NFV ISG PoC Proposal – CloudNFV Open NFV Framework

1 NFV ISG PoC Proposal

1.1 NFV PoC Project Participants

Include additional manufacturers, operators or labs should additional roles apply.

PoC Project Name: CloudNFV Open NFV Framework Project

Network Operators/ Service Providers:

Telefonica		Contact: Diego Lopez (diego@tid.es)
Sprint		Contact: Farid (Fred) Feisullin (Farid.Feisullin@sprint.com)
Architect/Coordinator (Vendor): CIMI Corporation		Contact: Tom Nolle, tnolle@cimicorp.com
Vendor B:	6WIND	Charlie Ashton, Charlie.Ashton@6WIND.com
Vendor C:	Dell	Wenjing Chu, Wenjing_Chu@DELL.com
Vendor D:	EnterpriseWeb	Dave Duggal, dave@enterpriseweb.com
Vendor E:	Mellanox	Eran Bello, eranb@mellanox.com
Vendor F:	Metaswitch	Martin Taylor, martin.taylor@metaswitch.com
Vendor G:	Overture Networks	Mark Durrett, mark.durrett@overturenetworks.com
Vendor H:	Qosmos	David Le Goff, david.le-goff@qosmos.com
Vendor I:	Huawei	John Strassner, strazpdj@gmail.com

The CloudNFV core configuration is made up of 6WIND, Dell, EnterpriseWeb, Overture, and Qosmos. CIMI Corporation contributed the architecture and Huawei has provided special expertise in integrating with the TMF GB922 and GB942 framework.

We are open to inclusion of other vendors/operators in our process subject to resource constraints.

1.2 PoC Goals

At the high level, our goal is to provide implementation experience to the ISG relating to both high-level and specific issues that impact (or are impacted) by the specification process. These include:

- PoC Project Goal Number One: Demonstrate a framework for the application of NFV principles to actual service provider service creation, deployment and management practices. We propose to demonstrate four CloudNFV service processes—NodeBuilder, ServiceBuilder, ServiceBroker, and ServiceManager—and to align these both with NFV use cases and architecture and with OSS/BSS/NMS systems and service/operations practices. The goal is to provide a way of visualizing the potential impact of NFV on service velocity and agility, management efficiency and opex, and reliability/availability.

- PoC Project Goal Number Two: Identify the optimum way of describing the structure of a service in an NFV environment, including the definition of the NFVs, NFV connectivity with other NFVs, NFV connectivity with the overall end-to-end service, and NFV connectivity with "Infrastructure Services" that may have been pre-deployed and are shared by multiple services. We have implemented a model that composes low-level structures ("Nodes") that can represent not only VNFs or VNF packages, but also devices and services. We believe that many of the suggestions on describing VNF deployment and connection would not describe enough about either to permit them to be used with existing software elements that could provide network functionality—particularly open-source. That would mean that this software base could not be easily used to support NFV goals. VNF connectivity **must** define the network environment that the VNFs were written to run in; anything else is either superfluous or insufficient.

- PoC Project Goal Number Three: Demonstrate the optimum way of deploying a service, suitable to apply to VNF deployment in the cloud, on virtualized servers, on bare metal servers, and in network devices or CPE, including chips such as the EZChip or other network processors. We believe that a service will cross over between cloud-deployed VNFs, dedicated-server VNFs, VNFs that run on special blades in devices, etc. It is critical that all of these hosting options be supported or some of the valuable capabilities of specialized hosting may be lost, and cost savings reduced.

- PoC Project Goal Number Four: Establish the optimum way of defining management practices as "co-functions" of the service, composed in the same way and at the same time, and representing the service and its elements in the optimum way based on one or more existing sets of operations practices, or in a new way optimized for the virtual world. The functionality created by combining VNFs defines not only how a service works but how it should/must be managed, and the only way to harmonize management with functionality and at the same time create a link between virtualized elements and real services is to link them at the time of deployment by composing both side-by-side.

