Contreras *et al. J Wireless Com Network* (2021) 2021:114
https://doi.org/10.1186/s13638-021-01992-6

EURASIP Journal on Wireless
Communications and Networking

RESEARCH **Open Access**

Architecture for integrating vertical customer's programmability control of network functions and connectivity in a slice-as-a-service schema

Luis M. Contreras[1*], Samier Barguil[2], Ricard Vilalta[3] and Victor López[1]

*Correspondence:
luismiguel.contrerasmurillo@telefonica.com; lcontrmu@gmail.com
[1] Telefónica CTIO Unit, Madrid, Spain
Full list of author information is available at the end of the article

Abstract

Network slicing will permit offering to vertical customers tailored end-to-end logical networks in an on-demand fashion, on top of a common telecom infrastructure, achieving a Slices-as-a-Service (SlaaS) business model. This is possible due to the progressive introduction of network softwarization techniques, such as programmability and virtualization, into existing operational networks, enabling dynamic and flexible provision of slices. Those vertical customers could require the control not only of the network functions composing the end-to-end service, but also of the connectivity among them, e.g., for influencing the paths for steering traffic among function instances. However, this can be problematic since decisions from one vertical customer can collide with decisions from others. One aspect not yet sufficiently investigated is how to permit vertical customers to jointly control the service functions and the underlay connectivity, in such a way that could operate the allocated slice as if it was actually a dedicated network entirely for them. This paper explores some architectural proposition in this respect illustrated with some potential use cases and it provides an example of the provision of SlaaS for a vertical customer.

Keywords: Network slicing, Vertical services, Programmability

1 Introduction

Existing telecom networks are on transition towards a new operational model in which the network is expected to be consumed dynamically by external customers, e.g. vertical industries, through the ability of requesting the necessary capabilities and resources in a flexible manner. This new model is known as network slicing, concept fostered by the advent of 5G but that is of wider applicability and extendable to any kind of service (e.g., as an evolution of existing wholesale corporate services).

Network slicing represents a means for offering tailored services to customers in terms of capacity, functionality, and topology without altering or customizing the physical network infrastructure. This is done by leveraging on the multi-tenancy concept. A first step in this direction [1] has been the wide introduction of cloud computing facilities

Springer Open

distributed in telecom networks. This facilitates the dynamic instantiation of services and applications, while at the same time interacting with the operator transport network for ensure connectivity. However, this represented yet an overlay approach where the transport network accommodates traffic from cloud environments in an aggregated manner. This is clearly not enough for future service where a more granular and independent behaviour is expected.

This idea of network slicing as multi-tenancy with such granular approach per customer, i.e., customizing both compute and network environments to the specific customer needs, has been extensively explored recently in the literature. Previous survey works in [2–5] introduce the characteristics of network slicing as the mechanism for allowing the co-existence of different vertical industries on top of the same physical infrastructure, linked essentially to 5G. The work in [2] focuses on analysing the network slice concept from the perspectives of the infrastructure (the resources), network function (both control and user plane functions) and the overall service layers (touching different actors such as operators and verticals), also including a discussion on management and operation aspects (mostly orchestration and mapping of functions to resources). In [3] a description of main enablers of network slicing is provided, including the main stakeholders, and expected use case. The work in [4] focuses on methods for the allocation of resources to the demanded slices across different network segments. Finally, [5] complements the analysis with a wide overview of proposed architectures for network slice implementation. All in all, the existing work is mainly focused on the slice provision as a dedicated (logical) network for vertical customers, but without a view on how the vertical customers can operate the allocated resources. That is the main motivation for this paper, by introducing architectural ideas in that direction.

Thus, the aforementioned transition that telecom networks are experimenting is ongoing despite the fact that current telecom networks yet rely in part on old structures and management mechanisms. Network slicing will speed up the need for changes, and it is necessary to reconcile novel procedures with legacy assets as long as the transition is not totally finished. Undoubtedly, this is one of the great challenges that operators face nowadays.

One essential aspect yet to explore is the capability by the vertical customers of controlling, managing, and programming the slices individually, including not only the service part (i.e., the network functions) but also the transport network slice instance. This can raise a number of conflicts if the slices serving different customers are not appropriately isolated in both the control and forwarding planes.

This paper concentrates on the vertical customer control of the allocated slice by exemplifying an architecture enabling the programmability of the allocated network functions and their associated connectivity. The main contributions are:

- Description of mechanisms permitting vertical customers to simultaneously control functions and resources allocated in the form of a network slice.
- Proposal of a novel architecture allowing telecom operators to offer control capabilities to vertical customers of the allocated slice, both at network function and transport levels.
- Example of a vertical slice provision.

The structure of the paper is as follows. Section 2 describes in detail the concept of Slice as a Service, as new formulation of services offered to vertical customers, and presents how the programmable control of those slices can be accomplished by verticals. In Sect. 3, an overall architecture is presented, including two applicability cases, as well as how they can be integrated on legacy environments. In Sect. 4, it is provided an example of the provision of SlaaS for a vertical customer. Finally, in Sect. 5 some concluding remarks and future lines of work are presented.

2 Combined control of service functions and transport connectivity by vertical customers

Network slicing is a paradigm through which different virtual resource elements of a common shared infrastructure (in both connectivity and compute substrates) become allocated to a specific customer who perceived the resulting slice as a fully dedicated, self-contained network for it. The resources are virtualized through a process of abstraction of actually physical lower-level elements, providing a great flexibility and independence of the specific element being used along the customer service lifetime, which permits exercise advanced actions such as scalability, reliability, protection, relocation, etc. All of them represent an incredibly asset for a novel way of service provisioning with respect the present mode of operation in telecom networks.

These new capabilities are enabled by the increasing introduction of Software Defined Networking (SDN) [6, 7] and Network Functions Virtualization (NFV) [8, 9] within operational networks, a process commonly referred as network softwarization. On one hand, it allows automation by the introduction of complementary techniques that could permit triggered corrective actions in a closed loop manner. Secondly, it facilitates a rapid service provisioning through easy reconfiguration of network connectivity and fast deployment of network functions across distributed computing facilities, adapting the service in every moment to the specific circumstances of the network, now interacting gracefully.

The possibility of dynamically instantiating slices through automation enables the provision of slices in an on-demand fashion, dealing to the concept of Slice-as-a-Service (SlaaS). The final objective of a customer for requesting a slice is to dispose of a complete logical end-to-end network on which to deploy the vertical service with full guarantees. The network is then transformed into a production system merging both business and operation domains [10]. On one hand, different business models can be formed around the offering of network slices: Business-to-Business (B2B), Business-to-Consumer (B2C) and Business-to-Business-to Consumer (B2B2C), depending on the kind and variety of stakeholders addressed in the business part. On the other, distinct operational implications (life cycles, service objects, and slice scales) must be taken into account derived from the service scenario where the requested slice applies.

