Internet Research Task Force NFVRG G. Carrozzo, Ed.
Internet-Draft Nextworks
Intended status: Informational K. Pentikousis, Ed.
Expires: January 7, 2016 EICT
July 6, 2015

Recursive orchestration of federated virtual network functions
draft-felix-nfvrg-recursive-orchestration-00

Abstract

This document introduces a policy-based resource management and orchestration framework which aims at contributing towards the current namesake NFVRG near-term work items.

It describes key points of the recursive resource orchestration framework developed within the wider research area of federated virtual network function orchestration. The document also relates this effort with respect to other orchestration frameworks, thus addressing both the NFV research and practitioner communities.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Today's Internet is a concatenation of IP networks interconnected by many distributed functions integrated into a plethora of highly specialized middleboxes. These elements implement complex network functions like firewalls, NATs, DPI, traffic scrubbing, etc. The product is a quite complex and rigid internetworking system in which network administrators and users cannot easily determine what is happening to traffic flows as they go toward destinations. In the last decade networks, servers, storage technologies, and applications have all undergone significant changes with the introduction of virtualization, network overlays, and orchestration. Such technologies have allowed network operators and service providers to easily introduce a variety of (proprietary) hardware-based appliances in order to improve their network manageability as well as rapidly launch new services, keeping up with the pace of their users demand. Therefore, the current Internet looks like a concatenation of networks with many distributed functions, implemented via a plethora of highly specialized middleboxes which implement firewalls, DPI, NAT, traffic scrubbing, etc. [middlebox].

Software Define Networking and programmable virtualized network functions for flow processing are rapidly changing the current scenario, extending the support of network functions by virtualized and chained appliances beyond the virtual L2 switching over IP networks (e.g. VXLAN, GRENV, STT) and the basic LAN based flow pinpointing.

Virtual Functions of this kind, for network and non network (e.g. computing) tasks, are generally available in heterogeneous pools under different administrative domains, being them related to the hosting infrastructures in which they originate. It is emerging a need to interconnect, federate and implement policy control on these pools of virtual resources, in order to abstract different infrastructures, resources and functions, as well as procedures by physical operators and infrastructure owners. This can allow defining larger virtual overlays where different resources and functions are deployed, combined and handled in the form of virtual instances irrespective of the administrative domain and specific technology from which they originate. Examples of application contexts in which this federation of virtual function pools may occur are:

- o large scale experimentation over programmable networks, which allows to reserve slices of network and non-network resources from different federated providers to run experiments on network control, protocols and algorithms at large scale (e.g. [FELIX]);

- o virtual infrastructure operators, who intend to implement their network service offer over a completely virtual infrastructure in the form of virtual network nodes and functions, virtual servers and storage, etc., all procured as a service from physical providers.

This document discusses key points of the recursive resource orchestration framework developed within the wider research area of federated virtual network function orchestration. The proposed architecture allows federation and integration of different network and computing resources residing in a multi-domain heterogeneous environment across different providers. To achieve this, the architecture uses a combination of recursive and hierarchical configurations for orchestration, request delegation and inter-domain dependency management, with resource orchestrating entities (Resource Orchestrators, RO) responsible for synchronization of resources available in particular administrative domains.

This document is being discussed on the nfvrg@irtf.org mailing list.

2. Terminology

The following terminology is used in this document.

Virtual Network Function(VNF). One or more virtual machines running different software and processes, on top of industry standard high volume servers, switches and storage, or even cloud computing infrastructure, and capable of implementing network functions traditionally implemented via custom hardware appliances and middleboxes (e.g. router, NAT, Firewall, load-balancer, etc.)

VNF Island. A set of virtualized network functions and related network and IT resources under the same administrative ownership/control. A VNF island could consist of multiple zones, each characterized by a specific set of control tools & interfaces.

