This is a postprint version of the following published document:

Yousaf, F.Z., Sciancalepore, V., Liebsch, M. y Costa-Pérez, X. (2019). MANOaaS: A Multi-Tenant NFV MANO for 5G Network Slices. *IEEE Communications Magazine*, 57 (5), pp.103-109.

DOI: https://doi.org/10.1109/MCOM.2019.1800898

© 2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

MANOaaS: A Multi-Tenant NFV MANO for 5G Network Slices

Faqir Zarrar Yousaf, Vincenzo Sciancalepore, Marco Liebsch, and Xavier Costa-Perez

This article sheds light on potential benefits and implementation aspects when the MANO framework is abstracted into customized and distributed MANO instances, thereby empowering the MANO-as-a-service (MANOaaS) paradigm. In particular, such distributed instances are provided to different network tenants for a greater level of control on requested network slice(s).

Abstract

The dramatic densification of connected mobile devices and the expected use cases from the vertical industry demand an innovative network design that meets upcoming stringent requirements. The adoption and harmonized integration of novel concepts, such as network functions virtualization and network programmability, enables the system to master the high expectation — from the fifth generation communication network in support of flexibility — to provide tailored and mutually isolated network slices, high performance, agility, and automation. This effectively involves a number of technical challenges for managing and orchestrating physical and virtualized slice resources by means of an advanced management and orchestration (MANO) system. This article sheds light on potential benefits and implementation aspects when the MANO framework is abstracted into customized and distributed MANO instances, thereby empowering the MANO-as-a-service (MANOaaS) paradigm. In particular, such distributed instances are provided to different network tenants for a greater level of control on requested network slice(s). The notion of management level agreements in the context of MANOaaS is introduced as well as differentiated per tenant while being embedded into the proposed architecture. We also position the proposed MANOaaS concept and associated extensions to the MANO reference architecture from the viewpoint of standardization bodies and ongoing open source projects.

Introduction

The fifth generation (5G) network systems and platforms are envisaged to provide an extended set of services provisioned toward a wide variety of heterogeneous service verticals (e.g., automotive, industry, smart city, e-Health, logistics) generating ultra-high traffic density with differentiated but stringent performance, security, and reliability requirements. This involves the challenge of managing diverse and large-scale (isolated) services from heterogeneous verticals within the respective quality bounds (namely service level agreements, SLAs) over a shared infrastructure: the network slicing paradigm [1].

The core concept of network slicing enables the provisioning of virtualized but mutually isolated customized networks over a shared virtualized infrastructure platform. This is illustrated in Fig.. 1 where three network slice instances owned by different service verticals, also referred to as tenants, are deployed over a shared virtualized infrastructure. Each slice instance is composed of virtual network functions (VNFs) interconnected by virtual links (VLs) in the order specified by a VNF forwarding graph (VNFFG) to deliver a specific network service, including virtual evolved packet core (vEPC), virtual radio access network (vRAN), and so on.

Infrastructure resources (e.g., compute, network, storage) are automatically abstracted and dynamically allocated to each slice. The network services offered by the slices are consumed by higher-level (application) service instances pertaining to specific verticals. Therefore, multiple application services can share the services of a single network slice [3], or multiple slices can be utilized by a single service instance. A credible network slice management and orchestration (MANO) system is required to manage the infrastructure resources, network slices, and service instance(s) to ensure reliable service delivery within the quality of service (QoS) bounds [2].

In this regard, the European Telecommunications Institute (ETSI) Network Functions Virtualization (NFV) has developed an NFV-MANO framework [4] for the life cycle management (LCM) of NFV infrastructure (NFVI) resources (compute, network, storage) and the VNFs/VLs forming network slice or network service (NS) instance(s).

The NFVI owner can support multiple tenants where each tenant is allocated a quota of resources tailored to the tenants' service requirements. The tenant domain is characterized by this allotted resource quota. The tenants can instantiate and offer multiple NS instances to external customers within its domain of allotted quota of resources. As part of its service, the NFVI owner provides the MANO services to the various NSs belonging to different tenants. However, the current design of the NFV-MANO framework aims toward providing centralized management of NSs for multiple tenants. Such a centralized approach has inherent performance issues and management challenges. The focus of this article is on a distributed MANO approach, which enables more flexibility and scalability properties.

The authors are with NEC Laboratories Europe.

Slice Management: Challenges and Gaps

A centralized approach for the provisioning and runtime management of slices involves a discrete number of challenges on the LCM of network slices, the services running on top of them, and the resources orchestration associated with such network slices [5]. One of the core issues is the scalability when the number of tenants and the slices per tenant increase. This in turn increases the load (e.g., monitoring/processing load) on the central MANO instance leading to delayed responses in deriving and executing MANO actions to individual tenants' slices [6]. Another fundamental issue is reliability with a centralized management posing as a single point of failure. Besides scalability and reliability issues, having to rely on a centralized MANO system limits the autonomy of the tenants in implementing their own policies and/or rolling out new services or managing existing services. These issues are compounded in cases where the tenant services are running on remote sites or span multiple sites such as micro data centers or edge domains. Such sites providing virtual infrastructure are also referred to as NFVI-PoP.