A critical point on the overall provision of a slice is to allow control of the allocated abstract resources to the customer (e.g., the possibility of programming them). Without such control, the slice is simply made available but cannot be re-configured by the customer, leading to a kind of static network. On the contrary, if control capabilities are enabled for the customer, the network can be then flexibly managed, for example be reconfiguring forwarding paths adapting to changing conditions of traffic within the slice.

Different kind of slices can be considered from the operator perspective in that sense [11], as follows:

- *Internal slices*, as the slices in which the operator retains the total control and management capabilities. This kind of slices will be typically devoted to provider's internal services.
- *External slices*, offered to vertical customers which perceived them as dedicated networks, but which in reality run on top of shared infrastructure. For external slices is yet possible to distinguish:
  - Slices managed by the operator, where the operator performs the control and management of the slice and the vertical customer simply runs the service on top of the capabilities and resources offered by the operator.
  - Slices managed by the vertical customer, where the customer actually has control of the resources and functions allocated. The level of control could be limited to a set of operations and/or configuration actions, but in any case, the vertical has the possibility of governing the slice behaviour to some extent.

The latter case is the focal point of discussion in this paper. The referred control capabilities in that case should be enabled with care, since different actions from distinct customers could collide. The particular actions from each customer are performed on their specific slice, that is, on the abstracted resources previously allocated. However, those abstractions could affect to a common physical resource shared among slices. Thus, if contradicting configuring actions are in place one customer could negatively impact on the slice of another customer.

Figure 1 graphically represents this distinction, showing the different responsibilities in each case.

For vertical customers that could require a deeper level of control over the service and network resources of their end-to-end slices, there is a need of enabling control mechanisms with that purpose. In the case of the service functions, the objective of such control can be the control of the service logic, basically the configuration of the functions according to the service conditions. In the case of the network resource control, the purpose can be to provide those customers with the ability to manage dynamically the connectivity paths and the overall interconnection topology for each particular logical network.

These two levels of control show two different concerns, service and transport. From a programmability point of view, this approach was developed in [12] and it is illustrated in Fig. 2. This separation of concerns is convenient since the purposes are clearly differentiated, then facilitating operations. When considering a network slice, the vertical customer could require to manage both kind of concerns, as in the case of the external slices managed by the vertical. In that case, the programmability in the service stratum essentially allows the interaction with the network functions

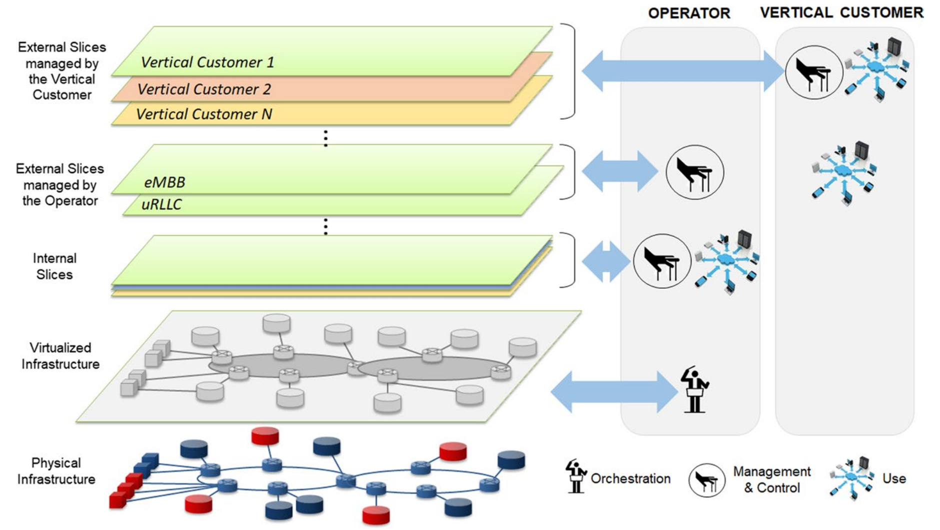


Fig. 1 Types of network slices according to management and control levels of responsibility

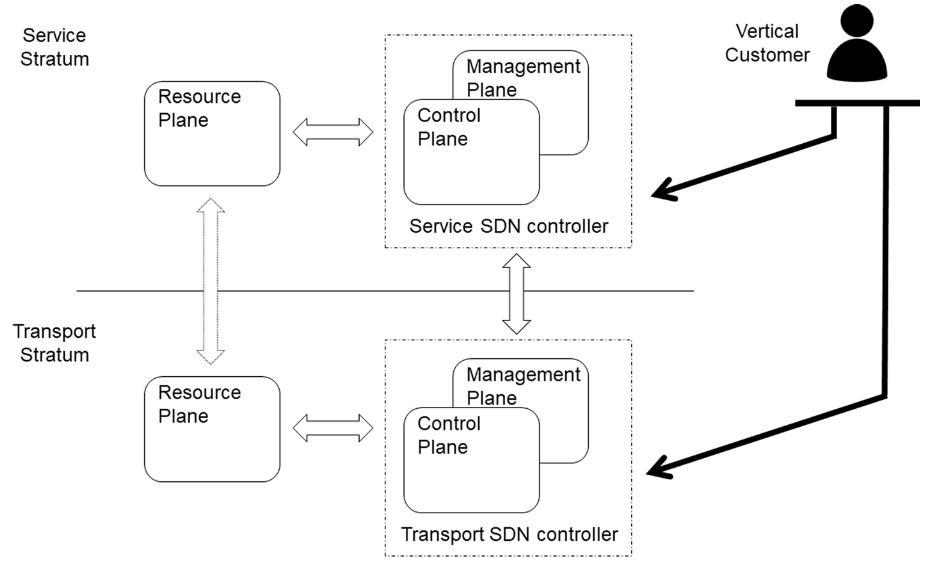


Fig. 2 Cooperating Layered Architecture for Software-Defined Networking (CLAS) [14] as leveraged by the vertical customer

composing the vertical service, while in the case of the transport stratum, the control is performed over the nodes providing connectivity end-to-end.

Furthermore, an interaction among controllers could be also expected if some degrees of automation are present. This could happen in situations when the service controller can leverage on the capabilities of the controller handling the connectivity for adapting the behaviour of the service function (e.g., bit rate adaptation), or alternatively, if the controller responsible of the connectivity can react to service situations (e.g., increase of bandwidth).