- PoC Project Goal Number Five: Develop the requirements for an NFV-compliant data center, including the hardware issues that would impact performance and stability, the platform software issues, and the mechanism for virtualization. We have created a harmonized, optimized, model for an NFV data center but have also learned the lessons of how that can be extended. We have already added a new vendor to our Active Data Center structure (Mellanox) and we are looking to integrate others. Our practices here can be a guide for NFV deployment in general.

- PoC Project Goal Number Six: Define the requirements for authoring VNFs and making them portable across multiple implementations. We have defined an architecture that does not require VNFs be authored to a specialized set of APIs, but we can support an arbitrary "Platform-as-a-Service" developer framework providing that the NFV user has licenses for the APIs involved and that they can be exposed as network interfaces (a URL, for example). Our goal from the first was to be able to create a VNF by "wrapping" any software component that exposed an interface, any hardware device, or any deployed service, in a CloudNFV description that would allow it to be deployed and managed using our model. That means that all these things, including existing devices and services, look like NFV to us, which is critical in transitioning to VNF-based service features.

- PoC Project Goal Number Seven: Determine the attributes for VNFs and VNF packages that would make them suitable/optimum in achieving the requirements defined by the ISG. For example, we have determined in our use of Metaswitch Clearwater IMS how VNF connectivity must be described and managed and what is needed to make a VNF horizontally scalable. We evaluated existing open-source software to meet the ISG use-case goals, and found Clearwater IMS from Metaswitch. The package was not intended to be a VNF, and we were able to deploy it without any changes to the software, largely because it has attributes that make it easy to convert into a VNF. We can describe those attributes and thus help guide selection of candidate software, and provide specification goals for real software.

- PoC Project Goal Number Eight: Evaluate the adequacy of explicit and implicit steps to assure openness from the NFV ISG work, and recommend additions/modifications. We seek to establish whether the current model for NFV and the current functional separation is adequate to insure openness and optimum in addressing the functional requirements set by the two white papers. We believe that our "totally open" approach allows the ISG to consider whether additional interfaces or functional separation would help advance the ISG's goals by encouraging component substitution, helping to break down proprietary barriers already being built.

1.3 PoC Demonstration

The CloudNFV PoC is already being hosted in Dell's Cloud in Santa Clara, California, USA and can be accessed directly from Dell's lab there or remotely from any location with reliable Internet service. We propose to use the latter capability to demonstrate the PoC at industry events in 2014, yet to be finalized but to include selected ISG and TMF activities as well as industry trade shows.

A1.4 Publication

All PoC results will be published on the CloudNFV website (http://www.cloudnfv.com) and also on the websites of the participating vendors. Results will be published under the Creative Commons License to encourage free distribution and attribution. Specific tests to validate the goals listed here will be scheduled openly and outside participation in viewing the tests and reviewing the results will be encouraged.

Test and publication will be incremental; each time a goal is met we will issue a report documenting our findings and publish the report as indicated here.

1.5 PoC Project Timeline

- What is the PoC start date? The CloudNFV project is already underway.
- (First) Demonstration target date: January 15 2014
- First PoC Report target date: February 15, 2014
- When is the PoC considered completed? We have defined this as an open-ended process and hope to continue to extend it in functional scope and participation.

