Supporting MANOaaS and Heterogenous MANOaaS Deployment within the Zero-touch network and Service Management Framework

Sand L. Correa, Luis M. Contreras, Elton V. Dias, Leonardo R. Furtado, Kaíque M. R. Cunha, Rúben F. Xavier, Leandro A. Freitas, Vinicius da Cunha M. Borges, Antonio Oliveira-Jr, Cristiano Bonato Both, Stuart Clayman and Kleber V. Cardoso

***Abstract*—With the deployment of network slicing in the fifth generation (5G) systems, telecom operators can partition their physical infrastructure into a number of distinct network services. However, the advantages of network slicing come at the price of higher complexity in operating and managing telecom networks. To cope with such complexity, the ETSI Zero-touch network and Service Management (ZSM) framework is designed as a next-generation management system that aims to ideally have all operational processes and tasks executed automatically. ETSI has defined a procedure to deploy the Network-Slice-as-a-Service (NSaaS) scenario using the ZSM reference architecture. Two important use-cases for a more ambitious NSaaS model are MANO-as-a-service (MANOaaS) and Heterogeneous MANO-as-a-Service (H-MANOaaS), where multiple instances of the same (MANOaaS) or different (H-MANOaaS) MANO frameworks are deployed over the same physical substrate. We propose a conceptual model for supporting the never-addressed H-MANOaaS use-case. We also offer a blueprint to integrate the MANOaaS and H-MANOaaS use-cases into the ZSM procedures and mechanisms. We then validate the H-MANOaaS deployment use-case with a proof-of-concept where our proposed solution is instantiated using a real-world slice-as-a-service platform and some relevant implementations of different MANO frameworks.**

***Index Terms*—Network slicing, Management and Orchestration Framework, NFV Infrastructure, Zero-touch network and Service Management Framework**

I. INTRODUCTION

Software Defined Networking (SDN) and Network Function Virtualization (NFV) have transformed the telecom industry, enabling faster and dynamic delivery of services. Network slicing [1] is the next stage of this transition. Through network slicing, different vertical customers (or tenants) can be allocated with resources (compute, storage, and connectivity) from a shared infrastructure in a virtualized fashion, but perceived by tenants as fully dedicated to them. Based on virtualization, offered services can be continuously adapted over time (e.g., service duration) and space (e.g., desired logical topology) to tenants' specific needs. While network slicing increases flexibility and efficiency in telecom networks, it also brings the necessity of enabling service deployment mechanisms, assisting service fulfillment, and providing tools for service assurance. Thus, the advantages of network slicing come at the price of higher complexity in operating and managing telecom networks.

To deal with this complexity, European Telecommunications Standards Institute (ETSI) has designed a reference framework to support zero-touch, full automated End-to-End (E2E) network and service management in next-generation networks, denoted Zero-touch network and Service Management (ZSM) framework [2]. ZSM supports E2E management by defining Management Domains (MDs) as an abstraction to support separation of administration concerns. In ZSM, an E2E MD interacts with individual MDs (i-MDs) to form a hierarchical structure where each i-MD directly manages infrastructure resources within a single network domain, while E2E MD composes MD management services for managing E2E network services.

ETSI has specified procedures to deploy typical E2E network slicing scenarios using ZSM [3]. One of such scenarios involves the Network-Slice-as-a-Service (NSaaS) model, where a tenant is aware of the existence of its Network Slice (NS) instance. In [3], E2E MD is a provider of E2E network slicing-related management services while each i-MD represents an NS subnet (e.g., RAN, Core). An i-MD may also operate an instance of a Management and Orchestration (MANO)[1] framework (e.g., BroadBand Forum Cloud Central Offices (BBF CloudCO) [4] in the Core subnet, or Open RAN (O-RAN) [5] in RAN segment), which is responsible for the life cycle management of i-MD's NFV Infrastructure (NFVI) and Virtualized Network Functions (VNFs) forming the NSs instantiated within i-MD.

An important use-case for a more ambitious NSaaS model is the one where instances of the same MANO framework (e.g., ETSI NFV MANO[2]) are provided per slice so that tenants can manage and orchestrate their slices for a greater level of

[1]MANO refers in a general way to a management and orchestration framework

[2]ETSI NFV MANO denotes a framework to manage network services and VNFs

Sand Correa, Vinicius Borges, and Kleber Cardoso are with Universidade Federal de Goiás, Goiânia, Brazil: sandluz,vcmborges,kleber@ufg.br.

Elton Dias, Leonardo Furtado, Kaique Cunha, Ruben Xavier are with Universidade Federal de Goiás, Goiânia, Brazil: eltonvivot,rodriguesleonardo,kaiquematheus,rubenfranca@discente.ufg.br.

Antonio Oliveira-Jr is with Universidade Federal de Goiás, Goiânia, Brazil and Fraunhofer Portugal AICOS, Porto, Portugal: antoniojr@ufg.br.

Luis M. Contreras is with Telefonica I+D / CTIO, Madrid, Spain.

Stuart Clayman is with the University College London, London, England: s.clayman@ucl.ac.uk.

Leandro is with Instituto Federal de Goiás, Inhumas, Brazil: leandro.freitas@ifg.edu.br.

Cristiano is with the University of Vale do Rio dos Sinos, Brazil:cbboth@unisinos.br.

control over their resources, services, and policies. This model is known as MANO-as-a-Service (MANOaaS), and aspects of its implementation are discussed in [6], [7]. An even more ambitious NSaaS model is a generalization of MANOaaS, where instances of heterogeneous MANO frameworks are provided per slice. Such generalization, which we call Heterogeneous MANOaaS (H-MANOaaS), presents several advantages. Next, our analysis of CAPEX and OPEX reduction is based on features discussed in [8].

From the operator perspective, H-MANOaaS allows one or more sites providing virtual infrastructure (referred to as NFVI-PoPs) to be dynamically partitioned among different MANO frameworks, allowing heterogeneous MANO platforms to co-exist over these sites if operationally required. For example, using H-MANOaaS, the operator can dynamically create two optimized internal slices over the resources of a single cloud domain: one managed by a BBF CloudCO instance and accommodating fixed access services, and the other managed by an ETSI NFV instance and hosting Core Network services. This type of sharing, which cannot be achieved using MANOaaS, fosters better use of NFVI over time and according to the evolution of the services to be offered by an operator, causing potential CAPEX reduction. Although most current Virtualized Infrastructure Managers (VIMs) can support partitioning the resources into isolated zones where multiple MANO platforms can be deployed, this solution is VIM-dependent, constraining the sharing to MANO frameworks that work with the same VIM component. To avoid such limitation, resource partitioning in H-MANOaaS is realized by a functional entity distinct from the VIM. Since H-MANOaaS is VIM-independent, it can operate with multiple virtualization technologies, ranging from lower-level virtualization tools (e.g., Xen, KVM) to container platforms (e,g., Docker, Kubernetes). Consequently, resources can be virtualized/de-virtualized on-demand using various virtualization technologies. This feature increases hardware consolidation providing potential CAPEX reduction. In addition, decoupling the evolution paths of software, hardware, and virtualization infrastructure allow each to be upgraded independently of the other, causing potential OPEX reduction.

From the tenant perspective, H-MANOaaS allows the tenant slice to be deployed with the MANO framework of the tenant's preference. In this case, the MANO framework becomes a slice parameter to be chosen and configured according to the tenant's needs. This flexibility complements the view where optimized slices are on-demand created for a given service. In fact, on-demand deployment of MANO frameworks can be very useful for both tenants and operators. As new verticals are supported by the telecom industry, new functionalities that require management and, possibly, new types of MANO frameworks tend to appear. By allowing MANO frameworks being on-demand provisioned on top of a slice, specific management elements can be deployed dynamically in the correct slice. This solution emphasizes DevOps practices which accelerate service provisioning, leading to potential OPEX reduction.