2.1 Programmable control of service functions

The service functions composing the vertical service will be instantiated dynamically as Virtual Network Functions (VNFs) leveraging on the ETSI NFV architecture. These virtualized instances of service functions run on top of the NFV Infrastructure (NFVI), formed by the compute, storage and associated networking resources. The VNF management and operation usually resides in conventional Element Managers. The architecture is complemented by the Virtualized Infrastructure Manager (VIM), in charge of managing the interaction of a VNF with the NFVI, the VNF Manager (VNFM) responsible of the VNF lifecycle, and the NFV Orchestrator, coordinating all the necessary actions among components in the architecture for instantiating and deploying the services composed of network functions.

In [13] two distinct SDN controllers for different purposes are proposed for the NFV architecture. On one hand, the so-called Tenant SDN Controller (TC) which controls associated VNFs in a programmatic manner. On the other hand, the Infrastructure SDN Controller (IC) enabling the programmable control of the virtualized connectivity infrastructure. Both SDN controllers can communicate and interact in order to synchronize actions, e.g. scaling up infrastructure while reconfiguring function capabilities. Figure 3 shows the role of both controllers.

The TC could accomplish simple configuration and management tasks. In some other situations, the TC can assist on the configuration of advanced service capabilities. For instance, in the case of Service Function Chains (SFC) [14] the TC can act as a SFC controller [15] programming classification rules at the entry of the SFC domain or instructing the Service Functions about the semantics of the context information encapsulated in the SFC's Network Service Headers (NSHs) [16]. Also, as part of the service logic, and leveraging on the communication between TC and IC, the TC could trigger the IC for populating rules for lookup purposes at the Service Function Forwarder (SFF) elements which are the elements responsible for forwarding network

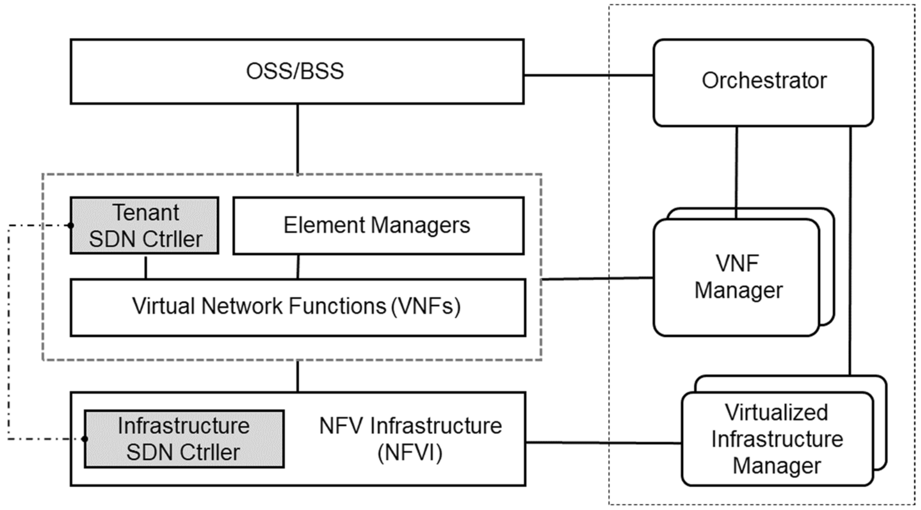

OSS/BSS
Orchestrator
Tenant SDN Ctrller
Element Managers
Virtual Network Functions (VNFs)
VNF Manager
Infrastructure SDN Ctrller
NFV Infrastructure (NFVI)
Virtualized Infrastructure Manager

Fig. 3 Roles of Tenant and Infrastructure SDN controllers in ETSI NFV framework

traffic to and from Service Functions within the SFC domain. The SFFs can then be realized through part of the networking infrastructure in the NFVI.

2.2 Programmable control of transport connectivity

The virtualization of the transport network can be exploited at the time of provisioning and orchestrating the virtualized service functions of the slice. The idea leverages on the concept of Wide-area Infrastructure Manager (WIM) as defined by the ETSI NFV architecture as the element devoted to manage virtualize capabilities in the WAN [17]. When referring to slices, such an entity could be associated to a Transport Slice Controller, as defined in [18] either complementing or being part of an overarching SDN transport network control environment as in [19, 20].

The WIM is in charge of controlling the entire WAN infrastructure, and as such able to perform changes and reconfiguration in the network. Thus, a first option for granting control capabilities to the vertical customer could be simply providing access to the operator's WIM. This, however, can be complex and overwhelming considering that multiple verticals could require from such an access simultaneously. Alternatively, it could be feasible to instantiate a tailored WIM per vertical adapted to the specific control needs of each customer.

In this sense, two alternative models can be considered, as described in [21]. One model is denominated *Full WIM*, where the vertical customer has direct control on the networking connectivity assets, and an alternative one, named *WIM Agent*, where the control is indirectly performed through a centralized control element or network controller which actually has the full control of the transport network. Figure 4 shows the two alternative models.

For the *Full WIM* case to work, the transport connectivity resources have to be entirely dedicated to each vertical customer to avoid configuration conflicts and network inconsistencies. It is also implies that the vertical's WIM control entity has to support at its South-Bound Interface (SBI) all the variety of configuration methods for accessing and configuring the devices forming the underlying transport substrate. This is essentially

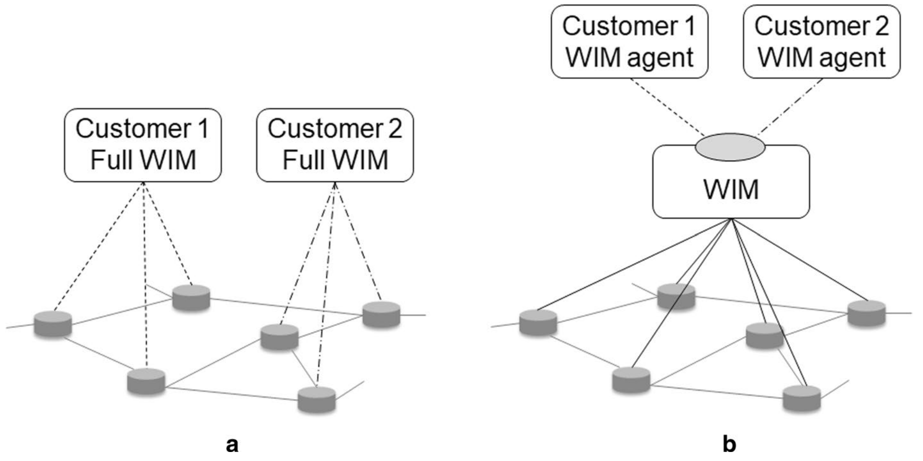


Fig. 4 Alternative models for transport control: **a** Full WIM and **b** WIM agent

complex since in addition to multiple possible vertical WIMs flavours, in the transport network there will typically be a large variety of devices from different manufacturers.