2 NFV PoC Technical Details

2.1 PoC Overview

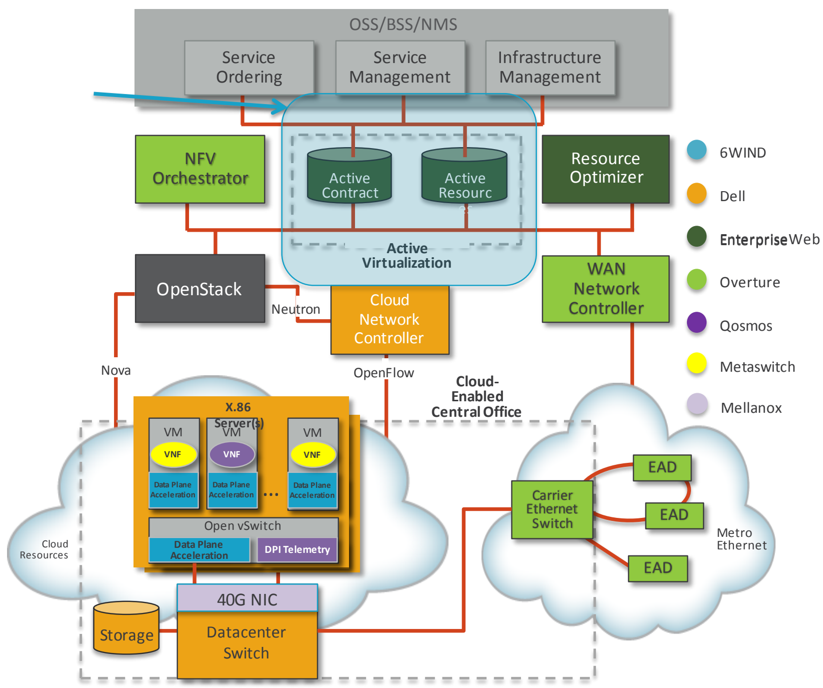
Our PoC is based on an implementation of NFV that has been evolving since April 2013. The architecture was designed to align functionally with both the TMF and ISG frameworks. We exercise this architecture using the Metaswitch Project Clearwater IMS project, which provides open-source code that conforms to modern cloud practices and is inherently scalable and resilient as the ISG requires.

At a high level, CloudNFV is a data-model-driven implementation of NFV based on a flexible software platform that provides us considerable latitude in creating "virtual" databases from information collected from a wide variety of sources, including MIBs. This flexibility means that we can present nearly any standard or open interface at any appropriate point with only minimal development. It is that capability that has allowed us to implement the NFV specification while that specification is still a work in progress. We have made some assumptions on interfaces and data structures but we can conform to the final specifications published by the ISG as they become available.

The implementation of CloudNFV is running in Dell's Solution Center in Santa Clara, and this center is open for inspection and hands-on demonstration subject to scheduling of the activity and the availability of a suitable space for meeting. However, it is not necessary to run CloudNFV from that location; our testing and development has been done remotely from the sites of the various participants. With adequate time and credentialing we can support remote access by ISG leadership to review data models, results, etc.

CloudNFV was designed to support multi-facility federation, and we are prepared to extend the scope of the PoC (in time and functionally) to incorporate other facilities provided by network operators or other sanctioned partners. In general, each such facility is represented to CloudNFV by a Service Model Handler that translates modelled service elements into resource commitments. The development requirements for a Service Model Handler depend on the nature of the resources in the secondary facilities, and we would have to review whether such requirements could be resourced prior to committing to use secondary facilities.

The following diagram outlines the architecture of CloudNFV:



We have provided an implementation that is a superset of the ETSI CloudNFV architecture:

- It extends "upward" to define services end to end, integrates with the TMF GB922 data model for Products, Services, and Resources. This is created by our Active Contract and Active Resource structure, and we do it so that we can more easily integrate with OSS/BSS systems and so that we can reflect service-level data into deployment of VNFs, which we believe to be critical in unifying VNF deployment and management with actual service-order flows. This architecture also accelerates service creation/agility by unifying NFV and TMF management. Finally, it accommodates any distribution of service/resource information across actual repositories or sources, since all data structures are virtual.

- It exercises "derived operations" in a totally composable way based on the TMF GB942 "NGOSS Contract" model. Every "Node" in a service structure defines its own Management Visualizer, which is essentially a totally flexible MIB. The Visualizer defines management variables and links each one to a derivation that can parse down the chain of the service structure and across into the resources that are committed to the service. Because the definition of variables/derivations is totally flexible, any convenient management representation can be assigned to any Node. All Management Visualizers can express their variables through any convenient API; we visualize to a GUI in the demo for convenience.

