among which the Poc uses the following two sub-APIs:
    - *MEC (Multi-access Edge Compute) Exposure & Experience Management API*. The game developer uses this API for discovery and utilization of MEC resources and services. In other words, this API used to find an optimal cloud site which may provide the computing resources (e.g., VMs or containers) for the gaming software deployment.
    - *MEC Edge Cloud API*. The game developer uses this API for game software artifact management, such as installing or removing the game software.
- *Network QoD (Quality on Demand).* This API is used to establish the network connections between the gaming devices (i.e., application client) and the game server in the cloud (i.e., application server).

As shown in Fig. 7, CAMARA is positioned as the ZSM service consumer, which consumes the services offered by the E2E MD. CAMARA's Transformation Function module is responsible for translating the service APIs into the corresponding E2E MD NBIs, as well as invoking them on the E2E MD.

This implies that the E2E MD needs to resume two responsibilities. First, it needs to offer the NBIs to which the CAMARA service APIs can be mapped. In this PoC, we propose to use the IETF Network Slice Service model (draft-ietf-teas-ietf-network-slice-nbi-yang) as the NBI to which the CAMARA Network QoD can be mapped. Since this PoC involves only fixed network (i.e., no wireless network segments), the IETF model is adequate to specify the network service required by the Network QoD. Furthermore, as mentioned in section 2.1.4, providing service isolation among game developers/vendors is a necessary capability of an NaaS platform. Thus, the E2E MD needs to have network slicing capabilities, and this is another reason why we choose the IETF Network Slice Service model as an NBI. Another type of the NBIs which the E2E MD needs to provide is the cloud management APIs to which the CAMARA Edge Cloud API is mapped. We propose to use OpenStack or Kubernetes APIs for this type of the NBIs.

Second, the E2E MD is also responsible for the service creation and orchestration needed to fulfil these NBIs. To achieve this, the E2E MD decomposes an E2E service into the corresponding management services offered by the Transport and the Cloud MD, and then invoke them on the MDs. The cross-domain integration fabric facilitates the communication between the E2E MD and the Transport and the Cloud MD. The Transport and the Cloud MD register their Management Services to the cross-domain integration fabric. The E2E MD then discovers and consumes these services.

We propose to implement the E2E MD on ETSI OSM. OSM already provides some capabilities of slicing and cloud resource management. So, we can reuse the existing code with some (if not minor) extension to complete the implementation.

The Transport MD is responsible for the service provisioning and network configurations of the physical network. Its northbound interfaces (i.e., Management Service APIs) are intent-like (in the case of this PoC), model-driven, technology agnostic, and open standard-based. In this PoC, we propose to implement the Transport MD's northbound interface and transport service orchestration on ETSI TFS. TFS is an open-source, micro-service based, and cloud-native SDN controller providing features for both flow management (service layer), and network equipment integration (infrastructure layer).

The Cloud MD is responsible for cloud resource (i.e., VMs) management and cloud service orchestration, such as AR/VR gaming software artifact installation and deployment in cloud, gaming service instantiation, activation and termination. We propose to use Openstack or Kubernetes as the Cloud service orchestrator.

Table 1 enumerates the design components of the PoC.

Table 1 Design Components of the PoC

ZSM Design Components	Design Component Elements	Implementation
ZSM Consumer (CAMARA)	Exposure Gateway (Service APIs)	CAMARA Network QoD API and Edge Cloud API
	Transformation Function	CAMARA Internal API Mapper
E2E MD	E2E Service Orchestration	ETSI OSM
Transport MD	Network Service Orchestrator	ETSI TFS
	Data analytics and closed-loop (*Note: due to time constraint, we may postpone this feature until a next PoC*)	ETSI TFS
	Fiber Access Network Controller	Vendor product
	Transport Network Controller	Vendor product
Cloud MD	Cloud Service Orchestrator	OpenStack or Kubernetes
Cross-domain Integration Fabric	YANG-PUSH Receiver, Service Discovery, Service Registration, etc.	TeraflowSDN
	Kafka-YANG Data Mesh	IETF Kafka-YANG project

2.1.8 Interfaces Between Design Components

The PoC implementation uses standards-based interfaces and open source APIs between the design components (MDs, Integration Fabric, etc). These interfaces are highlighted in Figure 8 and are explained below:

1- **CAMARA Service API.** APIs which offer NaaS to OTT application developers, i.e., AR/VR gaming developers in our case.