In the case of the *WIM agent*, since the full control is retained by the centralized WIM controller, the configuration conflicts can be naturally resolved by the central configuration point, which facilitates a customized (even limited) set of control capabilities to the WIM agent. From the integration perspective, the vertical's WIM control entity only requires to consume the centralized North-Bound interface (NBI), which largely simplifies the WIMs implementations, since the NBI offers unified methods.

Here the *WIM agent* approach is adopted as the mechanism for enabling the vertical customer to manage the slice connectivity. Thus, the vertical customer is granted during the lifetime of the slice with a *WIM agent* as SDN controller for the allocated transport connectivity resources. That agent or SDN controller instantiated for the vertical could be whatever implementation of preference by the customer (e.g., ONOS, OpenDaylight, etc.), simply ensuring the interaction with the centralized WIM NBI. The functionality expected for the vertical agent is the steering of the traffic flows in the slice, the control and management of the virtual paths and topology, and the associated performance and fault management.

3 Proposed architecture

The architecture proposed in this paper permits the vertical customer to get programmatic control over the service functions and the network connectivity constituting the slice provided by the operator. It is graphically summarized in Fig. 5. Once the slice is instantiated, the vertical customer is provided with access to both Function and Network control. The Function control permits to manage and configure the service functions composing the end-to-end vertical service in the different data centers or NFVI PoPs where those functions are deployed. This Function control is linked to the service stratum, then associated to the particular service logic of the vertical. The service functions are deployed as dedicated virtual machines or containers, for instance, on commodity

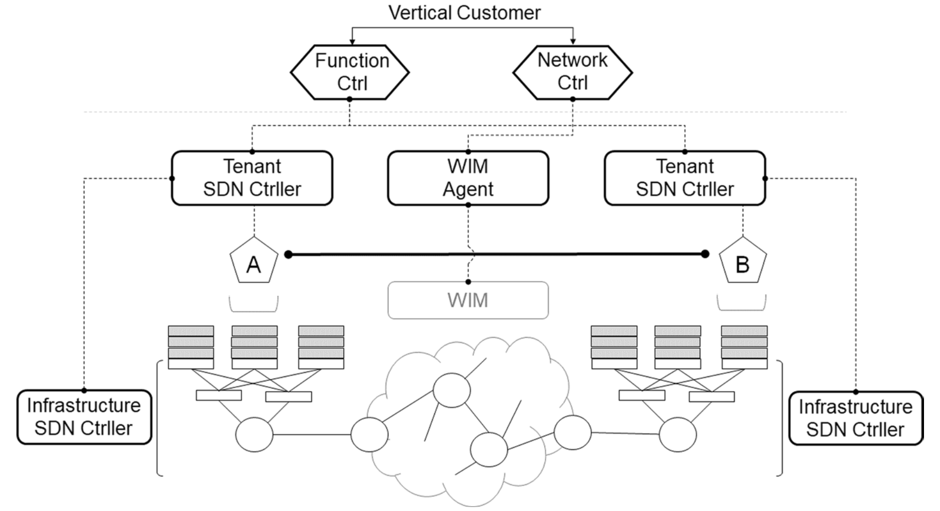


Fig. 5 Architecture enabling vertical customer control of service functions and their connectivity

servers within the data center. The Function control has a correspondence with the TC of the architecture in [13] in Sect. 2.1. Such correspondence could be as loose as a simple interaction through a well-defined API (as provided by the specific provider of the function to be controlled), or can be as tight as a direct implementation of the TC itself, e.g. through a virtualized instantiation of such controller for the allocated functions.

The Function controller, through the interaction between the TC and the IC in each data center, can indirectly interact with the networking infrastructure of each data center, that is, the router acting as data center gateway (connecting to the WAN) and the leaf and spine switches that typically serve to interconnect the compute nodes hosting the functions. This interaction can happen for instance to accompany scaling events at the service function side (e.g., [22, 23]), thus ensuring that the connectivity associated to the capacity of the network function does not become a bottleneck or, on the contrary, that some connectivity resources can be released, if needed.

The Network controller assists the vertical customer on the control of the transport connectivity on the WAN. The Network controller has a correspondence with the *WIM agent* in Sect. 2.2, and again such correspondence could be loose or tight depending on if it is made through APIs (through an agreed interface, e.g. the ONF Transport API [24]) or as an actual instantiation of the *WIM agent*.

As a result, the vertical customer perceives the slice not only as a dedicated logical network, but also as a *fully controllable* logical network, following the operational model of external slice managed by the vertical customer. The main advantage of this proposal is, then, the feasibility of offering programmatic control to the vertical customer on both functions and connectivity level, not possible with conventional slicing methods.

3.1 Scenarios of applicability

The following scenarios of applicability illustrate how the proposed architecture can assist on the operation of vertical slices.

3.1.1 Service chain across multiple NFVI PoPs

The ETSI NFV architecture supports multi-Point of Presence (PoP) configurations, where a PoP is defined as the physical location where network functions are instantiated, corresponding to a data center. When expanding more than one single PoP, multi-site connectivity should be performed to connect the functions local to each PoP, as described in [25]. For a vertical customer running the service end-to-end it can be necessary interact with the WAN for ensuring the proper behaviour of the service functions interconnection, forming a service chain. Such service chains can be mapped to a dedicated slice. In this manner, if the vertical customer requiring the slicing needs to perform such control actions on both the service and the connectivity, the operator can provide access to both service functions and network resources assisting the vertical in the management and control of the inter-site network slice connectivity.

The architectural schema in Fig. 5 represents this scenario.

3.1.2 Non-public network (NPN) integrated with operator's network

The digital transformation of productive environments by vertical industries is enabling the emergency of non-public networks (NPN) connected to public operator's networks.

Different interconnection models are foreseen ranging from standalone NPNs up to NPNs that show different degrees of integration with public networks [26]. The latter are the ones of interest for this paper, since such model implies also some level of interaction among the vertical service and the operator infrastructure.