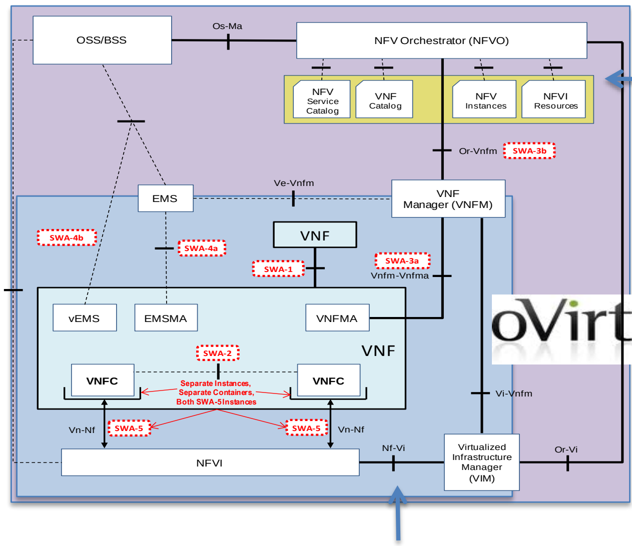
- It integrates with network and IT resources using a highly flexible model similar to that of OpenStack, but extensible far beyond OpenStack. A Node is linked to deployment on a resource pool by a pair of URIs. One defines the deployment/connection process needed, and this is the Service Model Handler, and the other points to the resource pool itself—the Dell cloud in our demo. Our current Service Model Handler operates through OpenStack, but we can define multiple handlers and associate a Node with any one of them. We plan to deploy Handlers that operate through OpenDaylight, for example, and have just issued a Call for Contribution to our project Group on LinkedIn.

The model is completely open; we can expose all of the ISG-sanctioned interfaces but we also support open substitution of functionality and hardware at every level of our own structure. All software components are referenced through URL/URI links, and all hardware resources likewise. You can see all of the places in the XML where we define processes and resources by URI; this is to make any of these components open. We will allow component substitution at any such point.

Our openness is demonstrated by two very specific points. First, we can on-board any software component that exposes its functionality through interfaces and can be managed through a standard management port/MIB. There are no special APIs required, so CloudNFV VNFs could be ported to other similarly open NFV implementations. So far, all of the NFV implementations that address on-boarding in any specific way and that provide management/orchestration capability will require special APIs to control integration of VNFs into MANO. We do not. This means that we can draw immediately on open-source tools without requiring forking of the projects and modification of code. Second, we have **extensive** documentation on integration with our platform and an open program to support it. We hope to use our open framework to rapidly expand our initial use case to a broader set of ISG use cases, and to include more and more vendors in the process. We have hundreds of PPT slides and hours of video tutorials on NFV in general, our own architecture, and integration with us. We have whole presentations on our management model, for example.

1.2.1 PoC Goal Mapping to NFV Documents

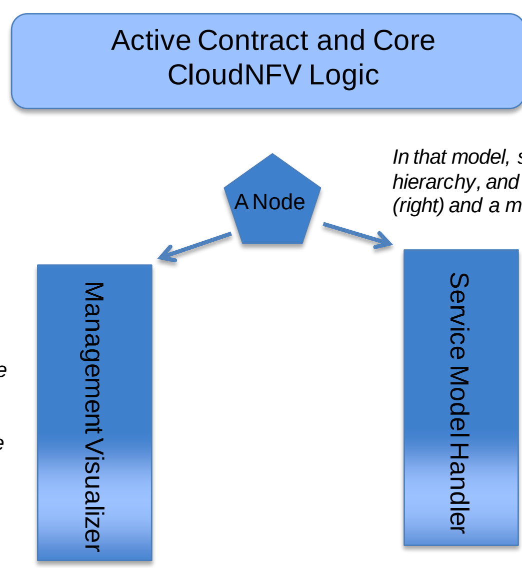
Our goals can be mapped to the NFV Architecture document just released, as the following series of diagrams shows. First, CloudNFV defines two functional layers of a service, one that is logical and virtual and is sustained by our Active Contract data structure and core architecture, and the other by a series of extensible Service Model Handlers that link the abstract model of a service to a specific instantiation on resources.