- As discussed in section 2.1.6, this PoC uses the Network QoD API, MEC Exposure & Experience Management API, and MEC Edge Cloud API.

2- **Domain-specific, tech-agnostic interface.** E2E MD NBI is domain-specific (Transport vs. Cloud), but technology-agnostic.

- In this PoC, we use the IETF Network Slice service model (ietf-teas-ietf-network-slice-nbi-yang), which allows us to specify transport slicing service without specifying the underlay networking technology. In addition, we also use OpenStack or Kubernetes APIs for cloud services.

3- **IETF YANG-PUSH (RFC8641).** Event notifications and telemetry data streaming from the transport MD.

- Due to time constraint and network controller limitations, a YANG-PUSH mechanism and data streaming feature may be postponed until a next PoC. However, in this PoC, we will develop event notifications and support service discovery.

4- **IETF Customer Service Models.** Customer service models (RFC8309), such as L2SM or L3VPN-SVC, may be used for Transport MD NBI. However, note that, as a ZSM federated solution, our implementation should not limit a MD's NBI to a specific standard. Thus, MD interface adaptors are used in E2E MD to decouple it from any specific MD NBI standard.

- In this PoC, the L3VPN-SVC (RFC8299) model is used.

5- **IETF ACTN MPI**. The transport network controller serves as Management Functions in the Transport MD. We use ACTN MPI (RFC8453), a collection of IETF transport network YANG models, as their interfaces.

- Note: In this PoC, the access network controller is not ready to provide a standard-based NBI. So, we only use it to manually configure the access network equipment (prior to the demo).

- Note: OSU Tunnel Service API in Figure 8 refers to NCE-T controller's OSU service creation/deletion APIs. This API is based on the draft-ietf-ccamp-otn-tunnel YANG model (one of the ACTN MPI models), with minor extensions to support the OSU container type.

6- **Kubernetes or OpenStack APIs**. Cloud service orchestration APIs need to support features such as creating/terminating VR gaming containers.