This scenario assumes that the vertical customer leverages on the operator infrastructure for deploying part of its end-to-end service. This requires from the operator side the enablement of the programmable control of functions and connectivity. Taking as example 5G-based services, and assuming that a User Plane Function (UPF), in charge of forwarding and processing the user plane traffic, is deployed in the internals of the NPN, this would imply that the vertical will be able to manage and control not only the service functions forming the vertical service (e.g., data bases, content repositories, or even 5G control plane functions) but also the connectivity among the NPN UPF and the rest of elements (UPF and others) deployed at the operator's side. Figure 6 graphically described a generalization of this case.

3.2 Integration with non-programmatic networks

As described before, traditional wholesale service will evolve building on top of the idea of network slicing, sophisticating also the kind of service offerings for this market segment, by including the on-demand instantiation of service functions and their connection. The control schema for verticals will necessary have to interact with non-programmable parts of the network, since brownfield scenarios have to be expected.

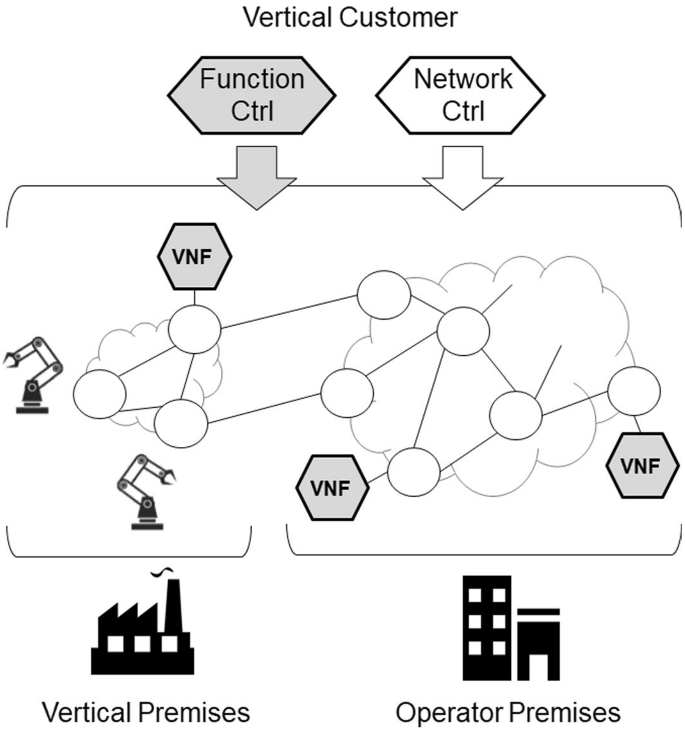


Fig. 6 Vertical customer control of service functions and their connectivity in NPN scenario

In the case of the virtualized service functions, the interaction with the legacy approach comes from the integration with physical, monolithic network functions already deployed in the network, known as Physical Network Functions (PNFs). From the service perspective, since the function logic is expected to be the same for both virtual and physical versions of the functions, no major issues could be expected, other than potentially a limitation due to constrained resources (either in the virtual or physical case). A further interworking restriction could come from the data plane encapsulation methods to perform the communication between functions, but also this can be assumed to be similarly supported in both cases. From the management perspective, it is a common trend in the industry that the original management systems will evolve to include SDN capabilities, on one hand, and that those systems will run as software components (e.g., virtual machines) on top of common-of-the-self (COTS) servers.

With respect the programmability of the connectivity substrate, when moving to the SDN paradigm, the integration of legacy, non-programmable devices is commonly performed through the development of plugins or adaptors to interact with either the management systems of such legacy infrastructure or directly with the equipment though the particular Command Line Interface (CLI) of each manufacturer.

In this manner, a path for integration of programmable and non-programmable infrastructure can be facilitated with the approach here considered.

4 Experimental section: exemplary provision of SlaaS

This section exemplifies the provision of SlaaS for a vertical customer. We leverage on SONATA framework [27] for the slice creation on the provider's side. Once created, the slice is made available for the vertical customer, which can access the slice capabilities (i.e., VNFs and network capabilities) through programmable interfaces.

The scenario of provision considers two separated and interconnected NFVI PoPs in different geographical locations. The SONATA Service Platform interacts with the different VIMs in each of the PoP as well as with the transport network, via the controlling WIM, as represented in Fig. 7. The interaction with the VIMs is through the corresponding API (e.g., Openstack Nova), while the communication with the WIM is performed by means of the ONF TAPI. The VIMs will be in charge of deploying the VNFs running the service functions, while the WIM will connect both PoPs end-to-end. The figure shows the physical view of the network which provides the resources to the slice, as well as the logical view as the dedicated network perceived by the vertical customer.

The workflow of slice creation is sketched in Fig. 8, representing the interaction among the orchestration elements controlling the physical resources in Fig. 7. The SlaaS assumes the deployment of two VNFs, one on each of the PoPs, as well as connectivity service connecting both. The slice will be requested in the form of a Network Service (NS) either by the vertical customer (if having access to the SONATA platform) or directly by the slice provider after a request from the vertical. The platform will process the NS requests, eliciting the requirements of the slice in terms of forming VNFs, their location, and the necessary connectivity among them for the resulting service chain. After the reception of the NS request, the SONATA platform will interact with the VIMs of each of the PoPs for instantiating the VNFs. After that. The platform will also request