The outer box represents the boundaries of the core logic of CloudNFV, which are the functions that operate on and from Active Contract

The inner box represents the management, resource, and Service Model Handler elements of CloudNFV

Second, our structure of a service is one that consists of "Nodes" that represent deployable elements or collections thereof. Each Node defines both a path of deployment (the Service Model Handler) and a path of management (the Management Visualizer). These create the "Orchestration" and "Management" parts of the ISG MANO specification, in a functional sense, as seen below:



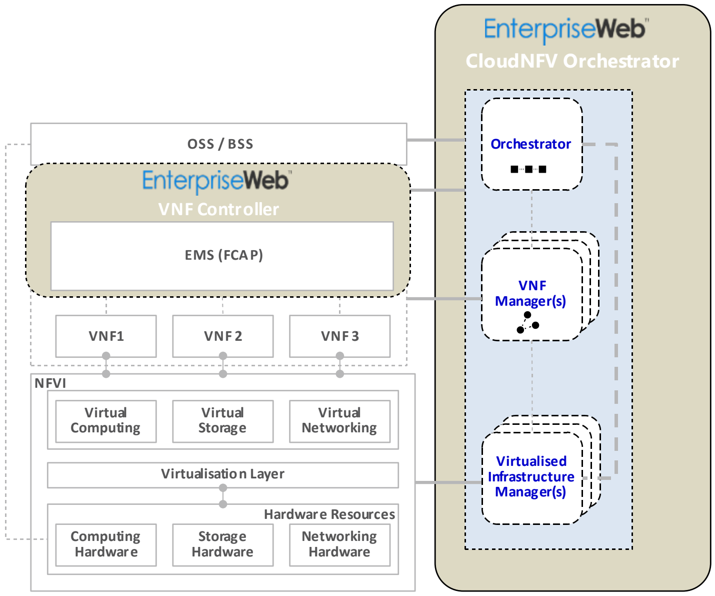
Deployment and orchestration tasks take place here, and this is where the model of the service resides

In that model, service elements are represented by a node hierarchy, and each node defines both a service dimension (right) and a management dimension (left)

The Management Visualizer is responsible for presenting resource information from Active Resource in a format and via an interface that the Architect selects. This information can be used directly (via a GUI) or presented to a management system

The Service Model Handler is responsible for deploying and connecting the elements of the Node that references it. This is what interfaces to things like OpenStack or Open Daylight, so all of the deployment technologies in the cloud, virtualization, and SDN connect here

Third, this conception of a split orchestration and management model maps to the ISG architecture as shown here:



Modified Version of "NFV reference architectural framework", ETSI GS NFV 002 V1.1.1 October 2013, Figure 4

2.2 PoC Scenarios

Describe the high level scenario(s) that will be demonstrated. When require, provide a dedicated network diagram.

Scenario 1 – Build up an inventory of deployable elements, both virtual functions and devices/network services, using the CloudNFV NodeBuilder function, and demonstrate how an Architect can frame the onboarding of VNFs and the inclusion of legacy network components for composition into services.

Scenario 2 – Using the Node inventory created by the first scenario, demonstrate how an Architect can build retail or wholesale service offerings using the ServiceBuilder, structured according to TMF GB922, and how resource requirements and interface connectivity is described/defined.

Scenario 3 – Using the Service inventory created by Scenario 2, demonstrate how the ServiceBroker processes service order is processed, service-specific parameters are included, and the service is deployed onto resources with proper management connections enabled.

Scenario 4 – Using the ServiceManager component of CloudNFV, demonstrate how deployed services from the ServiceBroker are supported through the management lifecycle, including how resource management data is correlated with service operations, how policy management controls the way that service logic can impact shared resources, and how historical data is made available for predictive service analytics.