Despite the advantages of H-MANOaaS, to the best of our knowledge, no previous work has discussed implementation aspects or run-time mechanisms for supporting it. Yet, this model is essential to design complete network slicing scenarios. Motivated by this gap, in this article, we make the following contributions:

- We provide an overview of the MANOaaS and H-MANOaaS models and introduce the first conceptual model to provide H-MANOaaS, underlying differences with the conceptual model for supporting MANOaaS.
- We sketch a blueprint for deploying the MANOaaS and H-MANOaaS use-cases, relying on ZSM procedures and exploiting the recursion property of MDs in ZSM.
- We validate the H-MANOaaS deployment use-case through a proof-of-concept where our proposed solution is instantiated using the NECOS slice-as-a-service platform [9] and relevant implementations of different MANO frameworks.

II. Conceptual Overview of MANOaaS and H-MANOaaS

Third Generation Partnership Project (3GPP) defines the NSaaS model as the scenario where NSs can be offered by the operator to tenants, leaving up to them the possibility to offer their own services and NS Instances (NSIs) to their customers on top of NS services provided by the operator.

ETSI NFV supports the NSaaS model by having each tenant being allocated a quota of NFVI resources tailored to the tenant's service requirements [10]. In this model, the tenant can offer multiple NSIs to its customers within its domain of the allotted quota of resources. However, the MANO services of these various NSIs are performed by a single, centralized ETSI NFV instance managed by the NFVI owner. The reason for this centralized MANO instance lies in the design of ETSI NFV, which is not conceived to share NFVI with other ETSI NFV instances. Indeed, this way of supporting the NSaaS model involves several challenges in the life cycle management of NSs. A major concern is scalability when the number of tenants and slices per tenant increases. Another issue is that tenants have limited autonomy to manage their own NSs by having to rely on a centralized MANO framework.

To overcome these challenges, [6] proposes the MANOaaS model as an extension of the NSaaS model. As illustrated in Fig. 1(a), the core idea of MANOaaS is the provisioning of an NSI and a virtualized abstraction of an NFV MANO framework per tenant. Fig. 1(a) shows a single administrative domain, which can be characterized by one or more NFVI-PoPs. The tenant MANO instance is called t-MANO, while the centralized MANO instance, managed by the NFVI owner, is referred to as c-MANO. The t-MANO instances provide tenants the required autonomy to manage their own resources, services, and policies throughout their respective NSIs. The c-MANO maintains administrative control over the deployed t-MANO instances and, depending on the agreed management level, it can delegate operational control to t-MANO instances, enabling tenants to perform management actions and life cycle operations over their respective NSIs. To implement MANOaaS, the t-MANO stack must be an abstract image of the c-MANO system stack. The t-MANO functional blocks,

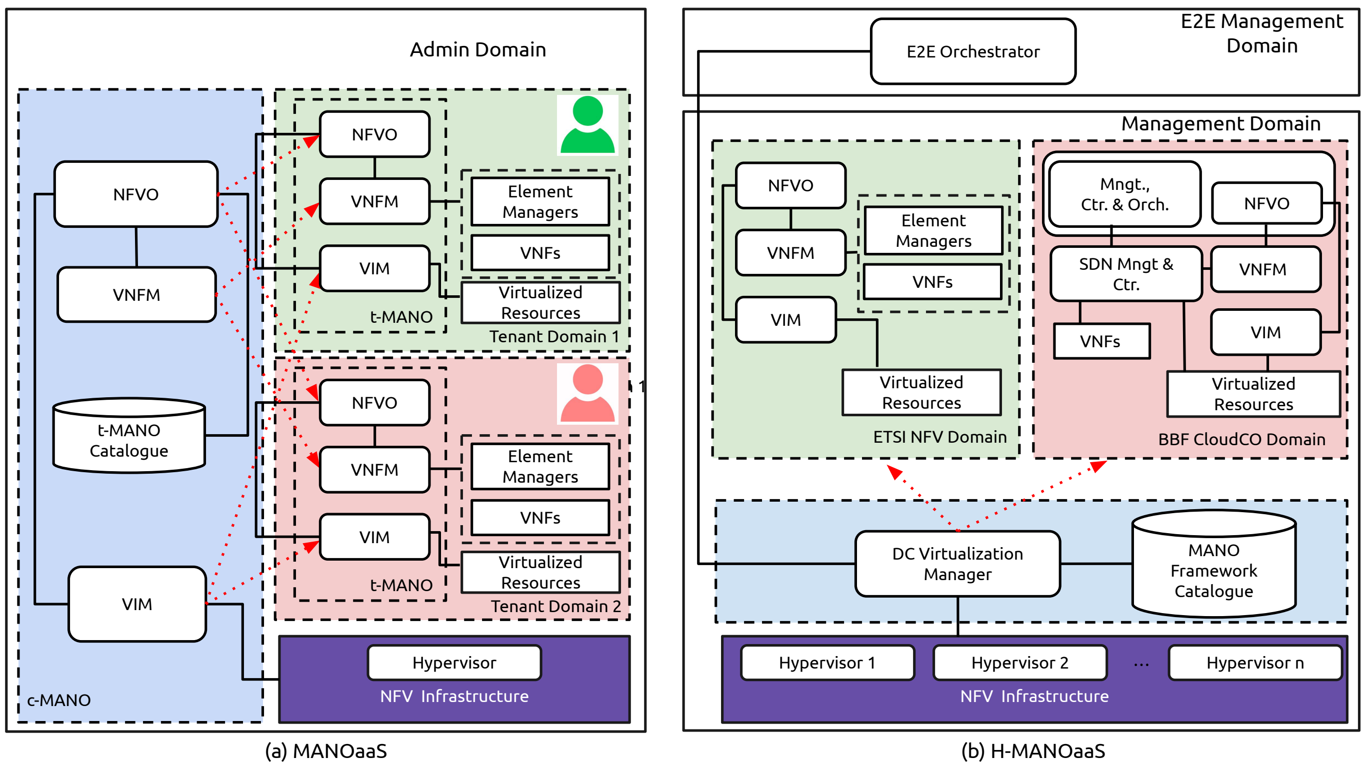


Fig. 1. Conceptual models of (a) the MANOaaS model with extended version of the ETSI NFV framework and (b) the H-MANOaaS model with vanilla version of ETSI NFV and BBF CloudCO frameworks.

e.g., NFV Orchestrator (NFVO), VNF Manager (VNFM), and Virtualized Infrastructure Manager (VIM), are realized as Virtual Management Functions (VMFs) within the tenant domain and implemented as virtualization recipients as virtual machines (VMs) or containers. To maintain administrative control of the t-MANO instance, there is a logical peer relationship between t-MANO functional blocks and the corresponding c-MANO components. Consequently, MANOaaS requires a homogeneous MANO framework for both tenant and centralized MANO instances.

During the last few years, several MANO frameworks have emerged leveraging telecom services on top of an NFVI. These frameworks were conceived targeting different motivations and use-cases. For instance, ETSI NFV designs for general NFV use-cases, ETSI MEC provides for running mobile edge applications, BBF CloudCO focuses on cloudification of operator's Central Offices, and O-RAN designs for running radio stack into VNFs. As illustrated in Fig. 1(b), some frameworks, such as ETSI NFV and BBF CloudCO, can share the same NFVI for better substrate use. As discussed previously, such sharing presents several advantages for operators and tenants. Therefore, there is a need to extend the NsaaS model to other situations rather than just those proposed by 3GPP and MANOaaS. It is particularly important to extrapolate the NSaaS model to the situation where heterogeneous MANO frameworks are created per slice. To fulfill this need, in this work we propose H-MANOaaS. The core concept of H-MANOaaS is the provisioning of an on-demand, independent, and moldable execution environment, on top of a slice.