VNF Zone. A set of virtual network functions grouped for homogeneity of technologies and/or control tools and/or interfaces (e.g. L2 switching zone, optical switching zone, OF protocol controlled zone, other transit domain zone with a control interface). The major goal of defining SDN zones is to implement appropriate policies for increasing availability, scalability and control of the different resources of the island. Examples of zone definitions are available in popular Cloud Management Systems (CMS) like Cloudstack (e.g. refer to the Cloudstack Infrastructure partitioning into regions, zones, pods, etc., [cloudstack]) and OpenStack (e.g. refer to the infrastructure partitioning in availability zones and host aggregates [openstack]).

Transit network domains. The network domains use a Bandwidth on Demand Interface to expose automatically and on-demand control of connectivity services and, optionally, inter-domain topology exchange. In order to federate resources belonging to distant facilities (i.e. islands/zones) it must be ensured that interconnectivity is provided on-demand and with a specific granularity.

Slice. A user-defined subset of virtual networking and IT resources, created from the physical resources available in federated VNF Zones and VNF Islands. A VNF Slice has the basic property of being isolated from other slices defined over the same physical resources, and being dynamically extensible across multiple VNF Islands. Each VNF Slice instantiates the specific set of control tools of the specific zones it traverses.

Resource Orchestrator (RO). Entity responsible for orchestrating end-to-end network service and resources reservation in terms of compute, storage and network functions over the infrastructure, as

well as delegating end-to-end resource and service provisioning in a technology-agnostic way.

Resource Managers (RMs). Entity responsible for controlling and managing different types of resources and/or network functions.

3. Recursive orchestration in federated virtual environments

3.1. Problem Statement

The coordinates creation of a virtual environment with pools of virtual network and non-network functions from heterogeneous, multi-domain and geographically distributed facilities requires appropriate tools for resource and virtual function management and control capable of orchestration and policy control across VNF islands, zones and domains defined above.

Elements that belong to this control and orchestration layer operate in a hierarchical way (parent-child) for efficient multi-domain information management and sharing. This is generally referred as Inter-island Orchestration Space [FELIX-D2.1] [FELIX-D2.2].

Once the set of virtual network and non-network functions is determined, reserved and deployed across the different islands, the resulting virtual network environment is ready for being used as a User Space by any tool or application the user wants to deploy in it.

In the Inter-island Orchestration Space (see Figure 1), Resource Orchestrators (RO) are responsible for orchestrating end-to-end network services and resources reservations in the whole infrastructure. Moreover, ROs should be able to delegate end-to-end resource and service provisioning in technology-agnostic way.

ROs are connected to different types of Resource Managers (RMs), which are in turn used to control and manage different kinds of technological resources. For example, the VNF RM WAN side provides connectivity between L1/L2 network domains at the two ends. This management can be achieved using frame, packet or circuit switching technologies and should support different protocols.

On the other hand, the VNF RM (LAN side) manages the network infrastructure composed of SDN-enabled devices, e.g. OpenFlow switches or routers. In short, it can control the user traffic environment by updating flow tables in physical devices.

In addition, the Virtual Function pool RM for computing resources is responsible for setting up and configuring computing resources, i.e.

creating new virtual machine instances, powering on/off instances, network interface card configuration, etc.

Authentication and Authorization Infrastructure (AAI) for authenticating and authorizing users, is a cross layer function in the Inter-island Orchestration Space, because it serving as a 'trust anchor' to facilitate authN/authZ procedures in federated facilities.

Similarly, Monitoring allows to retrieve, correlate and abstract statistics from the different components of the physical and virtual resource pools and from the user's slice.