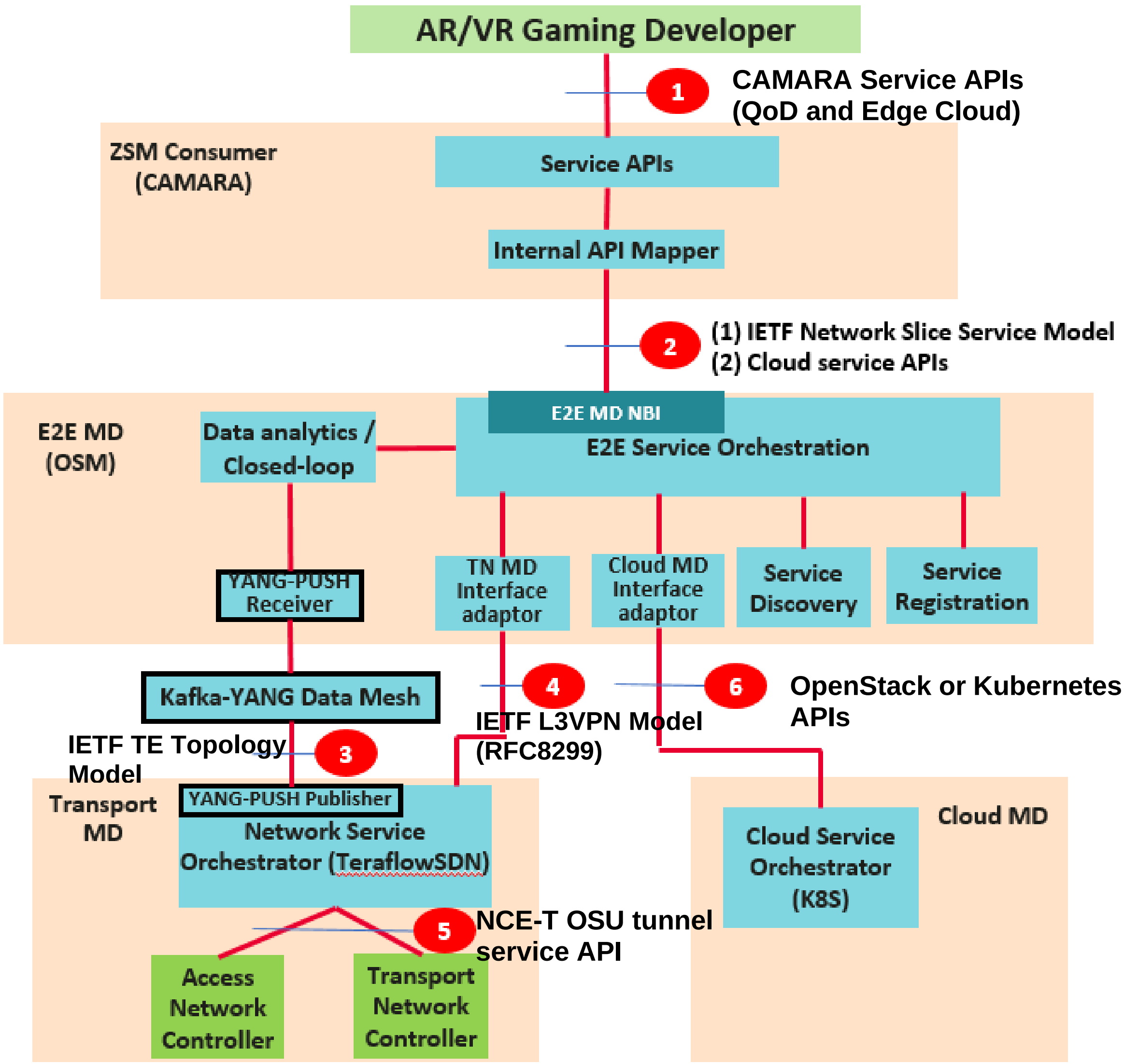

Figure 8: Interfaces between Design Components/Elements

2.2 PoC Software and Hardware setup

Figure 9 depicts the PoC architecture. The transport network (the carrier network) in this demo is an optical network. The network equipment (including physical network SDN controllers) used for the PoC will be commercial products. The software we use and develop for this PoC will be open source projects. The open-source projects in this PoC is illustrated in “yellow” boxes in Figure 9 below, and the vendor products are illustrated in “blue” boxes.