Fig. 1(b) presents the conceptual model of H-MANOaaS. The figure shows a single administrative domain, characterized by the E2E Management Domain and one Management Domain (NVI-PoP). Similar to MANOaaS, H-MANOaaS uses abstract images of MANO frameworks to realize its concept. However, because H-MANOaaS targets heterogeneous MANO frameworks, its realization requires resources to be partitioned by a functional entity different from the VIM, since VIM interfaces with the instantiated MANO framework components do not necessarily have to match completely. This entity, denoted Data Center Virtualization Manager (DCVM), must exist in each NFVI-PoP. An implementation of the DCVM, named Data Center (DC) Slice Controller, is presented in the NECOS (Novel Enablers for Cloud Slicing) project [9]. NECOS architecture introduces a lightweight NSaaS model for creating, reconfiguring, and decommissioning multi-domain cloud-network slices. One feature of such architecture is an infrastructure resource management model for allocating on-demand VIMs for each NS, realized by the DC Slice Controller running inside a DC and managing the DC's pool of resources that must be allocated to the NS Subnet Instances (NSSIs). DC Slice Controller instantiates virtual resources and deploys an on-demand VIM over them, using abstract images of VIMs. This approach allows existing VIMs to be simultaneously employed in different slices without modification (i.e., vanilla version), assuring proper isolation and offering extra flexibility. While the role of the DCVM has been introduced in [9] through the DC Slice Controller, the latter is focused specifically on VIMs and does not address on-demand

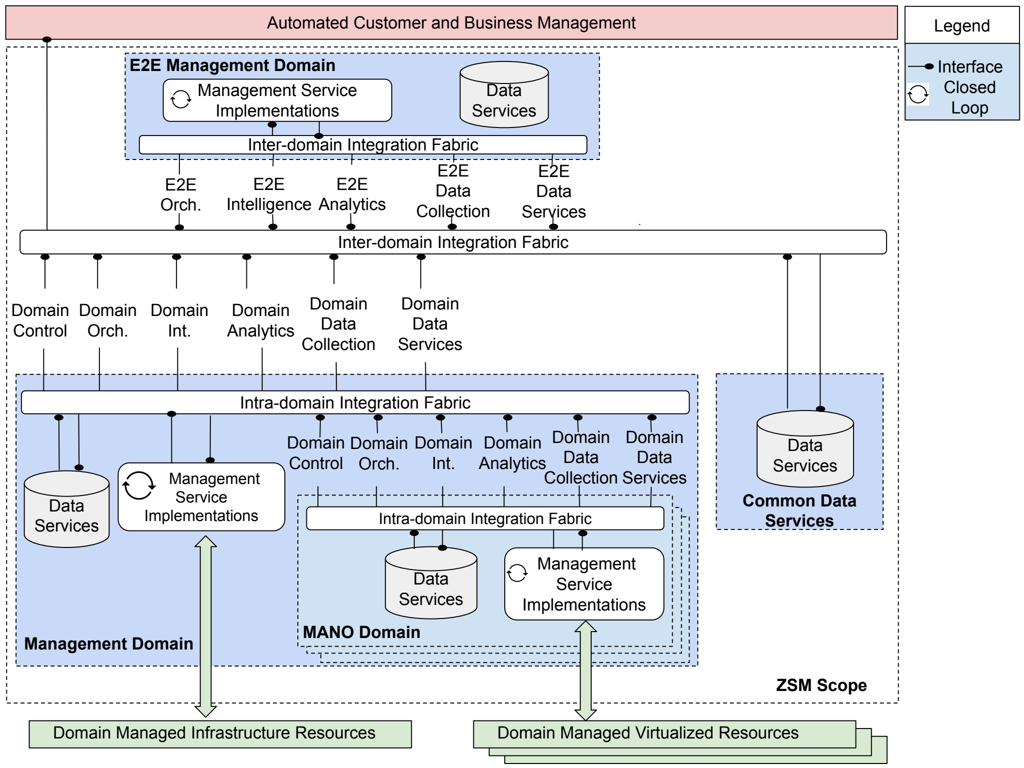


Fig. 2. ZSM framework reference architecture formed by the E2E MD and individual MDs [2]. E2E MD composes the management services for managing E2E network services. Individual MDs directly manage the infrastructure resources within a single network domain.

MANO aspects. Therefore, we extend the VIM on-demand flexibility to deal with on-demand MANO frameworks, as illustrated in Fig. 1(b). This flexibility is achieved by the DCVM provisioning on-demand, and possibly, heterogeneous MANO framework instances per slice.

From an API design perspective, in H-MANOaaS, MANO frameworks are deployed per slice and they can only manage and orchestrate resources allocated to the slice. Slice creation and slice elasticity is carried out by an E2E orchestrator running in the E2E Management Domain and the DCVM in each NFVI-PoP. Thus, the usable control interfaces to accomplish these tasks are defined between the E2E orchestrator and DCVM.

III. ZSM Framework Reference Architecture

The ZSM framework architecture follows service-oriented principles in order to build a service-based framework for inter-domain network and service management. As illustrated in Fig. 2, the framework architecture comprises a set of components, namely, MDs, management services, data services, and integration fabric. These components are designed in a way that the framework architecture meets design principles such as separation of concern, modularity, extensibility, and scalability. Those characteristics are complemented by the use of intent-based interfaces and closed-loop operation with Artificial Intelligence techniques to promote full automation of the management operations.

The ZSM framework architecture comprises MDs to support the separation of management concerns. An MD defines a scope of management delineated by a business, administrative, technological, or other boundaries (e.g., zones in DCs) that comprises management services and their exposure toward external service consumers. In this scope, an MD is responsible for automating the control, orchestration, and assurance of managed resources (e.g., physical and virtual resources) and services. E2E MD is a special MD that orchestrates the management services provided by i-MDs for cross-domain end-to-end management. Decoupling of MDs at different levels (i-MDs and E2E MD) reduces complexity and enables i-MDs to evolve independently from end-to-end management operations. Each MD, including E2E MD, exposes a set of self-contained, loosely-coupled management services (e.g., data collection services, analytic services, domain intelligence services), accessed through standard interfaces. These services are the basic ZSM system modules that can be deployed and scaled independently. Data services are also available in each MD, allowing separate data storage and processing, promoting data sharing between functional components inside MD, and cross-domain data exposure. In particular, data services in Common Data Services can be used by domain and E2E intelligence services to drive domain-level and cross-domain closed-loop automation, respectively. Finally, integration fabric is the mechanism defined in the architecture for supporting flexible communication and interoperation of the management services within each MD (through intra-domain integration fabric) and across different MDs (through inter-domain integration fabric).

ETSI ZSM is not the only standard for cross-domain network and service management. Lifecycle Service Orchestration (LSO) specification [11] offers a framework for cross-domain management. In LSO, Connectivity Services are orchestrated by a Service Provider, who plays the role of E2E MD, and Partner Domains represent i-MDs. Compared to LSO, ZSM is developed emphasizing the integration of control loops employed in individual domains and E2E autonomous management. However, in the scope of this article, we mainly explore the MD concept of ZSM.

Finally, from an API design perspective, in ZSM, the interaction of E2E MD with the i-MD is confined within the interface between the E2E orchestrator and the i-MD controller. The i-MD controller can then use its own methods and interfaces for controlling the MD's resources. H-MANOaaS is in line with this design.

IV. NSaaS Scenario using the ZSM Framework

Basically, the ZSM framework architecture is a two-level hierarchical structure formed by E2E MD and i-MDs. This section first presents how the NSaaS model is deployed using a two-level hierarchical ZSM framework architecture. Then, we present our proposal to enable H-MANOaaS and MANOaaS models as ZSM use-cases using the recursion property of ZSM.