Figure 1 shows a parent RO coordinating orchestration actions at NFV island level under the responsibility of two child ROs, each orchestrating different types of RMs for the different kinds of virtual network and non-network function pools.

```
  +---+           +-------------------------------------------+    +---+
  |   |           |          RESOURCE ORCHESTRATOR - RO       |    |   |
  |   |-----------|                (parent)                   |----|   |
  |   |           +-------------------------------------------+    |   |
  |   |                    |                              |       |   |
  |   |                    |                              |       |   |
  |   |  +-------------------------------------+     +---------+   |   |
  | M |  |                  RO                 |     |    RO   |   |   |
  | O |--|               (child)               |     | (child) |---|   |
  | N |  +-------------------------------------+     +---------+   | A |
  | I |        |             |            |                |       | A |
  | T |        |             |            |                |       | A |
  | O |  +-----------+ +----------+ +----------+    +-----------+  |   |
  | R |  |  VF POOL  | | VNF POOL | | VNF POOL |    |  Virtual  |  |   |
  | I |--|  MANAGER  | | MANAGER  | | MANAGER  |    |    pool   |--|   |
  | N |  |(computing)| |(LAN side)| |(WAN side)|    | manager(s)|  |   |
  | G |  +-----------+ +----------+ +----------+    +-----------+  |   |
  |   |        |             |             |              |        |   |
  |   |  +----------+  +----------+ ----------+      +----------+  |   |
  |   |--|    VF    |  |    VNF   | |    VNF   |     |    VNF   |--|   |
  +---+  +----------+  +----------+ +----------+     +----------+  +---+
```

Figure 1: Recursive Orchestration architecture model

3.2. Resource Orchestrator

RO is the entity that orchestrates the different resources in the Inter-island Orchestration Space.

There are two different modes in which RO may operate:

- o Parent
- o Child

For an inter-island federation, RO operates in parent mode and attaches to child ROs, whilst in child mode ROs communicate with RMs.

One of RO's main objectives is to forward requests within the infrastructure, either by:

- o Passing user requests to the appropriate resource management systems (RMs) in the layer below, as with any hierarchical mode.

- o Proxying requests between other ROs in a recursive mode, depending on the federation policy that is configured for the domain where the RO is located. For this to work it is necessary to ensure similar interfaces for each orchestrator.

Key functions of the RO can be summarized as follows. RO manages the different VNF islands and users in terms of their resource and data access policies. It mediates between the user and the technology specific to a Resource Manager (RM) by means of delegation. It is expected that different RMs will handle, for example, technology-dependent aspects in SDN domains (VNF RM LAN side) and transit network domains (VNF RM WAN side), as well non-network resource pools. As part of this mediation, the RO will engage in the creation (provisioning), maintenance, monitoring, and deletion (release) of the used slices.

RO also maintains a high-level cross-island topological view, which summarizes the different resources pools available along with their inter-connections. This topology view is initialized and updated by the underlying RMs, thus implementing a distributed hierarchical resource discovery function. It determines which domains and which inter-domain resources should be used to instantiate a given end-to-end service for a particular experimenter's slice.

For example, based on a user request for a given type of service to be instantiated in two remote islands, parent RO determines which specific resource domains should be involved.

Finally, RO coordinates and ensures that the correct sequence of actions takes place with respect to the operation of the technology-specific RMs. This includes the provisioning of the slice resources as per user's requirements.

RO also collects and correlates status alarms and warnings on resources, either generated by the resources themselves or the services managing them. This is done on a per-slice basis and proceeds with reporting/notifying the corresponding users.

Different deployment models are possible for a Resource Orchestration entity:

- o hierarchical centralized (see Figure 2.A)
- o distributed in chain (see Figure 2.B
- o distributed full-mesh (see Figure 2.C)
- o hierarchical hybrid (see Figure 3)

The hierarchical hybrid model is deemed to guarantee the optimal trade-off between effectiveness of control, federation, trust adjacencies and scalability.

```
             +-----------+
             |     RO    |
             +-----------+
             /     |    \                 +------+   +------+   +------+
            /      |     \                |  RO  |---|   RO |---|  RO  |
           /       |      \               +------+   +------+   +------+
    +------+   +------+   +------+
    |  RO  |   |  RO  |   |  RO  |
    +------+   +------+   +------+
                (A)                                     (B)


                 +--------------------+
                 |                    |
             +------+   +------+   +------+   +------+
             |  RO  |---|  RO  |---|  RO  |---|  RO  |
             +------+   +------+   +------+   +------+
                 |         |                    |  |
                 |         +--------------------+  |
                 +---------------------------------+
                                (C)
```