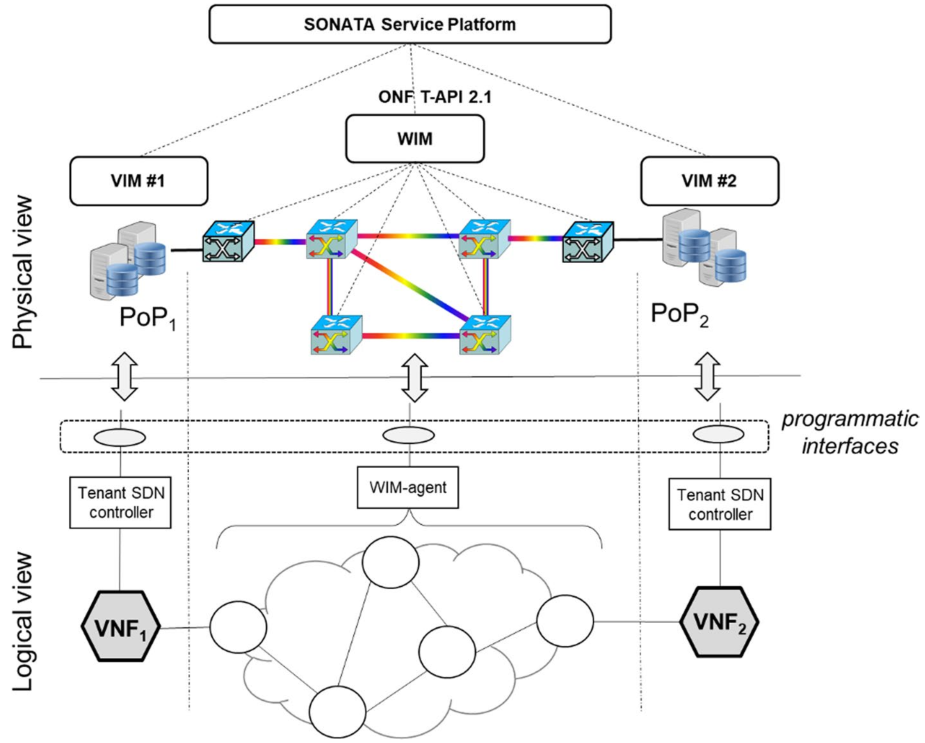


Fig. 7 Scenario of SlaaS provision

the WIM the creation of the connectivity service between VNFs, traversing the transport network.

Once the connection is established, the constitutive service chain associated to the slice becomes created and available for the vertical to operate. As result, the vertical customer gets the logical view represented in previous Fig. 7.

Figure 9 shows the functional messages interchanged during the workflow execution for the SlaaS creation.

Once created, the service provider is in condition of offering to the vertical customer the APIs for the control of the VNFs and the transport network, as described before. The programmatic control of the VNFs will depend on the particular implementation of the function, while for the transport network resources TAPI could be used since it natively allows recursiveness.

Finally, Fig. 10 depicts the CDF of the deployment time of the vertical customer slice. The observed results validate the idea of an affordable commercial on-demand provision of SlaaS for network operators.

5 Results and discussion

Current technologies, leveraging on programmability and virtualization, facilitate the provision of networks tailored to the needs of vertical customers, including service functions and the associated connectivity. That provision is performed in the form of a network slice, dedicating specific resources (both at compute and network level) for the service, guaranteeing isolation to avoid impacts from any other slice.

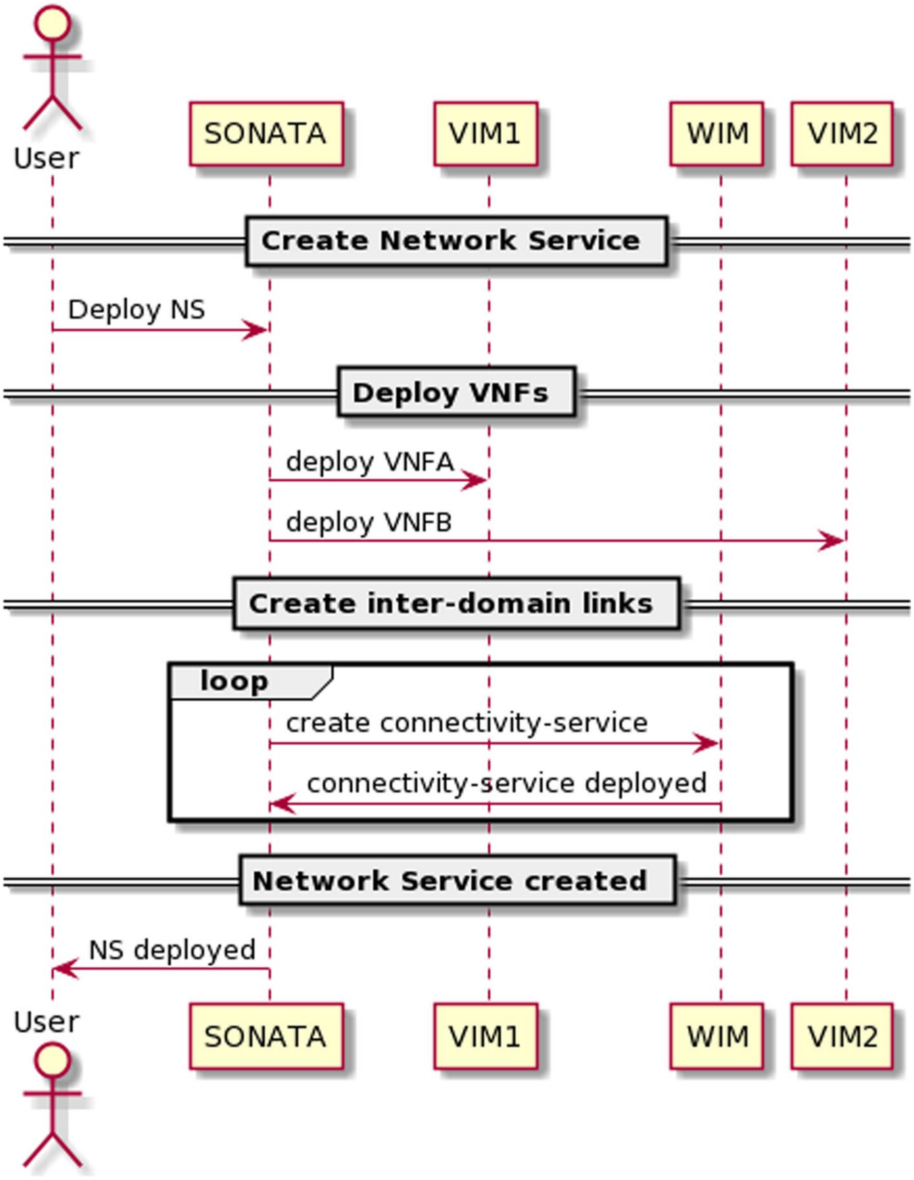
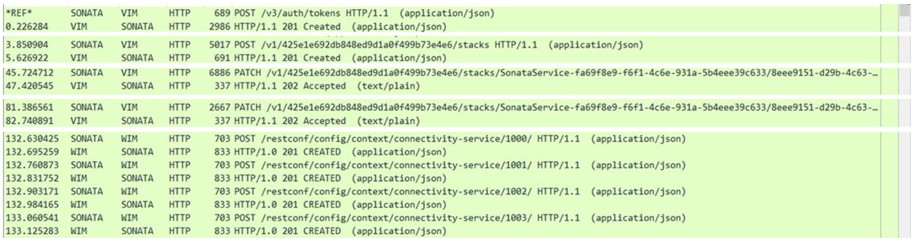