A. Instantiating the NSaaS Model

ETSI describes the procedure to deploy and instantiate the NSaaS scenario using a two-level hierarchical ZSM framework architecture [3]. The general process is illustrated in Fig. 3, and is as follows: tenant directly makes an NSI allocation request to the operator, providing NS-related requirements, e.g., attributes of NS and service profile (message 1). Based on those requirements, E2E MD decides to create a new NSI for this request (message 2). E2E MD first derives NSSI requirements and sends them to the i-MDs involved with slice creation, here labeled as as $MD1, \dots, MDn$ (messages 3.1.a to 3.n.a). The goal of these messages is to check the feasibility of required resources in each i-MD and to make a reservation for them if i-MD can fulfill the request. When all reservations are completed (messages 3.1.b to 3.n.b), E2E MD continues slice allocation (message 4). It sends requests to allocate specific NSSIs to the involved i-MDs (message 5.1 to 5.n). On receiving the request from E2E MD, i-MDs begin the procedures for NSSI allocation (messages 6.1 to 6.n). After completing the allocation of the respective NSSI, each i-MD sends a response back to E2E MD (messages 7.1 to 7.n). The final response for NSI creation is sent back to the tenant (message 8) based on the reply from all involved i-MDs.

In the process described in Fig. 3, each i-MD usually runs a centralized MANO framework instance within its scope. This instance is responsible for creating, managing, and orchestrating all NSSIs instantiated within the i-MD. Thus, each i-MD maps to a MANO domain. Consequently, i-MDs are subject to design principles of the associated MANO framework. Since current MANO frameworks are not originally designed to share NFVI with other MANO framework instances, MANOaaS and H-MANOaaS are not supported by a two-level hierarchical deployment.

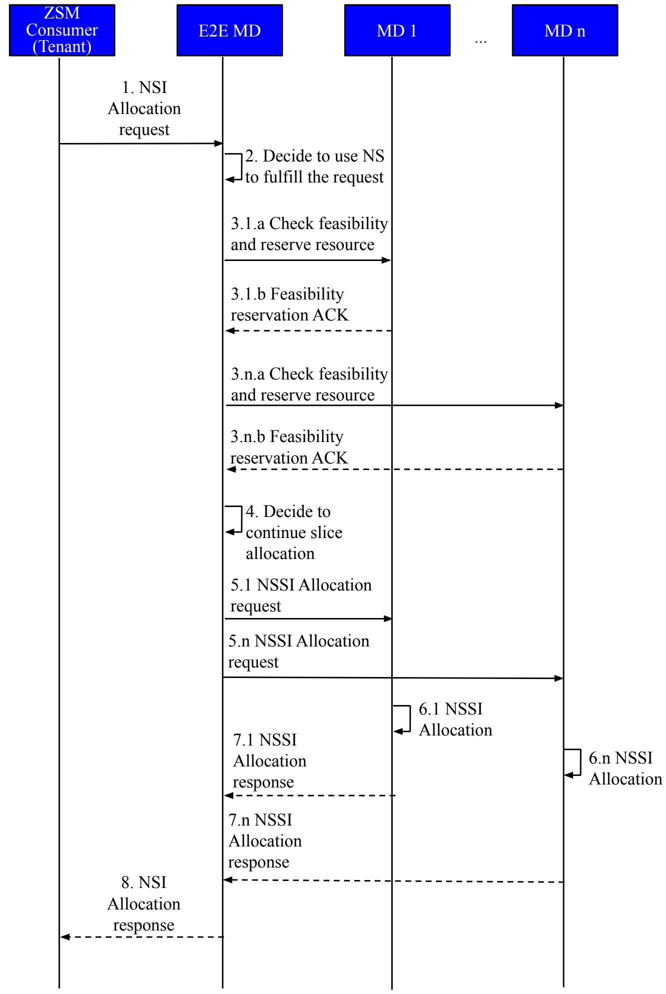


Fig. 3. NSaaS scenario deployment using a two-level hierarchical ZSM framework.

B. Instantiating MANOaaS and H-MANOaaS Models

The ZSM framework architecture can support multiple levels of domain hierarchies that are recursively organized [2]. In this configuration, an MD may be recursively composed of other MDs that still interact according to the ZSM framework. We use this architectural property of ZSM to decouple MANO domains from i-MDs. By recursively creating MANO domains within the scope of an i-MD, MANO operations over a single slice are decoupled from infrastructure domain management. This decoupling allows ZSM to support MANOaaS and H-MANOaaS as deployment scenarios.

Fig. 2 shows a ZSM framework architecture deployment using a three-level hierarchical structure and having such a decoupling. E2E MD can manage and orchestrate NSIs that may span multiple MDs; i-MDs are responsible for managing and partitioning the domain's NFVI (NFVI-Pop) among different NSSIs; while MANO domains within i-MDs are in charge of MANO functions over their slices, which may span multiple i-MDs. Each MANO domain in this context maps to a slice. Such a three-level hierarchical structure avoids overloading a centralized heavyweight MANO system with slicing capabilities, provides good separation of concern, and allows information to be shared and management agreements to be negotiated between different levels of MDs. Since the MANO domain is itself an MD, it comprises management services, data services, integration fabric, and managed resources, as evident from Fig. 2. Management services and data services map to management and orchestration functions provided by the MANO framework. The integration fabric maps to communication mechanisms that allow the MANO framework to expose management services (e.g., domain data collection) to i-MDs for management-level compliance. Finally, managed resources are essentially virtual resources allocated to the slice.

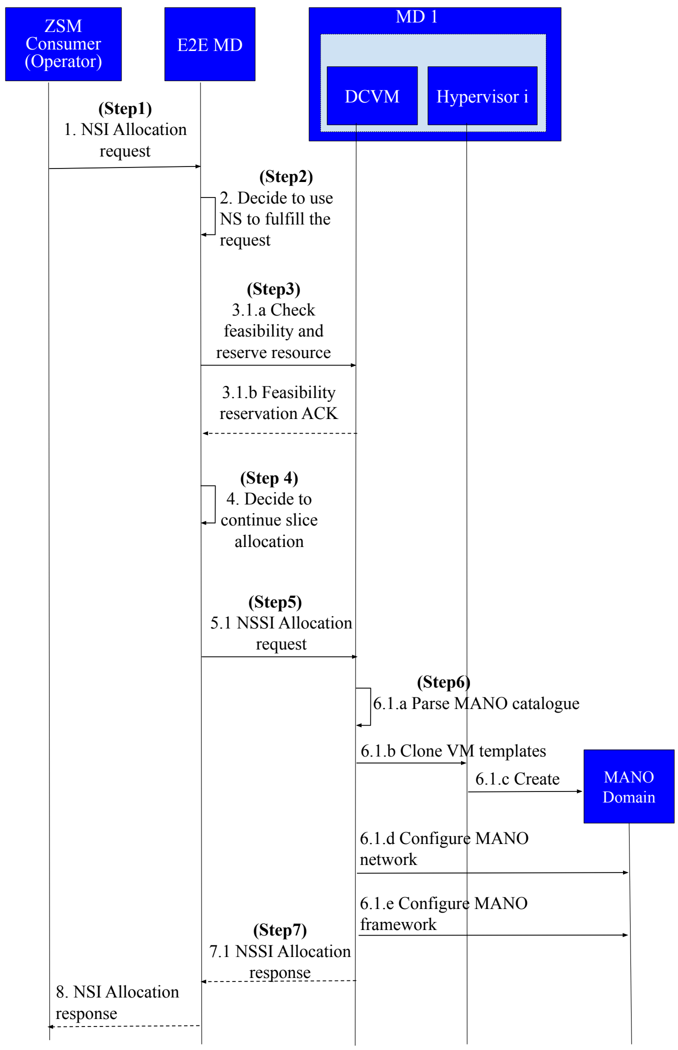


Fig. 4. H-MANOaaS scenario deployment using a three-level hierarchical ZSM framework.