Figure 2: RO deployment models

```
                +-----------+                    +-----------+
                |     RO    +--------------------+     RO    |
                +-----------+                    +-----------+
                /     |    \                    /     |    \
               /      |     \                  /      |     \
              /       |      \                /       |      \
       +------+   +------+   +------+     +------+   +------+   +------+
       |  RO  |   |  RO  |   |  RO  |     |  RO  |   |  RO  |   |  RO  |
       +------+   +------+   +------+     +------+   +------+   +------+
```

Figure 3: Hybrid RO deployment model

4. Policy-Based Resource Management

Aside from the main functions described above, each Resource Manager is also part of the Authentication and Authorization Infrastructure (AAI).

AAI provides the necessary mechanisms to authenticate and authorize users, as well as provide accountability. In order to realize these functions, our architecture suggests the implementation of a ClearingHouse (CH) , which establishes the root of a trust chain. This chain can then be used to verify the identity and privileges of all actors in this architecture.

By using a certificate-based approach, the architecture has flexibility to easily federate different VNF islands. By installing ClearingHouse certificates, actors can be verified against different ClearingHouses, and thus can utilize a multitude of resources.

A ClearingHouse (CH) comprises a set of related services supporting AAA operations. CH serves as a central location to lookup information about members, slices and other available services in the VNF island.

There are three groups of CH services:

o Registration and management services to lookup for available services in the facility as well as register new members, projects and slice objects.

o Authentication and Authorization services to manage the credentials of all entities and enforce predefined policies.

o Accountability services to facilitate tracking of all transactions.

These services are offered by CH with the help of the following functions and authorities.

- o The Member Authority (MA) is responsible for managing and asserting user attributes. It generates member certificates for identification purposes and credentials to specify the attributes and roles associated with each member. The MA maintains a database of registered members and their associated information including, but not limited to, certificates and credentials, SSH (Secure Shell) and SSL (Secure Sockets Layer) keys as well as the human readable identity information like real name, institute, contact details.

- o In addition, a Certificate Revocation List (CRL) can also be accessed from MA for use in certificate verification process.

- o The Slice Authority (SA) creates and manages slice objects and the associated member credentials (called slice credentials). Slice credentials map member roles and privileges on a slice object, i.e., slice credentials authorize user actions at aggregates within a slice context. SA also enables related operations on slice objects like look up, modify, renew, etc.

- o The Project Service (PS) hosted at SA maintains a list of existing projects and asserts the member roles.

- o The Service Registry (SR) serves as the primary network contact point as it keeps a record of all available registered services such as SA and MA and offers their URIs.

- o The Logging Service (LS) realizes accountability by storing the transaction details between user-agents and aggregate managers.

The user-agents and ROs can communicate with the CH through XMLRPC calls over a secured connection (SSL).

AAI is ultimately responsible for granting access to the resources, and can be further extended through policies, which are a set of rules defined by the administrators to implement an upper-level control on the resource usage (e.g. defining a maximum virtual memory value for a VM resource or a maximum number of flow spaces).

4.1. Certificate-based authN/authZ (C-BAS)

Since VNF pools are finite, access to virtual functions and resources should be policed according to set authorization levels throughout the life-cycle of each experiment.

Access control is also required to ensure that infrastructures remain operational.

Although a number of solutions for authentication (authN) and authorization(authZ), such as Kerberos and LDAP, already exist, they have several shortcomings: tight-coupling of authN/authZ mechanisms with the implementation of the architecture; little or no regard for re-usability (i.e., one authN/authZ architecture cannot be reused by a different infrastructure); and no support for a standard access interface between networks and the authN/authZ architecture.