Therefore, distributing MANO capabilities is required to address such issues. While the literature provides interesting works in the area of distributed management, in the specific context of NFV-MANO there are few proposals, not comprehensive enough to tackle the entire problem spectrum. For instance, [12] proposes the management of network slices in a multi-domain (i.e., multiple NFVI-PoP) environment, where the slice resources are managed by an overarching MANO system. This approach can distribute the management of NFVI resources and VNFs to individual domains, but the end-to-end orchestration is still performed centrally. A more distributed option is highlighted in [9], where NFVI-PoPs have their own (single) dedicated NFV-MANO instances inter-coordinated in a peer-to-peer fashion via the NFVO entity. This approach inherits the delay issues due to the peer-to-peer communication. A similar approach is adopted by [10], which leverages on the container technology to implement a lightweight MANO stack for resource starved individual edge NFVIs. These proposals support partial distribution and lack the granularity of providing MANO services to individual tenants. In terms of supporting multi-tenancy, [7] proposes an architecture where fully autonomous NFV-MANO systems are deployed for each tenant, leaving the full coordination of such NFV-MANO instances to an overarching inter-slice resource broker entity that manages multi-domain network slices. Also, [8] proposes a distributed approach but solely at the level of the virtual infrastructure manager (VIM) in the view of end-to-end slicing with resources being allocated to the slice in multiple distributed data centers. Instead of relying on a single VIM for the management of infrastructure resource in a data center, an instance of VIM is created on demand for each slice and exposed to the slice's owner. This enables more flexibility to apply custom strategies and policies, while isolation in infrastructure control between multiple tenants is intrinsic with that approach. However, the proposal does not address avoidance or resolution of conflicts when multiple VIMs share physical resources. Also, on-demand MANO aspects are not addressed in [8].

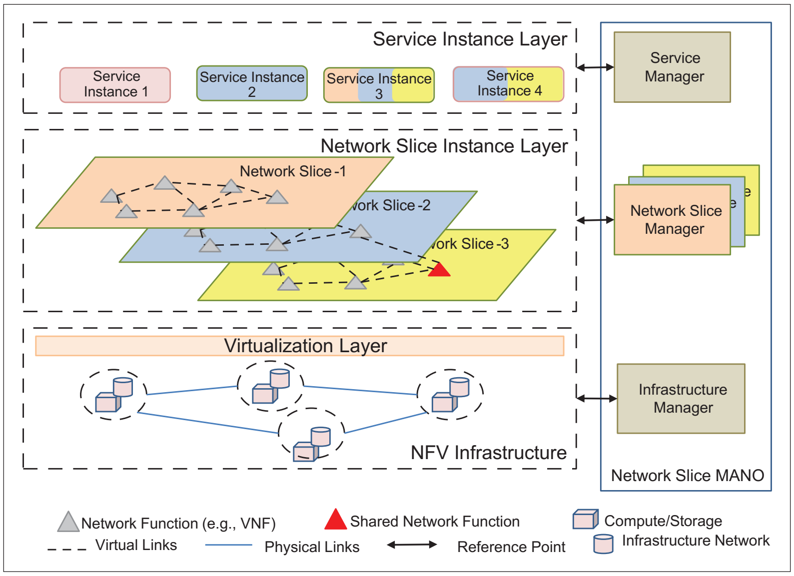

Figure 1. Overview of the concept of slicing management.

The above cited approaches have some degree of centralization and thus do not address the highlighted challenges in their entirety. They also lack the dynamism and flexibility that are needed to provide managed autonomy to tenants toward managing their own slice resources. In light of these issues and gaps, we present our novel solution of MANOaaS that not only provides each tenant the ability to manage its own slices by realizing virtualized abstraction of per-tenant MANO instances, but also has the distinguishing feature of managing the tenants' autonomy through negotiation and enforcement of management level agreements (MLA).

MANOaaS: Conceptual Overview

The MANOaaS concept is proposed as an extension of the ETSI NFV-MANO model [4]. The core concept is the provisioning of a virtualized abstraction of a NFV-MANO system to the respective tenants, which we refer to as tenant MANO (t-MANO). In cloud terms the tenants are said to be provided MANOaaS.

A t-MANO instance shall provide the tenants with the required autonomy to manage and orchestrate their own resources, services, and policies. The central MANO (c-MANO) system maintains administrative control over the deployed t-MANO instances, but, depending on agreed MLAs, the operational control of the c-MANO is offloaded/delegated to t-MANO instances. The MLA is negotiated between the tenant and the c-MANO provider, and determines the scope of the delegated operational capabilities of c-MANO providing either a full or partial set of the features, capabilities, and services. This enables the tenant to exercise MANO functions over its respective resource slices (rSlices) and service slices (sSlices) with minimum reliance on the c-MANO stack. An sSlice is essentially the same as a network slice but within a tenant's domain. An rSlice is the dimensioning of the infrastrucure virtual resources allotted to a tenant within its domain.