Fig. 8 SlaaS provision workflow



```
*REF*        SONATA   VIM      HTTP     689 POST /v3/auth/tokens HTTP/1.1  (application/json)
0.226284     VIM      SONATA   HTTP    2986 HTTP/1.1 201 Created  (application/json)
3.850904     SONATA   VIM      HTTP    5017 POST /v1/425e1e692db848ed9d1a0f499b73e4e6/stacks HTTP/1.1  (application/json)
5.626922     VIM      SONATA   HTTP     691 HTTP/1.1 201 Created  (application/json)
45.724712    SONATA   VIM      HTTP    6886 PATCH /v1/425e1e692db848ed9d1a0f499b73e4e6/stacks/SonataService-fa69f8e9-f6f1-4c6e-931a-5b4eee39c633/8eee9151-d29b-4c63-…
47.420545    VIM      SONATA   HTTP     337 HTTP/1.1 202 Accepted  (text/plain)
81.386561    SONATA   VIM      HTTP    2667 PATCH /v1/425e1e692db848ed9d1a0f499b73e4e6/stacks/SonataService-fa69f8e9-f6f1-4c6e-931a-5b4eee39c633/8eee9151-d29b-4c63-…
82.740891    VIM      SONATA   HTTP     337 HTTP/1.1 202 Accepted  (text/plain)
132.630425   SONATA   WIM      HTTP     703 POST /restconf/config/context/connectivity-service/1000/ HTTP/1.1  (application/json)
132.695259   WIM      SONATA   HTTP     833 HTTP/1.0 201 CREATED  (application/json)
132.760873   SONATA   WIM      HTTP     703 POST /restconf/config/context/connectivity-service/1001/ HTTP/1.1  (application/json)
132.831752   WIM      SONATA   HTTP     833 HTTP/1.0 201 CREATED  (application/json)
132.903171   SONATA   WIM      HTTP     703 POST /restconf/config/context/connectivity-service/1002/ HTTP/1.1  (application/json)
132.984165   WIM      SONATA   HTTP     833 HTTP/1.0 201 CREATED  (application/json)
133.060541   SONATA   WIM      HTTP     703 POST /restconf/config/context/connectivity-service/1003/ HTTP/1.1  (application/json)
133.125283   WIM      SONATA   HTTP     833 HTTP/1.0 201 CREATED  (application/json)
```

Fig. 9 Message interaction for slice creation

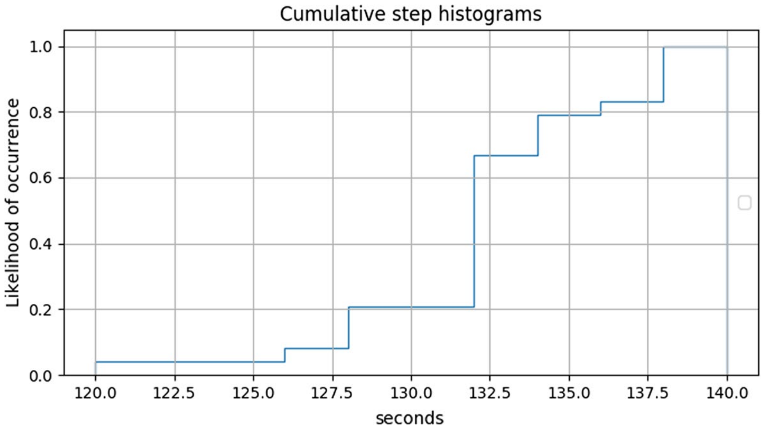


Fig. 10 CDF of vertical customer slice creation

The next step on such slice service offering is to allow the vertical customer to manage the allocated assets, in such a way that the vertical perceives the slice as a fully dedicated network under its own control. The way of doing that is to offer interfaces or APIs that could permit that fact. In the case of VNFs, the referred interfaces will be particular to the specific function considered. Regarding connectivity, the controller handling the connections associated to the slice could provide limited but sufficient views per vertical, or even existing developments, such as TAPI, can be used, since allow for recursiveness. This paper presents such architectural framework that enables the control by vertical customers of both service functions and network connectivity allocated to the slice. The viability of the on-demand provision of the slice has been validated through experiments orchestrating functions in different locations and connecting them forming a service chain. The provider can then offer programmatic interfaces to the verticals for providing control of the allocated assets, which will perceive the slice as an entirely dedicated network.

6 Conclusions and further work

On-demand Slice-as-a-Service is foreseen as the solution to be offered to vertical customers in the near future for satisfying their communication needs in the systematic digitalization of industrial segments. While mechanisms for provisioning those slices are progressively being introduced in operational networks, it is not yet resolved how the vertical customers will be able to control not only the functions composing the final service but also the underlying connectivity. In that manner, the vertical can have full control of the allocated resources as if the slice was actually a separate network. This paper proposes an architectural approach in that respect, considering some scenarios of applicability.

As future lines of work, it is necessary to incorporate into the provisioning lifecycle of the network slices the capability of control by the verticals, including the instantiation of the control entities here described (for functions and network) and their connection to the control entities maintained at the provider side. Such connection is expected to be done through well-known APIs (e.g. TAPI), also yet to be fully defined. Finally, it is necessary to ensure that the management from each vertical does not interfere other slices, that is, to ensure isolation mechanisms in this proposed architecture, including ways of resolving configuration conflicts at WIM level.

Abbreviations

B2B: Business-to-Business; B2C: Business-to-Consumer; B2B2C: Business-to-Business-to Consumer; CLAS: Cooperating Layered Architecture for Software-Defined Networking; CLI: Command Line Interface; COTS: Common-of-the-self; eMBB: Enhanced Mobile Broadband; FCAPS: Fault, Configuration, Accounting, Performance and Security; IC: Infrastructure SDN Controller; NBI: Northbound Interface; NFV: Network Function Virtualization; NFVI: NFV Infrastructure; NPN: Non-Public Network; NS: Network Service; NSH: Network Service Header; ONF: Open Networking Foundation; PNF: Physical Network Function; PoP: Point of Presence; SBI: Southbound Interface; SDN: Software Defined Networking; SFC: Service Function Chain; SlaaS: Slice-as-a-Service; TAPI: Transport API; TC: Tenant SDN Controller; UPF: User Plane Function; VIM: Virtual Infrastructure Manager; VNF: Virtual Network Function; WIM: WAN Infrastructure Manager.

Acknowledgements

This work has been partly funded by the European Commission through the projects 5G-PPP H2020 5GROWTH (Grant Agreement No 856709), EU-TW 5G-DIVE (Grant Agreement No 859881) and by the Spanish AURORAS (RTI2018-099178-B-I00) project. This information reflects the consortia views, but neither the consortia nor the European Commission are liable for any use that may be done of the information contained therein.