C-BAS, certificate-based authN/authZ solution, is designed to serve all these requirements and include i) multiple authoritative source of trust, ii) flexible system of authorization, and iii) synchronization of authN/authZ entities to realize federations.

For example, the Registry Service of C-BAS may be exploited to implement load balancing and failover features. In addition, the evolved architecture of C-BAS makes it robust against disruptions and interference from attackers and enables support for various member roles and permissions.

C-BAS employs X.509 certificates and SFA styled credentials to realize AAA services.

The implementation of C-BAS is publicly available (www.eict.de/c-bas) and is based on eiSoil (github.com/EICT/eiSoil/wiki) thus exploiting its plugin capabilities that enable importing the functionality from one plugin module to another.

4.2. Resource Managers

4.2.1. VNF Pool manager functionality

A VNF pool manager is a functional entity in charge of controlling a specific type of VNFs, being the equivalent of the SFA Aggregate Manager (see [SFA]). As such, a VNF pool manager is a Resource Manager within the federation, capable of discovering resources, capabilities ans functions from physical infrastructure, abstracting them before publishing to the supervising RO and eventually capable of managing specific technology-specific configurations and provisioning towards the actual resource layer.

Whilst the northbound interface of the Resource Manager is abstract and unified across different technology domains (e.g. based on REST or XMLRPC), the southbound interface is based on the specific interfaces exposed by the different types of resources (e.g. OpenFlow, NETCONF, SNMP, CLI, OVSDB, etc.)

4.2.2. OpenFlow-based VNF pool manager

The VNF pool manager LAN side could be OpenFlow-based and provide the mechanisms to control the network infrastructure inside a domain with SDN-enabled hardware (typically, OpenFlow-enabled switches and routers). The Inter-island Orchestration Space architecture is agnostic of the physical network resources. For a SDN domain based on OpenFlow users can control network behaviour by actively updating the flow tables of the network elements. This update is usually done by a SDN controller, which is configured according the user's requirements. Typically, the controller will respond to an event generated by a network element, such as a flow establishment request, and update the flow tables appropriately.

For the network manager, the VNF pool manager LAN side will provide management functionalities for the overall network resources and virtual network functions in the LAN or data center. This element acts as a proxy between the resources and the SDN controller, and grants or denies the forwarding of control messages. The VNF pool manager LAN side provides functions to build a unique flow space for every experimenter so that traffic is isolated and distinguished from that of other slices (e.g. like FlowVisor or OVX do).

These functionalities can prevent issues arising when several users wish to use the same physical resources. In detail, a flow space can contain a range of differentiators: source or destination IPs or MAC addresses, TCP or UDP ports, for example.

One way to separate the traffic is assigning a VLAN tag to each packet. In this case, the special purpose controller inspects the incoming packet, identifies the VLAN tag and sends it to the corresponding SDN controller.

4.2.3. Stitching Entity VNF pool manager

The Stitching Entity VNF pool Manager is among the pool managers WAN side, and is in charge to control the Stitching Entity (SE), a network element providing necessary translation mechanisms for a slice setup on top of the L2 protocol stack performed in order to hide from a user the real complexity of the multi-domain WAN transport network.

The main responsibility of the SE is to provide at least one of the following network functions:

- o QinQ, to encapsulate slice traffic into a transport network Ethernet frames,

- o VLAN translation mechanism to hide from a user the actual VLAN tagging, used by carrier networks while interconnecting two or more VNF islands.

The SE RM communicates with an RO, a parent control entity, to i) advertise an internal topology and capabilities of the SE under its control, ii) receive requests, and iii) notify the RO about success and failure events.

A single SE RM must relate only to a single RO and must be implemented in each VNF island. A single SE RM is responsible for a single SE or a group of SEs, which belong to a network domain and act as an entry point to the island infrastructure.