Fig. 4 illustrates the procedure to deploy and instantiate the H-MANOaaS scenario using the three-level hierarchical ZSM framework architecture. Due to space constraints, a single i-MD is considered as an alternative to the procedure shown in Fig. 3. The i-MD runs the DCVM and multiple types of hypervisors. The operator requests allocation of an NSI providing NS-related requirements and the MANO framework to be deployed (message 1). Upon receiving this request, E2E MD decides to create a new NSI (message 2) and contacts i-MD to check if it can fulfill the NSSI request (message 3.1.a). After receiving the E2E MD message, DCVM checks the resource availability. If it can provide the required resources, it reserves them and sends an acknowledgment back (message 3.1.b). After receiving the response from i-MD, E2E MD proceeds with slice allocation (message 4). On receiving a request to allocate an NSSI (message 5.1), DCVM parses the MANO catalog to verify if all relevant images are available (message 6.1.a). After successful verification, DCVM sends a message to the corresponding hypervisor (*i*) to instantiate the MANO domain (messages 6.1.b). The hypervisor then duplicates the virtual disk(s) of the parsed images and instantiates the MANO domain from those disks (message 6.1.c). After the MANO domain is instantiated, DCVM proceeds to configure the network connectivity of the MANO domain (message 6.1.d). This operation involves creating, configuring, and connecting virtual interfaces and switches employed by the instantiated framework. This operation also isolates new instantiated NSSI from others. Finally, DCVM configures the instantiated framework according to instructions defined in the slice descriptor (message 6.1.e). Such configurations correspond to the remaining tasks performed by DCVM necessary to adjust the instantiated framework to make it fully operational. After configuring and validating the new instantiated framework, DCVM returns the NSSI and MANO framework handle to E2E MD (message 7.1), which responds back to the operator (message 8).

The procedure to deploy the MANOaaS scenario is similar to the one illustrated in Fig. 4. However, in the MANOaaS use-case, the i-MD runs an instance of a MANO framework that plays the role of c-MANO instance. The c-MANO exchanges messages with E2E MD, similar to the ones shown in Fig. 4, to instantiate the t-MANO instance.

V. Proof of Concept

An experimental setup was designed, where the procedure described in Fig. 4 is validated using the NECOS slice-as-a-service platform. Based on this setup, we validate the H-MANOaaS scenario deployment where a single i-MD recursively creates MANO MDs and evaluate the suitability of H-MANOaaS. For simplification, only three elements of NECOS are considered: (i) *Slice Tenant* is the end consumer of the slice, representing the owner-operator, requiring the deployment of MANO frameworks internally in some DC; (ii) *Slice Orchestrator* manages and controls all end-to-end slicing capabilities within operator's infrastructure; and (iii) *DC Slice Controller* is the entity playing the role of the DCVM, i.e., in charge of partitioning DC resources and provisioning NSSI. In this setup, we perform the deployment of three real-world implementations of different MANO frameworks, namely Open Source Management and Orchestration (OSM) [12], Central Office Re-architected as a Datacenter (CORD) [13], and Enterprise Application on Lightweight 5G Telco Edge (EALTEdge) [14]. Similar to the scenario in Fig. 4, we consider a single DC. Slice Tenant maps to the ZSM consumer, Slice Orchestrator corresponds to E2E MD services, the DC corresponds to i-MD, and each MANO implementation represents a MANO domain to be instantiated within the i-MD.

Framework implementations. The main components of each MANO implementation used in this setup are shown in Fig. 5(a), Fig. 5(b), and Fig. 5(c). OSM, illustrated in Fig. 5(a), is an open-source implementation of the ETSI NFV framework. OpenCORD is the open-source implementation of the CORD platform, i.e., a representative of Cloud Central Office. Fig. 5(b) illustrates SEBA (SDN-Enabled Broadband Access), a CORD profile focused on supporting residential access and wireless backhaul. Finally, EALTEdge is a blueprint under the 5G MEC system Blueprint Family from Akraino open-source project. As illustrated in Fig. 5(c), EALTEdge implements the main components of the ETSI MEC standard, i.e., MEP, MEPM, and MEO [15].

Testbed setup. The experiment was performed in two dual CPU Intel Xeon Silver, 20 cores and 40 threads, 128 GB of RAM, and 12 TB of disk. Slice Orchestrator and DC Slice Controller are installed in Ubuntu Server 18.04 VMs. The first server runs the Slice Orchestrator VM, while the second server runs the DC Slice Controller and Libvirt server.

VM templates. DC Slice Controller deploys the framework implementations through Libvirt using VM templates (images) created by Xen 4.9.2 hypervisor in a Linux system. OSM comprises a single VM template with a virtual disk size of 40 GB, 8 GB of RAM, and 2 vCPUs. OSM template includes OSM platform and vim-emu, a VIM emulator developed by OSM group. These services are deployed as docker containers in the template. OpenCORD implementation comprises one VM template with a virtual disk size of 100 GB, 32 GB of RAM, and 4 vCPUs. OpenCORD services are deployed as Kubernetes Pods. EALTEdge includes 3 VM templates: jump host (for MEO), center node (for MEPM), and edge node (for MEP). Jump host has a virtual disk size of 40 GB, 4 GB of RAM, and 4 vCPUs; center node has virtual disk size of 240 GB, 8 GB of RAM, and 8 vCPUs; edge node has virtual disk size of 120 GB, 4 GB of RAM, 4 vCPUs. All EALTEdge services are deployed as Kubernetes Pods.

Each framework implementation is deployed in a different slice. Therefore, steps 1 to 7 for slice instantiation, illustrated in Fig. 4, are executed once for each framework implementation. DC Slice Controller performs three main operations when deploying a framework implementation: cloning of VM template(s), network configuration, and framework configuration. We analyze the time taken to create a zero-touch H-MANOaaS deployment to validate the feasibility of the model. Since we are interested in validating the case with heterogeneous MANO frameworks per slice, we omit analysis of other operations related to the slice setup (e.g., asking for slice creation, finding an appropriate DC to instantiate NSI, and contacting the corresponding DC Slice Controller). Consider, however, that in NECOS, it takes on average less than 2 minutes to perform those actions [9]. Fig. 5(d) presents the total time consumed to deploy each implementation framework and the impact of the three operations performed by the DC Slice Controller on the total deployment time. These deployment times are detailed in Table I.

The time taken to clone VM templates is dominant among the performed operations. This time depends on the number of VMs to be cloned and the VM disk size. Since EALTEdge has three VM templates and one of them with a virtual disk size of 240 GB, VM cloning operation in such deployment consumes more time than the same operation in other deployments. OpenCORD has only one VM template, but with a disk size of 100 GB, thus, VM cloning in OpenCORD deployment is also high. OSM has only one VM template with a disk size of 40 GB, therefore, VM cloning in such deployment is the least time-consuming. The times taken to deploy OpenCORD, OSM, and EALTEdge are, respectively, approximately 20 minutes, nearly 2 minutes, and about 34 minutes. These deployment times depend on how much the image templates can be pre-configured. Usually, more configurations in the preparation phase translate into lesser time consumed to deliver a framework instance. Considering that such slices may last for hours, days, or even weeks, we conclude that H-MANOaaS is able to provide timely and fully automated heterogeneous MANO framework instances.