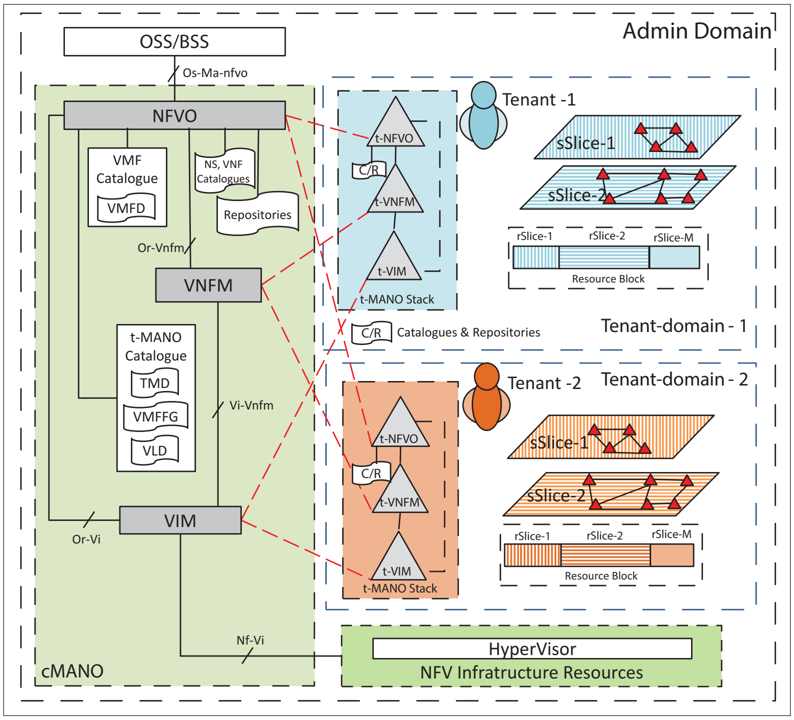

Figure 2. Conceptual model of the MANOaaS paradigm.

However, the c-MANO has full administrative rights over the respective t-MANO stack: it monitors the t-MANO for MLA compliance, provides services, features, and capabilities to the tenants that are outside the negotiated MLA bounds, and, under specific situations, it overrides the t-MANOs decision on actions.

Figure 2 shows the conceptual model of the MANOaaS paradigm. As can be seen, the c-MANO is an extended version of the NFV-MANO system [4] having the following essential functional blocks:

- The VIM for the management of NFVI resources
- The VNF manager (VNFM) for the LCM of the VNF(s) that are deployed and instantiated over the NFVI
- NFV orchestrator (NFVO) for the service and resource management of the network services that are formed by chaining various VNFs and characterized by the VNF forwarding graph

The three functional blocks interact with each other using standard interfaces to provide LCM of virtualized resources belonging to different tenants. LCM actions include instantiation, migration, scale-in/out/up/down, update/upgrade, and deletion of VNF/NS instances. These functional blocks are extended to manage the t-MANO instances, as explained later.

Moreover, the c-MANO extends the NFV-MANO with a virtual management function (VMF) catalogue and t-MANO catalogue containing t-MANO related descriptor files in addition to the standard catalogues with descriptor files, that is, the VNF descriptor (VNFD) file and NS descriptor (NSD) file, which are deployment templates specifying the operational, functional, resource, performance, and policy requirements of the VNFs and NSs, respectively. These new catalogues and descriptor files are explained later.

Figure 2 also shows two tenants, Tenant-1 (T1) and Tenant-2 (T2), which define two logical tenant domains within the same administrative domain, characterized by one or more NFVI-PoPs. Based on their service requirements, T1 and T2 request NFVI resource blocks of specific flavor (i.e., type and amount of specific resources) from the NFVI owner who is also the c-MANO provider. After the resource blocks allotment, the tenants then request the provisioning of a t-MANO system stack. The requested t-MANO instances operate with an agreed level of autonomy and independent of the c-MANO, and hence are potentially not bound to a specific location and able to be instantiated in a suitable NFVI-PoP within the provider's domain. Deployment location selection can be based on multiple factors, such as low-delay interactions/coordination with the c-MANO, for instance, with states and functions still associated with the c-MANO. It may also consider the proximity of the VNFs/NSs for low-delay t-MANO operations on local instances with the advantage of autonomous operations by the t-MANO as per the MLA description. The c-MANO monitors and enforces the t-MANO operations within the agreed MLA bounds. The MLA negotiation shall mainly involve what services/features/capabilities of the c-MANO system the t-MANO instances will have access rights to and the access levels. The tenants' request for resource blocks, t-MANO stacks, and MLA negotiation can be executed via the operation support system (OSS)/base station subsystem (BSS) or ad hoc graphic user interface (GUI)-based portal.

Once the t-MANO system stacks are deployed and instantiated for the respective tenants (the process details can be found later in this article), they are able to dimension (or slice) their respective allocated resource blocks into rSlice(s), and then create, deploy, configure, and instantiate their respective sSlice(s) without involving the c-MANO. Thus, T1 and T2 will have autonomy in performing fault, configuration, accounting, performance, and security (FCAPS) management and LCM operations and orchestration actions over their respective rSlice/sSlice instance(s) as well as for implementing their own policies without involving the c-MANO, whereas the degree of autonomy is efficiently bounded by the defined MLA contract.