4.2.4. Transit network VNF pool manager

The main responsibility of the Transit VNF pool manager (TN RM) is to support the FELIX architecture with network connectivity mechanisms within particular domains and between them.

In order to deliver the network services, it must be integrated with its southbound interfaces within a particular network domain. Such a domain can use different L1/L2 technologies and may be controlled by a Network Management System (NMS) or by specific interfaces or protocols that are technology-dependent, and unique in each case.

The Transit network VNF pool manager must communicate with an RO in order to i) advertise resources under its control, ii) receive requests, and iii) notify the RO about success and failure events.

A single TN RM must relate only to a single RO. A single TN RM is responsible for a group of particular network resources, which belong to a network domain and are usually managed by a single entity, i.e. a network administrator or NMS.

TN RM usually manages L1/L2 transport networks, which are composed of physical devices using frames/packets or circuit switching technologies and support different protocols, e.g. MPLS/GMPLS. In order to support inter-island connectivity between existing VNF islands, the TN RM also supports the management of VPN set up and tear down procedures.

The TN RM southbound interface can be based on Bandwidth on Demand interfaces, like GMPLS UNI or similar approaches.

4.2.5. RM for virtual computing

The function of the Computing Resource pool Manager (C-RM) is to provide a method to assign, set up and configure computing resources.

C-RM manages physical computing resources, and also the configuration of its own slicing mechanisms (e.g. common hypervisors or other virtualization stacks) and computer resources as presented to the user (OS images, network interface configuration, and so on).

The management of physical computing resources should provide a method for rebooting machines, remote control (of a machine's console), or hard power on/off of a machine experiencing problems, for example using a networked PDU (power distribution unit).

Management is typically performed only when problems occur, and when a slice is created, destroyed or modified. Migration of computing resources to other islands may also require reconfiguration. This includes the configuration of network interfaces of the computing resource, and setting the underlying resources (e.g. hypervisor, physical machine), such that those interfaces are bridged onto the correct physical interfaces. In particular, it may be necessary to configure a slicing mechanism in this bridging, in the case where multiple computing resources have to share a single physical interface. This would typically be achieved using a (software-based) SDN solution inside the virtualization platform. Once the SDN solution has been properly set up, it becomes an SDN resource, which is managed by the VNF pool manager LAN side.

5. Positioning w.r.t. existing Orchestration Frameworks

5.1. Openstack orchestration

Among cloud orchestration solution, OpenStack is the facto common reference through its Heat module [os-heat].

Openstack Heat implements an orchestration engine to launch multiple composite cloud applications based on templates in the form of text files that can be treated like code.

Many existing CloudFormation templates can be launched on OpenStack. Heat provides both an OpenStack-native ReST API and a CloudFormation-compatible Query API.

A Heat template describes the infrastructure for a cloud application in a text file. Infrastructure resources that can be described include: servers, floating ips, volumes, security groups, users, etc.

Templates can also specify the relationships between resources (e.g. this volume is connected to this server).

Heat also provides an autoscaling service.

Heat primarily manages cloud infrastructure, does not support federation and AAI is bundled in the OpenStack framework.

5.2. OpenMANO

OpenMANO is an open source project which implements the reference architecture for Management & Orchestration under standardization at ETSI's NFV ISG (NFV MANO) [openmano].

OpenMANO consists of two main SW components:

- o NFV VIM (Virtualised Infrastructure Manager) to provide computing and networking capabilities and to deploy virtual machines.

- o A reference implementation of an NFV-O (Network Functions Virtualisation Orchestrator), which allows creation and deletion of VNF templates, VNF instances, network service templates and network service instances.

OpenMANO does not support federation and AAI as of today.