TABLE I
DEPLOYMENT TIMES DETAILED

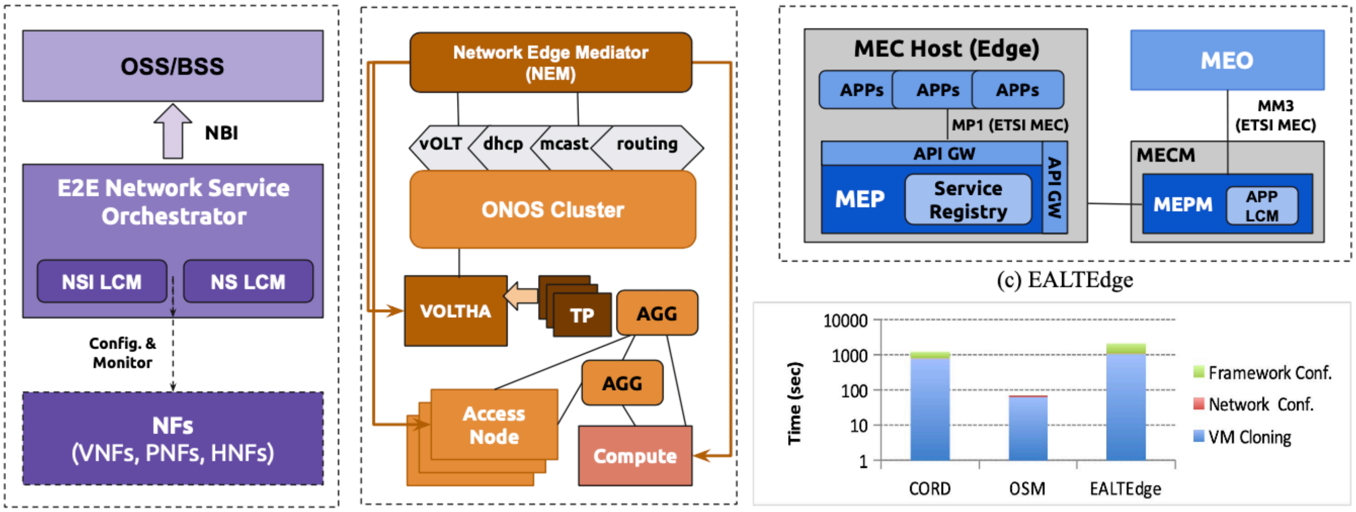
	VM Cloning		Network Conf.		Framework Conf.		Total time
	(s)	(%)	(s)	(%)	(s)	(%)	(s)
CORD	783.6	64.34	5.4	0.44	429	35.22	1218
OSM	63.6	89.84	5.4	7.62	1.8	2.54	70.8
EALTEdge	1056.6	51.20	33	1.60	973.8	47.20	2063.4

VI. Conclusion

In this article, we have advanced the state-of-the-art in providing NSs for the NSaaS model by introducing H-MANOaaS, a generalization of MANOaaS. In addition, we have validated H-MANOaaS use-case with a real-world slice-as-a-service platform and largely adopted open-source MANO framework implementations. MANOaaS and H-MANOaaS are variants of the NSaaS model. Therefore, we argue that such models should be introduced in standardization procedures to design comprehensive network slicing scenarios. An important component of H-MANOaaS is DCVM. Our current implementation of this component allows VNFs to be run directly inside VMs or as containers inside VMs. As a future work, we intend to extend DCVM to support multiple isolation levels, including bare metal. Also, we intend to perform a deeper performance evaluation.

Acknowledgment

This work was partially funded by the European Commission through the NECOS project (Grant Agreement No 777067). Luis M. Contreras has been also partially funded through EU Horizon Europe NEMO project (Grant Agreement No 101070118). Stuart Clayman is partially funded by a Huawei grant. Sand Correa, Kleber Cardoso, and Cristiano Bonato Both are partially funded by the Brazilian Ministry of Science, Technology, and Innovation (MCTIC)/Brazilian Internet Steering Committee (CGI.br)/São Paulo Research Foundation (FAPESP) through the Project entitled Slicing Future Internet Infrastructures (SFI2) under Grant 2018/23097-3 and Projeto Brasil 6G funded by the Ministry of Science,



(a) OSM (b) OpenCORD (c) EALTEdge (d) Deployment Times

Fig. 5. Details of proof-of-concepts. MANO framework implementations used in our experiments: (a) OSM, (b) OpenCORD, (c) EALTEdge. (d) Time to deploy each framework implmentation.

Technology and Innovation (MCTI) and the National Research Network (RNP).

References

[1] A. A. Barakabitze, A. Ahmad, R. Mijumbi, and A. Hines, "5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges," *Computer Networks*, vol. 167, p. 106984, 2020.

[2] ETSI, "Zero-touch network and Service Management (ZSM); Reference Architecture, ETSI GS ZSM 002 V1.1.1," Aug. 2019.

[3] ——, "Zero-touch network and Service Management (ZSM); End-to-end management and orchestration of network slicing, ETSI GS ZSM 003 V1.1.1," Jun. 2019.

[4] Broadband Forum, "Cloud Central Office Reference Architectural Framework, TR-384," Jan. 2018.

[5] O-RAN Alliance, "O-RAN Architecture Description 3.0," Nov. 2020.

[6] F. Z. Yousaf, V. Sciancalepore, M. Liebsch, and X. Costa-Perez, "MANOaaS: A Multi-Tenant NFV MANO for 5G Network Slices," *IEEE Communications Magazine*, vol. 57, no. 5, pp. 103–109, 2019.

[7] ETSI, "Network Functions Virtualisation (NFV); Evolution and Ecosystem; Report on Multi-tenancy in NFV Release 4, ETSI GR NFV-EVE 0018 V0.0.19," Oct. 2022.

[8] E. Hernandez-Valencia, S. Izzo, and B. Polonsky, "How will NFV/SDN transform service provider opex?" *IEEE Network*, vol. 29, no. 3, pp. 60–67, 2015.

[9] S. Clayman, A. Neto, F. Verdi, S. Correa, S. Sampaio, I. Sakelariou, L. Mamatas, R. Pasquini, K. Cardoso, F. Tusa, C. Rothenberg, and J. Serrat, "The NECOS Approach to End-to-End Cloud-Network Slicing as a Service," *IEEE Communications Magazine*, vol. 59, no. 3, pp. 91–97, 2021.

[10] ETSI, "Network Functions Virtualisation (NFV); Management and Orchestration, ETSI GS NFV-MAN 001 V1.1.1," Dec. 2014.

[11] MEF, "Lifecycle Service Orchestration (LSO): Reference Architecture and Framework, Service Operations Specification MEF 55," Mar. 2016.

[12] OSM, "Open Source MANO," https://osm.etsi.org/, 2021, Accessed date: 20/09/2021.

[13] ONF, "CORD," https://opennetworking.org/cord/, 2021, Accessed date: 20/09/2021.

[14] LFEDGE, "Enterprise Application on Lightweight 5G Telco Edge (EALTEdge)," https://www.lfedge.org/projects/akraino/release-4-2/enterprise-applications-on-lightweight-5g-telco-edge/, 2021, Accessed date: 20/09/2021.

[15] ETSI, "Multi-access Edge Computing (MEC); Framework and Reference Architecture, ETSI GS MEC 003 V2.2.1," Dec. 2020.

Biographies

Sand L. Correa (sandluz@ufg.br) is a Professor at Instituto de Informática (INF) at Universidade Federal de Goiás, Brazil.

Luis M. Contreras (luismiguel.contrerasmurillo@telefonica.com), M.Sc. in Telecommunications from UPM (1997), M. Sc. in Telematics jointly from UC3M and UPC (2010), and Ph.D. in Telematics from UC3M (2021) works as Technology Expert at Telefónica I+D / CTIO in Spain. Before that he worked in Alcatel and Orange.