As mentioned above, the t-MANO stack is an abstract image of the c-MANO system stack where the MANO functional blocks are realized, similar to VNFs, as VMF instances, and the reference points between the t-MANO VMF instances (i.e., t-VIM, t-VNFM, and t-NFVO) are realized over virtual links (VLs) within the tenant domain, as depicted in Fig. 2. The c-MANO still maintains administrative and management control of the t-MANO service instances, and there is a logical peer relationship between the t-MANO VMF instances and the corresponding c-MANO functional blocks. This peer relationship, depicted as red dotted lines in Fig. 2, enables the c-MANO to monitor the t-MANO for MLA compliance and to extend services, features, and capabilities to the tenants that are outside their respective MLA bounds.

If a tenant instantiates a slice that expands through the NFVI-PoPs of multiple NFVI provid-

Primary key	Secondary key	Parameter	Description
t mano id	ns id	tenant_id	The tenant id to which the t-MANO belongs
		tnfvo-id	The id of the t-NFVO component of the t-MANO stack.
		tvnfm-id	The id of the t-VNFM component of the t-MANO stack.
		tvim-id	The id of the t-VIM component of the t-MANO stack.
		time_to_live	The duration of the t-MANO instance
		resource_flavor	The resources (type & amount) for t-MANO instance
		resource_id_list	The list of resources and amount assigned for t-MANO
		domain_id	The domain where t-MANO instance is deployed
		do_auto_scaling	Permission to perform scaling operations
		do_auto_healing	Permission to perform healing operations
		do_migration	Permission to perform migration operations
		do_update	Permission to perform update operations on VNFs/VNFCs
		vmffg	Pointer to VMFFG
		scale_policy	Scaling policy
		migration_policy	Migration policy
		healing_policy	Healing policy
		mano_event_list	List of events executed by t-MANO
		allow_recursion	Permission for a t-MANO stack to instantiate further t-MANO stacks i.e., recursively.
		tmano_instance_list	The list of t-MANO instances recursively instantiated
		decision_profile	Indicating c-MANOs control profile for overriding t-MANOs decisions. Can be Proactive, Reactive or Autonomous, the latter indicating complete autonomy to t-MANO.
		tmano_state_info	Carries state information of the t-MANO system

Table 1. Example of negotiation parameters as part of Management Level Agreement (MLA).

Our approach focuses on the realization of the t-MANO stack by deploying its individual VMF components (i.e., t-NFVO. t-VNFM and t-VIM) in a similar fashion as a VNF, where the t-MANO VMF instances maintain peer relationship with the respective c-MANO functional blocks either over an IP-based network or via some RPC-based method.

ers, MANO operations are typically handled by each provider's c-MANO system, which coordinates with other providers' c-MANOs through well-defined interfaces at the orchestration level (e.g., the Or-Or interface [9]). An example of a hierarchical MANO system comprising multi-domain environments can be found in [13]. In support of MANOaaS for such cross-domain deployments, each provider's c-MANO shall instantiate a t-MANO instance for the associated NFVI-PoP in its domain. The inter-coordination between the tenant's multiple t-MANO instances is accomplished by direct interfaces, similar to the Or-Or interface, between the t-MANOs' orchestration functions (i.e., t-NFVO) without involving the c-MANO as long as the level of control for inter-domain orchestration is within the agreed MLA bounds. Any other inter-domain operations exceeding the MLA will be performed and inter-coordinated between the different providers' c-MANO systems.

Implementation Aspects of MANOaaS

Architectural Considerations

Our approach focuses on the realization of the t-MANO stack by deploying its individual VMF components (i.e., t-NFVO, t-VNFM, and t-VIM) in a similar fashion as a VNF, where the t-MANO VMF instances maintain peer relationships with the respective c-MANO functional blocks either over an IP-based network or via some RPC-based method. Figure 2 gives an overview of the deployment of a t-MANO stack vis-a-vis the c-MANO for T1 and T2. The tenants are allocated a quota of NFVI resource blocks by the c-MANO using the NFV-MANO standard method.

Also, the required VNF packages and relevant catalogues are also onboarded by the tenant to the c-MANO using the standard onboarding technique. Upon a tenant's request, the c-MANO deploys and instantiates the t-MANO stack by deploying the constituent VMF components (i.e., t-VIM, t-VNFM, t-NFVO) in a similar manner as it would have performed for a VNF/NS. For this purpose, the c-MANO requires t-MANO-specific deployment templates provided in the two newly proposed catalogues, namely:

- t-MANO catalogue consisting of:
  - –t-MANO descriptor (TMD) file
  - –VMF forwarding graph (VMFFG) file
  - –Virtual link descriptor (VLD) file.
- The VMF catalogue consisting of a VMF descriptor (VMFD) file.

The information inside the TMD file is used by the NFVO and the VNFM of the c-MANO system to instantiate a t-MANO instance for the tenant, which consists mainly of the three VMFs interconnected by VLs, whereby the characteristics of the VLs are described by the VLD. The VMFFG, similar to the VNFFG, contains meta-data about the VMFFG itself, and references to VLs, VMFs, connection points, and so on. The TMD, VMFFG and VLD are quite similar to NSD, VNFFG, and VLD but contain necessary extensions that are relevant to the t-MANO requirements. An overview on NSD, VNFFG, and VLD are provided in [4].