5.3. Other orchestration approaches: federated SDN infrastructures for research experimentation

The FELIX project [FELIX] is part of an international research experimentation infrastructure strategy (in Europe under the Future Internet Research Experimentation - FIRE - framework), with a special focus on SDN and Network Service Interface (NSI) developed by the Open Grid Forum. FELIX is implementing federation and integration of different network and computing resources controlled via SDN and NSI in a multi-domain heterogeneous environment across, initially spanning Europe and Japan. FELIX consortium has designed and is implementing an architecture that extends and advances assets previously developed in other Future Internet projects (e.g. OFELIA), by realizing the federation concepts defined in SFA [SFA] with a combination of recursive and hierarchical orchestration, request delegation and inter-domain dependency management. Other research testbeds have been working over the past year on federation of SDN resources. Three of them are particularly relevant on the SDN area: OFELIA, FIBRE and GridARS.

The OFELIA project [OFELIA] established a pan-European experimental network facility which enables researchers to experiment with real

OpenFlow-enabled network equipment and to control and extend the network itself in a precise and dynamic mode. The OFELIA facility uses the OpenFlow protocol (and related tools) to support network virtualization and control of the network environment through secure and standardised interfaces. OFELIA consists of two layers. The physical layer is comprised of the computing resources (servers, processors) and network resources (routers, switches, links, wireless devices and optical components). Resources are managed by the OFELIA Control Framework (OCF). Furthermore, the control framework layer contains components which manage and monitor the applications and devices in the physical layer. Aggregate Managers and Resource Managers are components of this layer, which can be seen as the combination of three components: Expedient is the GUI and allows the connection and federation with different Aggregate Managers via its plugins; Aggregate Managers (AMs) enable experimenters to create both compute and network resources via the VT AM and OF AM respectively; Resource Managers directly interact with the physical layer, provisioning compute resources (OFELIA Xen Agent) or flow rules to establish the topology (FlowVisor).

The FIBRE project [FIBRE] federates SDN testbeds distributed across Europe and Brazil. The FIBRE-EU system builds on top of the OFELIA OCF and incorporates several wireless nodes based on commercial Wi-Fi cards and Linux open source drivers. Unlike OFELIA, the FIBRE infrastructure is managed by different types of control and monitoring frameworks (CMFs). FIBRE deployed two top-domain authorities, one in Brazil and one in Europe, to manage and own resources in the respective continents. These inter-connected authorities interoperate to allow the federation of BR and EU testbeds.

In Japan, GridARS [GRIDARS] provides a reference implementation of the Open Grid Forum (OGF) Network Services Interfaces Connection Service (NSI-CS) protocol standard. GridARS can coordinate multiple resources (services), such as a network connection, virtual machines and storage spaces, via the NSICS protocol. It provides experimenters a virtual infrastructure, which spans several cloud resources, realised by multiple management domains including commercial solutions. GridARS consists of three main components.

6. IANA Considerations

No IANA considerations are applicable.

7. Security Considerations

This document proposes a new architecture for resource and VNF orchestration for the design of which security features are of utmost importance to proceed to operational deployments. Frameworks for Security in SDN are applicable to this document and are discussed in literature, for example, in [SDNSecurity], [SDNSecServ] and [SDNSecOF]. Security considerations regarding specific protocol interfaces are TBD.

8. Acknowledgements

This work has been partially supported and funded by the European Commission through the FP7 ICT FELIX project (Federated Testbeds for Large Scale Infrastructure Experiments, grant agreement no. 608638) and the National Institute of Information and Communications Technology (NICT) in Japan. The views expressed here are those of the author only. The European Commission and NICT are not liable for any use that may be made of the information in this document.

9. Contributors

Authors would like to acknowledge (in alphabetical order) the following contributors who have provided text, pointers, and ideas for this document:

- B. Belter (PSNC, Poland)
- C. Bermudo (i2CAT, Spain)
- T. Kudoh (Univ. Tokyo/AIST, Japan)
- J. Tanaka (KDDI, Japan)
- B. Vermeulen(iMinds, Belgium)

10. Informative References

[cloudstack]
"Apache CloudStack documentation. Cloud Infrastructure Concepts", Available online at http://cloudstack.apache.org/docs/en-US/Apache_CloudStack/4.1.0/html/Admin_Guide/cloud-infrastructure-concepts.html.