Elton V. Dias (eltonvivot@discente.ufg.br) is an undergraduate student at Instituto de Informática (INF) at Universidade Federal de Goiás, Brazil.

Leonardo R. Furtado (leonardodd128@gmail.com) is a Software Engineer and external contributor.

Kaíque M. R. Cunha (kaiquematheus@discente.ufg.br) is a Master student at Instituto de Informática (INF) at Universidade Federal de Goiás, Brazil.

Rúben F. Xavier (rubenfranca@discente.ufg.br) is a Master student at Instituto de Informática (INF) at Universidade Federal de Goiás, Brazil.

Leandro A. Freitas (leandro.freitas@ifg.edu.br) is a professor of network computing distributed systems at Federal Institute of Goiás, Brazil.

Vinicius da Cunha M. Borges (vcmborges@ufg.br) is a Professor at Instituto de Informática (INF) at Universidade Federal de Goiás, Brazil.

Antonio Oliveira-Jr (antoniojr@ufg.br) is a Professor at Instituto de Informática (INF) at Universidade Federal de Goiás, Brazil. He is also an Invited Senior Scientist with Fraunhofer Portugal AICOS.

Cristiano Bonato Both (cbboth@unisinos.br) is a professor of the Applied Computing Graduate Program at the University of Vale do Rio dos Sinos (UNISINOS), Brazil.

Stuart Clayman (s.clayman@ucl.ac.uk) received his PhD in Computer Science from University College London in 1994. He is currently a Principal Research Fellow at UCL EEE department. He co-authored over 75 conference and journal papers.

Kleber V. Cardoso (kleber@ufg.br) is a Professor at Instituto de Informática (INF) at Universidade Federal de Goiás, Brazil.

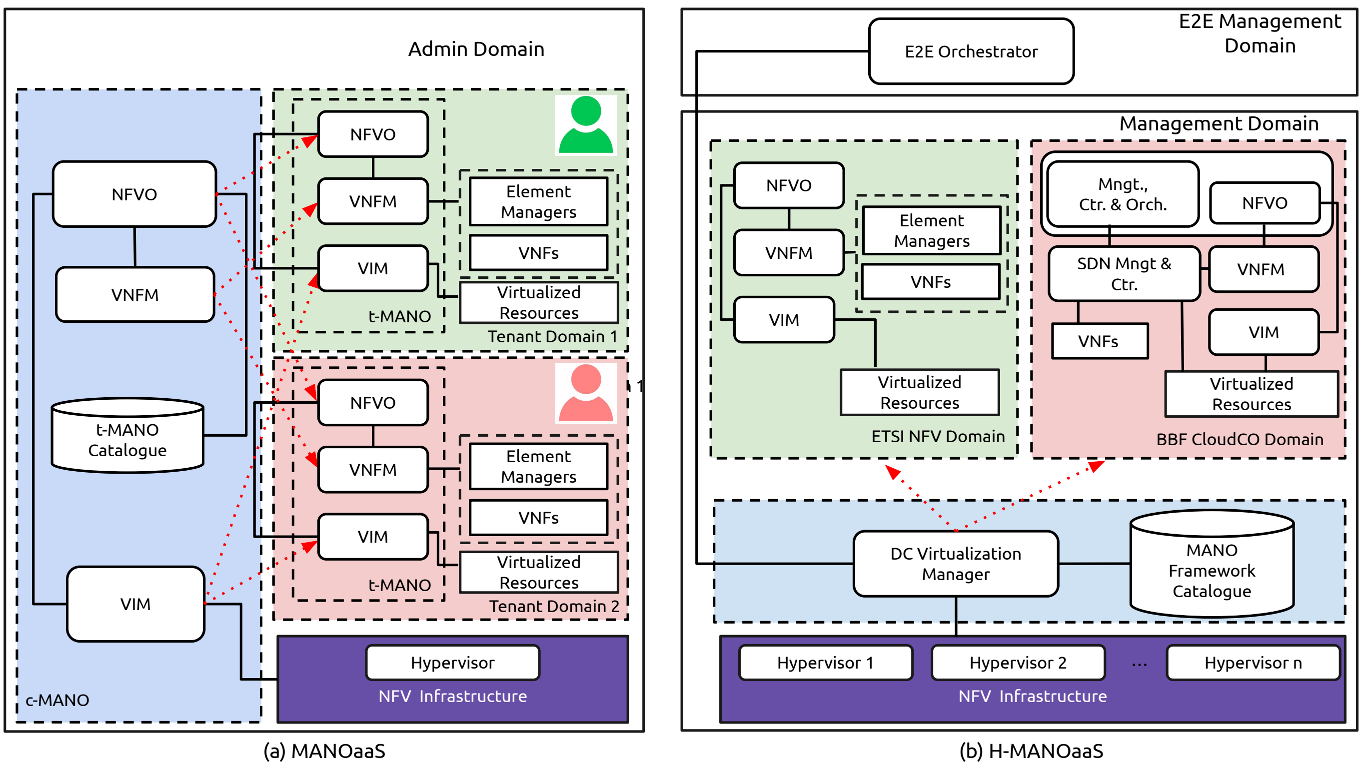


Fig. 1. Conceptual models of (a) the MANOaaS model with extended version of the ETSI NFV framework and (b) the H-MANOaaS model with vanilla version of ETSI NFV and BBF CloudCO frameworks.

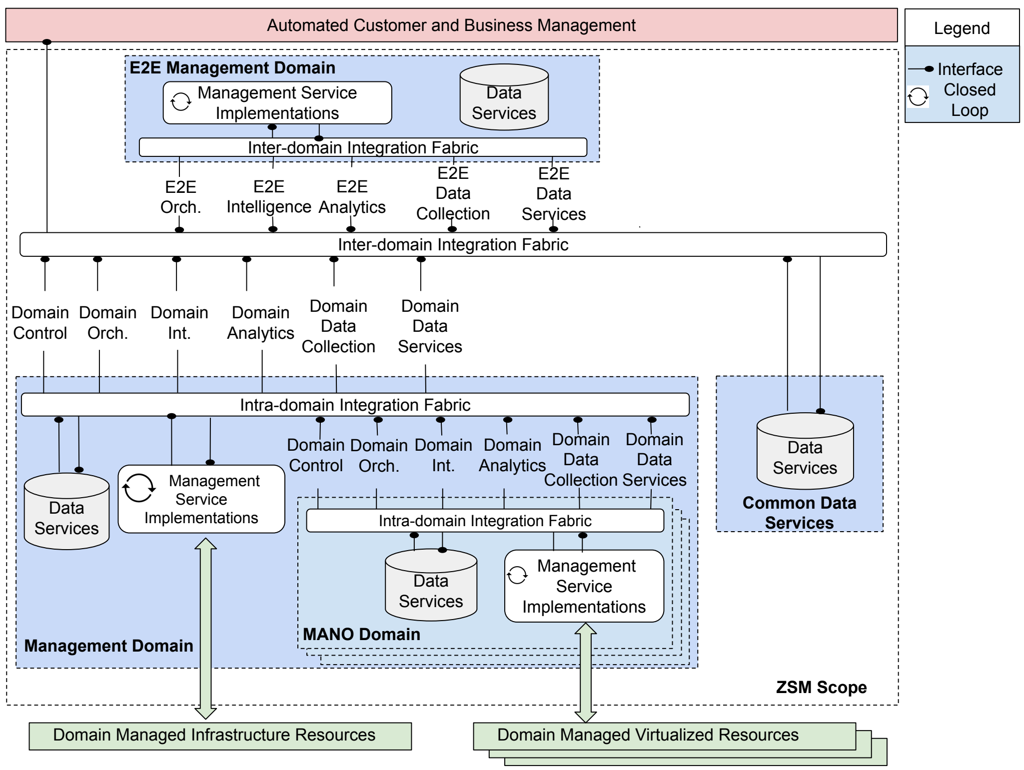


Fig. 2. ZSM framework reference architecture formed by the E2E MD and individual MDs [2]. E2E MD composes the management services for managing E2E network services. Individual MDs directly manage the infrastructure resources within a single network domain.