We assume that VLs between the t-MANO VMFs support the standard NFV-MANO reference points and the interfaces and operations defined over them, whereas the tenant requests the complete t-MANO stack. However, there might be some interfaces or operations that the t-MANO instance may not be allowed to access or execute due to MLA restrictions. The restricted interfaces and operations are executed by the c-MANO for the respective tenants' slices. The tenants may also maintain their own VNF/NS catalogues and repositories as part of their t-MANO service instance. Nonetheless, these catalogues and repositories must be first validated by the c-MANO before they are assigned to the t-MANO system.

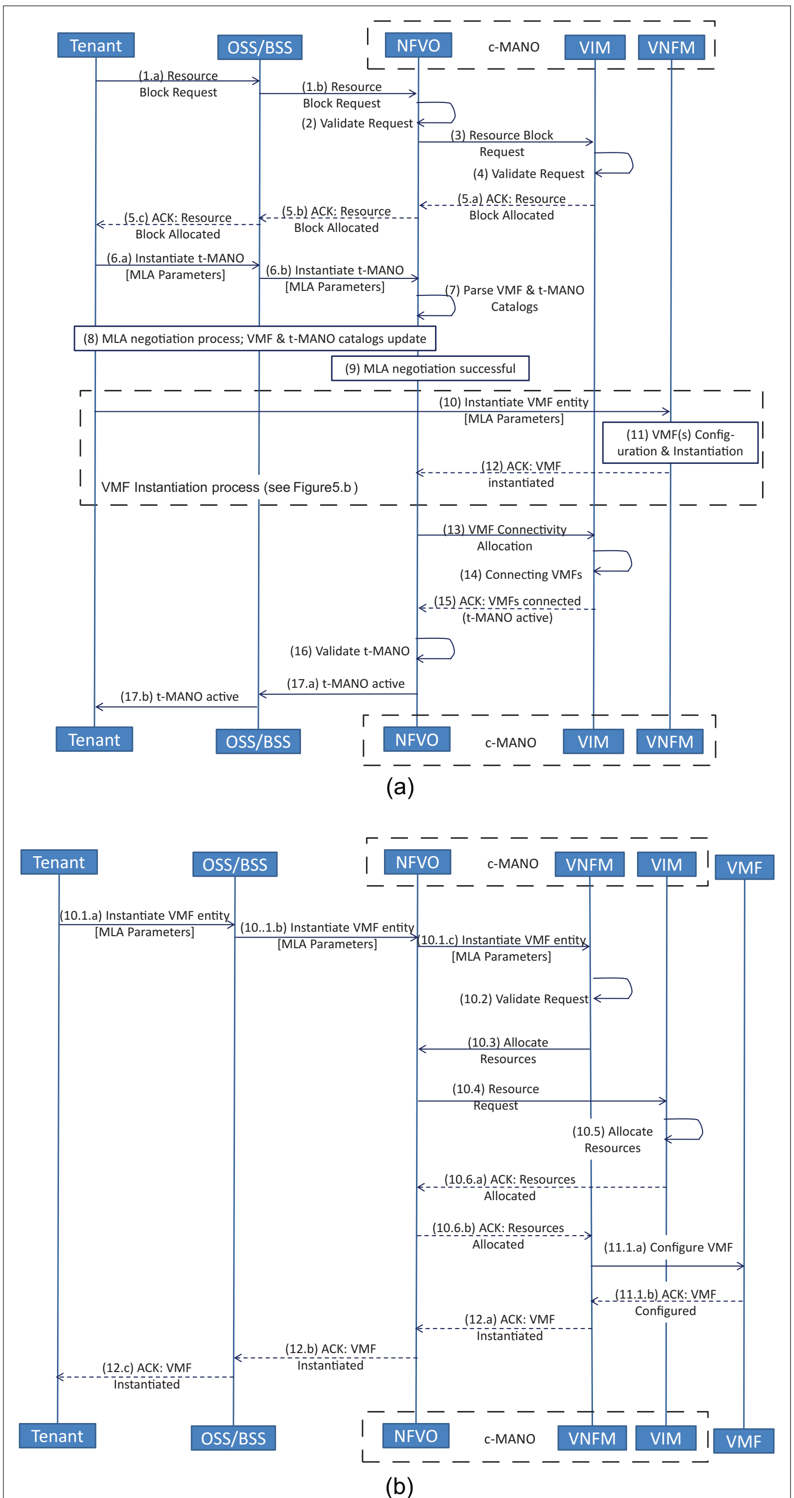

Figure 3. Process flow for instantiation and deployment of t-MANO: a) instantiation process of t-MANO stack; b) instantiation process of VMF instance.

The VMFD file is a deployment template that describes the respective VMF components of the t-MANO stack in terms of deployment, operational, and functional requirements. It also contains interface, connectivity and key performance indicator (KPI) requirements, and also specifies the services, features, and capabilities of the respective VMF components. Additionally, it shall also specify the parameters agreed during MLA negotiation specifying services, features, and capabilities of the respective VMF components that a tenant is allowed to access/execute. Thus, the VMFD files are more dynamic, unlike VNFD files, which are static deployment templates, and are updated during the MLA negotiation process. The VMFD can be updated each time the c-MANO updates any of its MLA parameters, for example, whenever the tenant creates a new sSlice. Thus, a t-MANO instance can be configured to recognize different MLAs for different sSlices belonging to the same tenant. Table 1 provides an example of a non-exhaustive list of the MLA parameters negotiated for a t-MANO instance. The TMD and VMFD files also contain information for identifying the various t-MANO stack instances and their location in a multi-site environment. This is evident from Table 1, where the parameters are indexed by the id of the t-MANO instance (t mano id) and the network slice(s) (ns id) managed by the respective t-MANO stack.