[FELIX] "FELIX Project website", http://www.ict-felix.eu.

[FELIX-D2.1]
R. Krzywania, W. Bogacki, B. Belter, K. Pentikousis, T. Rothe, G. Carrozzo, N. Ciulli, C. Bermudo, T. Kudoh, A. Takefusa, J. Tanaka and B. Puype, , "FELIX Deliverable D2.1 - Experiment Use Cases and Requirements", Available online at http://www.ict-felix.eu., September 2013.

[FELIX-D2.2]
R. Krzywania, W. Bogacki, B. Belter, K. Pentikousis, T. Rothe, M. Broadbent, G. Carrozzo, N. Ciulli, R. Monno, C. Bermudo, A. Vico, C. Fernandez, T. Kudoh, A. Takefusa, J. Tanaka and B. Puype, , "FELIX Deliverable D2.2 - General Architecture and Functional Blocks", Available online at http://www.ict-felix.eu., February 2014.

[FIBRE] T. Salmito, L. Ciuffo, I. Machado, M. Salvador, et al, , "FIBRE - An International Testbed for Future Internet Experimentation", 32th Simposio Brasileiro de Redes de Computadores e Sistemas Distribuidos (SBRC'14) , 2014.

[GRIDARS] [15] A. Takefusa, H. Nakada, T. Kudoh, Y. Tanaka and S. Sekiguchi, , "GridARS: An Advance Reservation-based Grid Co-allocation Framework for Distributed Computing and Network Resources", Lecture Notes, Computer Science of the Job Scheduling Strategies for Parallel Processing (JSSPP) vol.4942, pp. 152-168, April 2008.

[middlebox]
A. Greenhalgh, F. Huici, M. Hoerdt, P. Papadimitriou, M. Handley, and L. Mathy, , "Flow Processing and the Rise of Commodity Network Hardware", ACM SIGCOMM Computer Communication Review Volume 39 issue 2, April 2009.

[OFELIA] M. Sune, L. Bergesio, H. Woesner, T. Rothe, A. Kopsel, et al., , "Design and implementation of the OFELIA FP7 facility: The European OpenFlow testbed", The International Journal of Computer and Telecommunications Networking , December 2013.

[openmano]
"OpenMANO", Available online at https://github.com/nfvlabs/openmano/wiki.

[openstack]
"Scaling Openstack", Available online at http://docs.openstack.org/openstack-ops/content/scaling.html.

[os-heat] "OpenStack Orchestration - Heat", Available online at https://wiki.openstack.org/wiki/Heat.

[SDNSecOF]
Kloti, R., Kotronis, V., and P. Smith, , "OpenFlow: A Security Analysis", 21st IEEE International Conference on Network Protocols (ICNP) pp. 1-6, October 2013.

[SDNSecServ]
Scott-Hayward, S., O'Callaghan, G., and S. Sezer, , "SDN Security: A Survey", IEEE SDN for Future Networks and Services (SDN4FNS) pp. 1-7, 2013.

[SDNSecurity]
Kreutz, D., Ramos, F., and P. Verissimo, , "Towards Secure and Dependable Software-Defined Networks", Proceedings of the second ACM SIGCOMM workshop on Hot Topics in Software Defined Networking pp. 55-60, 2013.

[SFA] L. Peterson, R. Ricci, A. Falk and J. Chase, , "Slice-based Federation Architecture (SFA) v2.0", , July 2010.

Authors' Addresses

Gino Carrozzo (Ed.) (editor)
Nextworks
via Livornese 1027
Pisa 56122
Italy

Email: g.carrozzo@nextworks.it

Kostas Pentikousis (Ed.) (editor)
EICT
Torgauer Strasse 12-15
Berlin 10829
Germany

Email: k.pentikousis@eict.de