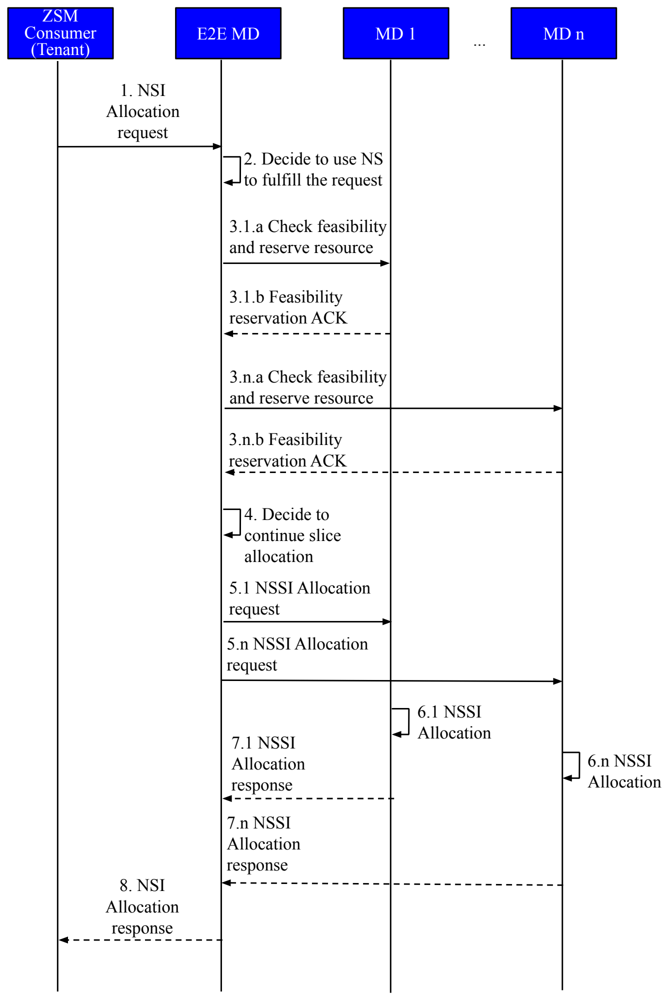


Fig. 3. NSaaS scenario deployment using a two-level hierarchical ZSM framework.

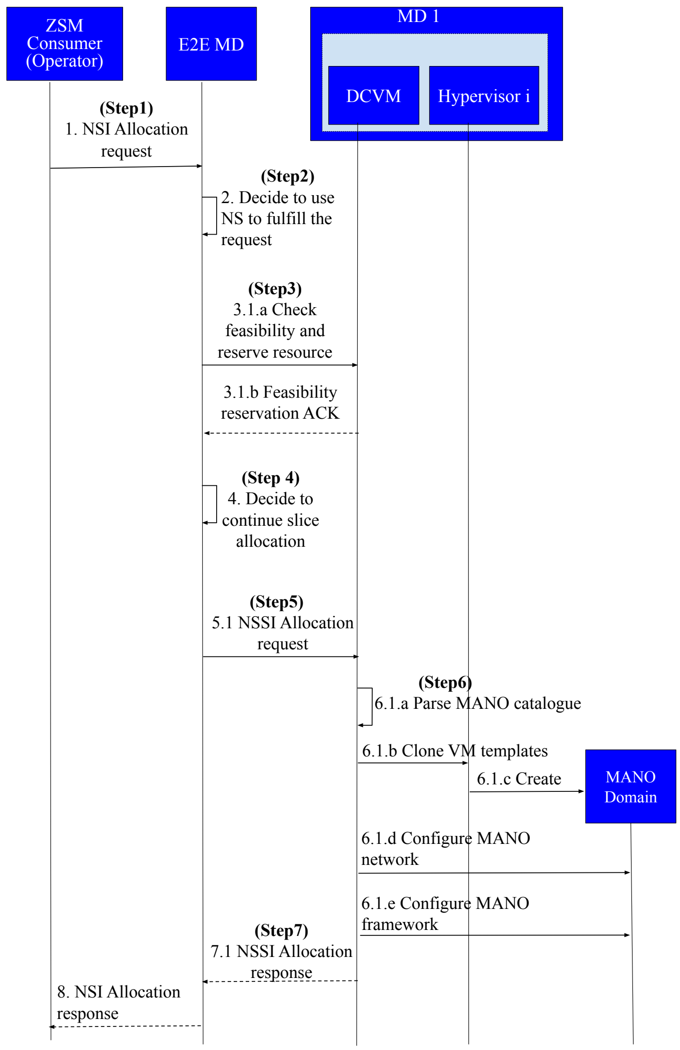
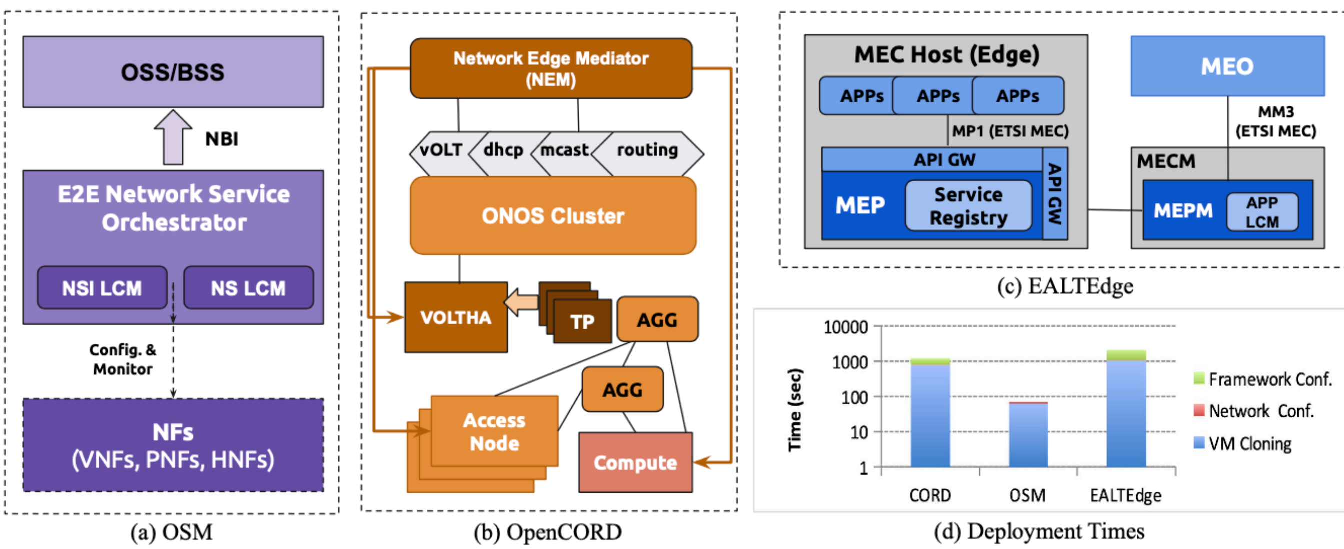


Fig. 4. H-MANOaaS scenario deployment using a three-level hierarchical ZSM framework.



(a) OSM (b) OpenCORD (c) EALTEdge (d) Deployment Times

Fig. 5. Details of proof-of-concepts. MANO framework implementations used in our experiments: (a) OSM, (b) OpenCORD, (c) EALTEdge. (d) Time to deploy each framework implmentation.