Process Overview of t-MANO Deployment and Instantiation

The c-MANO, with the help of the t-MANO catalogue and VMF catalogue, deploys a t-MANO stack for a tenant that can dimension the allocated resource block of NFVI resources into required rSlice(s) depending on the requirements of the tenant's sSlice(s). This is illustrated in Fig. 2, where the NFVI resource block allocated to a tenant is dimensioned into three resource slices, namely rSlice-1, rSlice-2, and rSlice-M. rSlice-M is dedicated to fulfilling the resource requirements specific to the t-MANO stack. Once the rSlices are created and the t-MANO deployed, the t-MANO performs regular LCM operations on the resource/service slices within the MLA bounds.

An overview of the general process for the deployment and instantiation of the t-MANO stack is illustrated in Fig. 3a, while Fig. 3b expands on the configuration/instantiation process of the VMF entities, which is part of the process depicted in Fig. 3a but presented as a separate figure for convenience. The c-MANO instantiates the t-MANO stack.

After the request has been validated (message-4) and the resource block assigned (message-5), the tenant requests instantiation of the t-MANO stack in its tenant domain (message-6). The request message also carries the MLA parameters (Table 1) specifying the access privileges that the tenant requires for accessing the service/feature/capabilities of the t-MANO stack. Then the NFVO parses the VMF and t-MANO catalogues (message-7) to verify that all relevant VMF packages and descriptor files are available and up-to-date.

Once the verification is completed, the NFVO and the tenant undergo the MLA negotiation process (step 8). Assuming a successful negotiation (step 9), the relevant VMF descriptor files are dynamically updated. Since the MLA has been successfully negotiated, the NFVO starts the process of instantiating and configuring the VMF instances (i.e., t-NFVO, t-VNFM, t-VIM) for the t-MANO stack requested by the user (process flow 10–12), as shown in Fig. 3b. Specifically, the tenant makes a request for the instantiation of a VMF entity to the c-MANO's VNFM entity via OSS/BSS (messages 10.1.a-b). The NFVO, based on the VMFD, forwards this request to the VNFM (message 10.1.c), which after validating the request (message 10.2) sends a message to NFVO to allocate the requisite resources for the requested VMF entity (message 10.3). The NFVO forwards this message to the VIM (message 10.4), which reserves and allocates the necessary resources from the resource block within the tenant domain for the requested VMF instance (message 10.5).

Once the VMF has been deployed, the VNFM configures the newly deployed VMF in the tenant domain with necessary configuration parameters specified in the VMFD and bounded by the agreed MLA parameters (message 11.1.a). Now the VMF configuration is confirmed (message 11.1.b), so the VNFM informs the tenant that the requested VMF has been instantiated (messages 12.a–c). Note that such a process is repeated for each different type of VMF block belonging to the t-MANO stack (i.e., t-NFVO, t-VNFM, and t-VIM).

While VMFs are being configured and instantiated, the NFVO instructs the VIM to establish connectivity between the t-MANO VMF instances to realize the respective reference points (messages 13–14 in Fig. 3a). The NFVO validates the operational integrity of the newly instantiated t-MANO by performing some validation tests (message 16). A successful validation notifies the tenant that the requested t-MANO is active and ready to be used (messages 17.a–b). The tenant can use it for creating, managing, and orchestrating its own resource/service slices and policies within the bounds of the agreed MLA.

To validate our proposal, we carry out a simulation campaign developing a simple network simulator in a well-known mathematical tool, MATLAB, where a different number of tenants may directly request resources from the c-MANO entity, as depicted in Fig. 2. The sizes of the resource blocks among the tenants are distributed randomly following the normal distribution. Requests within the tenants-domain for rSlice arrive following a normal distribution, with 30 percent requiring up to 80 percent of the total available capacity of the

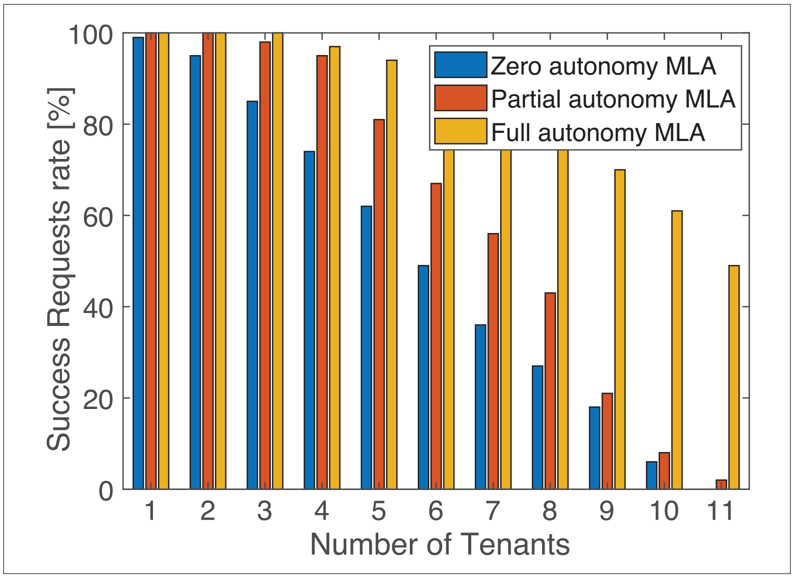

Figure 4. Success resource requests rate with multiple tenants and different negotiated MLAs.