Authors' contributions

LMC performed the conceptualization and design of the solution, and the original draft preparation. RV contributed the experimental part. All authors were involved in the discussion of the work and the writing of the paper. All authors read and approved the final manuscript.

Funding

This work has been partly funded by the European Commission through the projects 5G-PPP H2020 5GROWTH (Grant Agreement No 856709), EU-TW 5G-DIVE (Grant Agreement No 859881) and by the Spanish AURORAS (RTI2018-099178-B-I00) project. This information reflects the consortia views, but neither the consortia nor the European Commission are liable for any use that may be done of the information contained therein.

Availability of data and materials

Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

Declaration

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Telefónica CTIO Unit, Madrid, Spain. [2]Universidad Autónoma de Madrid, Madrid, Spain. [3]Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA), Barcelona, Spain.

Received: 8 January 2021 Accepted: 16 April 2021

References

1. L.M. Contreras, V. López, O. González De Dios, A. Tovar, F. Muñoz, A. Azañón, J.P. Fernandez-Palacios, J. Folgueira, Toward cloud-ready transport networks. IEEE Commun. Mag. **50**(9), 48–55 (2012)
2. X. Foukas, G. Patounas, A. Elmokashfi, M.K. Marina, Network slicing in 5G: Survey and challenges. IEEE Commun. Mag. **55**(5), 94–100 (2017)
3. I. Afolabi, T. Taleb, K. Samdanis, H. Flinck, Network slicing and softwarization: a survey on principles, enabling technologies, and solutions. IEEE Commun. Surv. Tutor. **20**(3), 2429–2453 (2018)
4. R. Su, D. Zhang, R. Venkatesan, Z. Gong, C. Li, F. Ding, F. Jiang, Z. Zhu, Resource allocation for network slicing in 5G telecommunication networks: a survey of principles and models. IEEE Netw. **33**(6), 172–179 (2019)
5. A.A. Barakabitze, A. Ahmad, R. Mijumbi, A. Hines, 5G network slicing using SDN and NFV: a survey of taxonomy, architectures and future challenges. Comput. Netw. **167**, 106984 (2020)
6. E. Rojas, R. Doriguzzi-Corin, S. Tamurejo, A. Beato, A. Schwabe, K. Phemius, C. Guerrero, Are we ready to drive software-defined networks? A comprehensive survey on management tools and techniques. ACM Comput. Surv. **51**(2), 1–35 (2018)
7. E. Kaljic, A. Maric, P. Begovic, M. Hadzialic, A survey on data plane flexibility and programmability in software-defined networking. IEEE Access **7**, 47804–47840 (2019)
8. N.M.M.K. Chowdhury, R. Boutaba, A survey of network virtualization. Comput. Netw. **54**, 862–876 (2010)
9. R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, R. Boutaba, Network function virtualization: state-of-the-art and research challenges. IEEE Commun. Surv. Tutor. **18**(1), 236–262 (2016)
10. X. Zhou, R. Li, T. Chen, H. Zhang, Network slicing as a service: enabling enterprises' own software-defined cellular networks. IEEE Commun. Mag. **54**(7), 146–153 (2016)
11. L.M. Contreras, D.R. López, A network service provider perspective on network slicing. IEEE Softwarization (2018). https://sdn.ieee.org/newsletter/january-2018/a-network-service-provider-perspective-on-network-slicing

12. L.M. Contreras, C.J. Bernardos, D. Lopez, M. Boucadair, P. Iovanna, Cooperating layered architecture for software-defined networking (CLAS), RFC 8597, 2019.
13. ETSI GS NFV-EVE 005, Network functions virtualisation (NFV); ecosystem; report on SDN usage in NFV architectural framework, V1.1.1, 2015.
14. J. Halpern, C. Pignataro, Service function chaining (SFC) architecture, RFC 7665, 2015.
15. M. Boucadair, Service function chaining (SFC) control plane components and requirements, draft-ietf-sfc-control-plane-08 (work in progress), 2016.
16. P. Quinn, U. Elzur, C. Pignataro, Network service header (NSH), RFC 8300, 2018.
17. ETSI GR NFV-IFA 022, Report on management and connectivity for multi-site services, V3.1.1, 2018.
18. R. Rokui, S. Homma, K. Makhijani, L.M. Contreras, J. Tantsura, Definition of IETF network slices, draft-nsdt-teas-ietf-network-slice-definition-02 (work in progress), 2020.
19. L.M. Contreras, Ó. González, V. López, J.P. Fernández-Palacios, J. Folgueira, iFUSION: Standards-based SDN Architecture for Carrier Transport Network, in: *IEEE Conference on Standards for Communications and Networking (CSCN)*, Granada, Spain, 2019.
20. Telecom Infra Project, Open Transport SDN Architecture Whitepaper, November 2020. https://cdn.brandfolder.io/D8DI15S7/at/jh6nnbb6bjvn7w7t5jbgm5n/OpenTransportArchitecture-Whitepaper_TIP_Final.pdf
21. S. Clayman, F. Tusa, A. Galis, L.M. Contreras, WIM on-demand—a modular approach for managing network slices, in: *IEEE Conference on Network Softwarization (NetSoft)*, Ghent, Belgium, 2020.
22. T.V.K. Buyakar, A.K. Rangisetti, A. Franklin, B.R. Tamma, Auto scaling of data plane VNFs in 5G networks, in: *13th International Conference on Network and Service Management (CNSM)*, Tokyo, Japan, 2017.
23. C.H.T. Arteaga, F.B. Anacona, K.T.T. Arteaga, O.M.C. Rendon, A scaling mechanism for an evolved packet core based on network functions virtualization. IEEE Trans. Netw. Serv. Manag. **17**(2), 779–792 (2020)
24. ONF, TAPI Reference Implementation Agreement, v2.1.3, 2020. https://wiki.opennetworking.org/display/OTCC/TR-547+TAPI+v2.1.3+Reference+Implementation+Agreement
25. ETSI GR NFV-IFA 032, Interface and information model specification for multi-site connectivity services, V3.2.1, 2019.
26. J. Ordonez-Lucena, J. Folgueira, L.M. Contreras, A. Pastor Perales, The use of 5G non-public networks to support industry 4.0 scenarios, in: *IEEE Conference on Standards for Communications and Networking (CSCN)*, Granada, Spain, 2019
27. C. Parada et al., 5Gtango: a beyond-mano service platform, in: *2018 European Conference on Networks and Communications (EuCNC)*, Ljubljana, Slovenia, 2018.

Publisher's Note