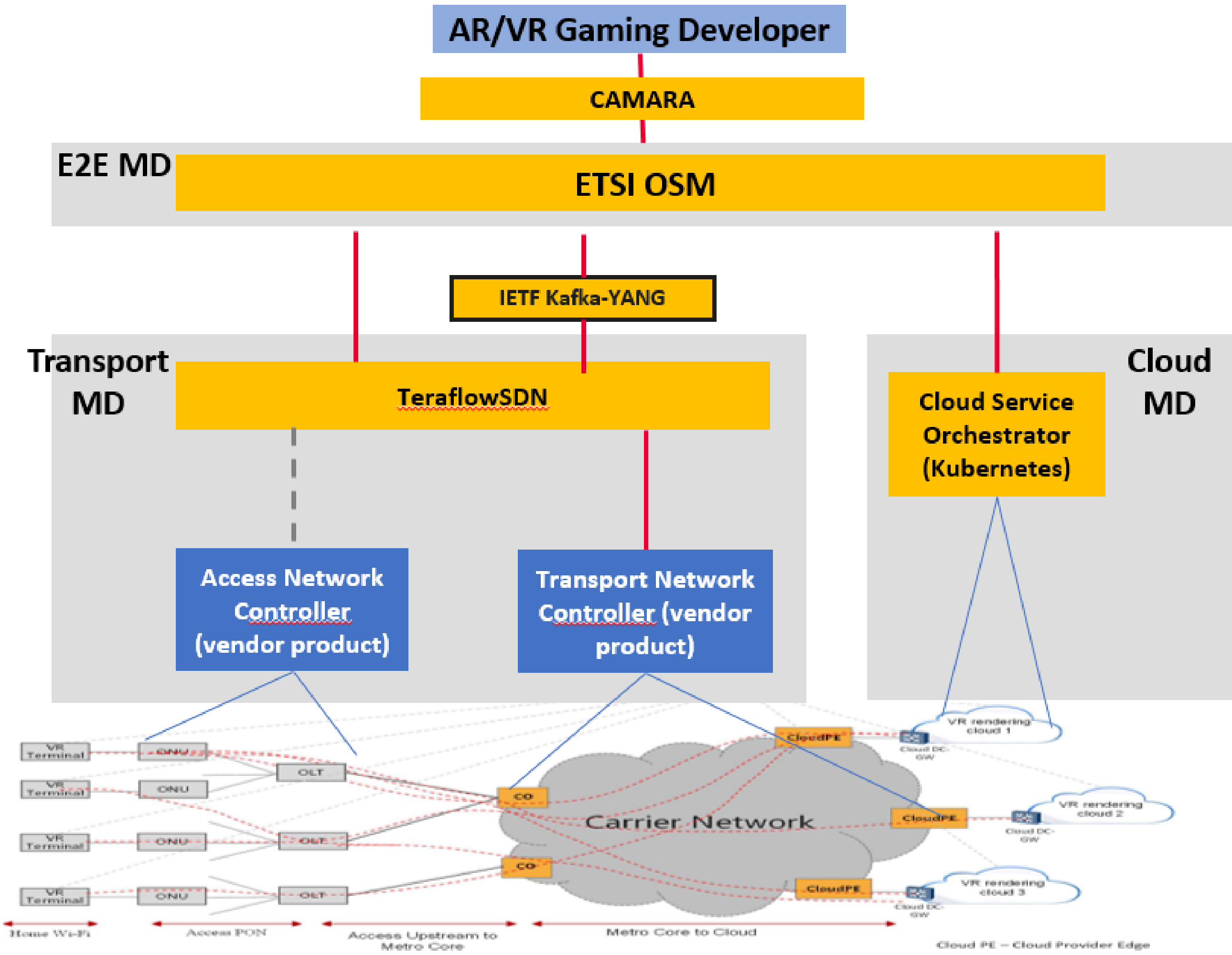

Figure 9: PoC software and hardware setup

The network controllers are controlled and orchestrated by ETSI TFS. The overall architecture of this system follows the ZSM framework. More precisely, the E2E MD, the cross-domain integration fabric, and the Cloud MD are implemented using open-source software. The Transport MD is implemented using both open-source software (i.e. ETSI TFS) and vendor products.

> Note: Access network equipment (ONU, OLT) is managed via the access network controller. However, the controller (vendor device) does not yet provide runtime NBIs. So, in this PoC, we can only use it to manually pre-configure the OLT (prior to the demo).

2.3 ZSM relevance of the PoC

This PoC attempts to implement and demonstrate the Cloud AR/VR use-case on a NaaS platform. Moreover, we try to develop a generic solution (i.e., apart from specific use-cases and technology) that can demonstrate a general framework of NaaS.

To meet this requirement, the ZSM framework is a suitable architectural choice for the solution. Furthermore, the solution contains the following ZSM design constructs:

- **Management Services of the MDs.** In this PoC, we implement three MDs: E2E MD, Transport MD, and Cloud MD. Each MD offers standards-based NBIs and provides the managements services.
- **Integration fabric**: We need to use this construct for interactions between MDs, as well as that between MFs within a MD.

In summary, this PoC develops a NaaS platform which offers services through CAMARA service APIs. The platform uses the ZSM framework as its software architecture. Furthermore, the PoC uses the cloud VR gaming use-case to demonstrate the functionalities of the NaaS platform.

2.4 Future Steps and Developments

Although this PoC intends to develop a NaaS platform, a full scale one with a comprehensive feature set, however, may take multiple phases to develop. This PoC only develops and demonstrates some basic NaaS operations, such as Cloud VR service creation and deletion via CAMARA service APIs. Many work are yet to be done to the platform in the future, including (but not limited to) the following:

- A full reference implementation of the integration fabric. (In this PoC, we only develop a basic service discovery and notification mechanism.)
- Intent negotiation and closed-loop features (relevant to and comply with ZSM 011 and 016)
- Add other MDs (such as RAN MD)
- Use of Digital Twin and AI technology

2.5 Additional information

N/A