system.Some random requests may also ask for resources more than the total available capacity, which will be rejected. Such values are chosen to meet the expected 5G requirements on the complete isolation of a few slices. We omit the analysis of the overhead required by each single approach due to the space limitation, leaving it to future work.

Preliminary results provide an empirical evaluation in Fig. 4. In particular, we show the impact of the degree of autonomy in terms of the success rate of t-MANO requesting resource reservation for, say, slice instantiation/scaling/migration at runtime within low to high resource contention scenarios. Considering that the underlying infrastructure resources are shared, a t-MANO has to reserve an rSlice for carrying out sSlice instantiation/scaling/migration operation. In the case of a t-MANO instance with an MLA granting full autonomy (i.e., the c-MANO operational control is fully delegated and resources are guaranteed to the t-MANO instance), it can be seen that the success rate is much higher than those of t-MANO stacks with partial or zero autonomy. This is because with zero or partial autonomy, the t-MANO has greater dependence on c-MANO for resource reservation. Their requests will be referred to c-MANO, where they will contend against other similar requests from other tenants. Thus, their requests may be rejected in situations of high resource contention. This behavior gets more pronounced when there are more tenants (higher resource contention) asking for finite resources, in which the t-MANO will exhibit a much lower success rate with respect to t-MANOs with full autonomy. In other words, the t-MANO getting full granted autonomy has a much higher success rate due to stringent resource guarantees.

MANOaaS in the Context of SDOS and Open Source Projects

The NFV MANO is the standard MANO framework proposed by the ESTI Industry Standard Group (ISG) on NFV. There are several open

[1] https://osm.etsi.org/

[2] https://www.onap.org/

Without being specific to any particular implementation of a MANO stack, some projects specify their own proprietary framework to wrap around complex functions for the handling of orchestration and lifecycle management tasks, but still offer a tenant sufficient level of control and customization.

source projects that are developing MANO platform like ETSI's OSM[1] and LFN's ONAP.[2] However, the open source projects do not align 100 percent with the standard NFV-MANO reference architecture. They cover the basic management functions of ETSI's NFV-MANO, like NFVO and VNFM functionalities, but they provide additional value-added features. For VIM, these platforms rely on other open source projects such as OpenStack as being more prominent options.

In consideration of enabling MANOaaS support, the control and semantics being exposed from a c-MANO provider to a tenant should be carefully considered, while keeping the concrete platform of c-MANO transparent. Considering that t-MANOs are semi-autonomous with still some level of dependence on c-MANO (e.g., to enforce rules for virtualized infrastructure management and states distribution between t-MANO and c-MANO components) needs to be thoroughly investigated and planned toward the design of suitable redundancy management and fail-over procedures. Consideration should be given whether the value added features, such as for data analytics, can be instantiated on a t-MANO and exposed to a tenant. The relevance to keep in mind of closer association of VNF context with components of the c-MANO and the t-MANO is, for example, further substantiated by a recently proposed study in the Third Generation Partnership Project (3GPP) on the analysis and harmonization of semantics used on one hand by a slice-specific network data analytics VNF, and on the other hand by the ONAP data analytics components.

Without being specific to any particular implementation of a MANO stack, some projects specify their own proprietary framework to wrap around complex functions for the handling of orchestration and LCM tasks, but still offer a tenant a sufficient level of control and customization. If a tenant is not interested in obtaining control of a customized MANO stack, [14] specifies a high-level application programming interface (API) named One-stop API, which abstracts low-level NFV-MANO functions but enables a tenant to instantiate and manage a tailored slice, whereas a feature denoted as plug-and-play exposes a combination of customized control functions and tools to a tenant in support of plugging proprietary control logic into a slice (vertical-in-the-loop). Reference [15] introduces the slice-as-a-service concept based on virtualized compute and networking resources, which serve as a base to onboard services per tenant request. A request is processed by the Lightweight Software Defined Cloud (LSDC) platform, which on one hand offers different levels of granularity and formats to request a slice, but then takes over the LCM of the instantiated service.

Conclusions

Service verticals require advanced management and orchestration of network resources in 5G network deployment. In this article, we have broken down the monolithic concept of the MANO stack to bring into play the network slicing paradigm by introducing the novel concept of MANO-as-a-service (MANOaaS). In particular, the centralized MANO (c-MANO) has been abstracted into multiple distributed instances (t-MANOs) pursuing differentiated management level agreements between infrastructure provider and tenants. We have:

- Presented a novel MANOaaS architecture showing the inter-connections between distributed instances and the centralized MANO stack
- Detailed autonomy negotiation process of the management level agreements between infrastructure provider and tenants
- Unveiled the similarity and open issues with the main standardization bodies and ongoing research projects

We believe that this solution represents an important and required step toward full network slicing isolation with additional performance and management gains in 5G networks.

Acknowledgment

This work has been partially funded by the European Union Horizon-2020 Project 5G-CARMEN under Grant Agreement 825012 and Project 5G-Transformer under Grant Agreement 761536.

References

[1] NGMN Alliance, "Description of Network Slicing Concept," NGMN 5G P1 Requirements & Architecture, Work Stream End-to-End Architecture, 2016.

[2] J. Salvat *et al.*, "Overbooking Network Slices through Yield-Driven End-to-End Orchestration," *ACM CONEXT 2018*, Dec. 2018, pp. 1–12.

[3] S. Zhang *et al.*, "An Opportunistic Resource Sharing and Topology-Aware mapping framework for virtual networks," *IEEE INFOCOM*, Mar. 2012.

[4] ETSI NFV ISG, "Network Function Virtualization (NFV) Management and Orchestration," GS NFV-MAN 001 v0.8.1, Nov. 2014.

[5] R. Mijumbi *et al.*, "Management and Orchestration Challenges in Network Functions Virtualization," *IEEE Commun. Mag.*, vol. 54, no. 1, Jan. 2016, pp. 98–105.

[6] Y. Li *et al.*, "Network Functions Virtualization with Soft Real-Time Guarantees," *IEEE INFOCOM*, Apr. 2016.

[7] "D3.3: 5G NORMA Network Architecture — Final Report," Sept. 2017; https://5gnorma.5g-ppp.eu/dissemination/public-deliverables.

[8] L. Freitas *et al.*, "Slicing and Allocation of Transformable Resources for the Deployment of Multiple Virtualized Infrastructure Managers (VIMs)," *Wksp. Advances in Slicing for Softwarized Infrastructures*, Montreal, Canada, June 2018.

[9] ETSI NFV ISG, "Report on Architecture Options to Support Multiple Administrative Domains," GR NFV-IFA 028 V3.1.1, 2018.

[10] R. Riggio *et al.*, "LightMANO: Converging NFV and SDN at the Edges of the Network," *IEEE/IFIP Network Operations and Management Symp.*, 2018.

[11] M. Bari *et al.*, "Orchestrating Virtualized Network Functions," *IEEE Trans. Network and Service Management*, vol. 13, Dec. 2016, pp. 725–39.

[12] R. Rosa, M. Santos, and C. Rothenberg, "MD2-NFV: The Case for Multi-Domain Distributed Network Functions Virtualization," *Int'l. Conf. and Wksps. Networked Systems*, 2015.

[13] A. Oliva *et al.*, "5G-TRANSFORMER: Slicing and Orchestrating Transport Networks for Industry Verticals," *IEEE Commun. Mag.*, 2018.

[14] "SliceNet: D2.2 — Overall Architecture and Interfaces Definition," Jan. 2019; https://slicenet.eu/deliverables/, accessedApr. 5, 2019.

[15] "NECOS: D3.1 — NECOS System Architecture and Platform Specification," July 2018; http://www.h2020-necos.eu/ documents/deliverables/, accessed Apr. 5, 2019.

Biographies

Faqir Zarrar Yousaf (zarrar.yousaf@neclab.eu) is a senior researcher at NEC Laboratories Europe, Germany. He completed his Ph.D. at TU Dortmund, Germany. His current research interest is in NFV/SDN in the context of 5G networks. He is also a delegate to the ETSI NFV standards organization, where he is a Rapporteur for four work items and has contributed to several standards. He has one granted patent and 16 filed patents, and his research work has been widely published.

[1] https://osm.etsi.org/

[2] https://www.onap.org/

VINCENZO SCIANCALEPORE (vincenzo.sciancalepore@neclab.eu) is a senior researcher and RAN specialist at NEC Laboratories Europe, Germany. He is currently focusing his activity in the area of network virtualization and network slicing challenges. He is currently involved in the IEEE Emerging Technologies Committee leading the initiatives on SDN and NFV. He was also the recipient of the national award for the best Ph.D. thesis in the area of communication technologies in 2015.

MARCO LIEBSCH (marco.liebsch@neclab.eu) is a chief researcher in the 5G Networking group at NEC Laboratories Europe, and has strong background knowledge in the areas of mobility and policy management, NFV, and the 3GPP EPC and its evolution toward 5G. He received a Ph.D. from the University of Karlsruhe in 2007. He has worked on different EU projects and is contributing to standards in the IETF and the 3GPP.

XAVIER COSTA-PEREZ (xavier.costa@neclab.eu) is head of 5G Networks R&D and deputy general manager of the Security & Networking Research Division at NEC Laboratories Europe. His team contributes to products' roadmap evolution as well as to European Commission R&D collaborative projects, and has received several awards for successful technology transfers. In addition, the team contributes to related standardization bodies: 3GPP, ETSI NFV, ETSI MEC, and IETF. He received both his M.Sc. and Ph.D. degrees in telecommunications from the Polytechnic University of Catalonia, Barcelona, and was the recipient of a national award for his Ph.D. thesis.