2.3 Mapping to NFV ISG Work

	Use Case	Requirement	E2E Arch	Comments
Scenario 1	UC#5	Virtualization, Coexistence	Sect 4.2	We demonstrate an end-to-end model of NFV using UC#5, illustrating virtualization of IMS functions and coexistence of NFV and non-NFV components

Scenario 2	UC#5	Open architecture	Sect 4.2	We demonstrate interfaces open at the ISG reference points and the value of offering other open interfaces ad hoc
Scenario 3	UC#5	Access to virtual function logic	Sect 4.2	We demonstrate techniques to reduce or eliminate the need to customize function logic for NFV use
Scenario 4	UC#5	Coexistence	Section 5.2	We demonstrate an architecture and data model compatible with both NFV-hosted service components and legacy components end to end
Scenario 5	UC#5	Integrated Management	Sect 5.2	We demonstrate a management model that manages any combination of NFV-hosted and legacy service elements based on TMF GB922 and GB942
Scenario 6	UC#5	Coexistence	Sect 5.2	We demonstrate an agile mechanism for controlling/deploying SDN and proprietary network elements in an open and extensible way
Scenario 7	UC#5	Virtualization	Sect 6.2	We demonstrate a complete end-to-end service data model and relate its elements to the parameters defined by the ISG
Scenario 8	UC#5	Service graphing	Sect 6.2	We demonstrate an effective way of describing function connection via interfaces that includes but does not mandate the use of graphing protocols/descriptions
Scenario 9	UC#5	Interfaces	Sect 7.2	We demonstrate a way of defining the interfaces and connectivity of virtual and real components that allows the data model to drive existing cloud and NMS interfaces for connection of elements
Scenario 10	UC#5	Management Models	Sect 7.2	We demonstrate four classes of management model to explore whether any, alone, would fully address NFV management needs
Scenario 11	UC#5	Infrastructure	Sect 7.2.4	We demonstrate a modelling of NFVI and the linkage between this model and the contract/service model
Scenario 12	UC#5	Infrastructure management	Sect 7.2.4	We demonstrate the application of IETF i2aex in NFV to improve network stability and to exercise

				policy control over attempts to control shared resources
Scenario 13	UC#5	Infrastructure	Sect 7.2.4	We demonstrate principles for the creation of a fully NFV-optimized data center
Scenario 14	UC#5	Scalability and Performance	Sect 7.2.4	We demonstrate via load testing the impact of traffic on optimized NFV hosting environments
Scenario 15	UC#5	Virtualization	Sect 7.2.4	We demonstrate the value of using cloud-ready open-source elements of IMS to fulfil UC#5
Scenario 16	UC#5	DPI and traffic sensitivity		We demonstrate the interplay between software logic and ISG requirements for scalability and availability

To aid in aligning our activities with the work of the ISG, we offer the following more detailed description of our activity referencing the sections of the architecture document.

Section 4.2 Summary of NFV Objectives

1. We seek to address the value of fitting NFV into current carrier cloud, SDN, and end-to-end service management and OSS/BSS frameworks as a means of guaranteeing that the capital efficiencies gained by NFV are not lost to operations cost increases. We do this by defining an open model of a “Node” that can envelope anything from a VNF to a real and running service, and impose management and deployment practices uniformly across all Nodes regardless of content. Operational practices are the same for all Nodes.

2. We seek to demonstrate a broader commitment to agile and open interfaces will provide enhanced decoupling of components and improved openness and participation. Every piece of logic, every resource, is represented by an open interface and supports extension through substitution, as long as the functional requirements at that point are met.

3. We week to demonstrate that mechanisms aimed at reducing the customization needed to create virtual functions for hosting, to the point of opening the opportunity to deploy **any** cloud-compatible software component, would enhance all of these goals. Our model of “wrapping” existing software to create VNFs means any software for which the operator has a license (or any open source software) can be made into a VNF. That means that out of the box, CloudNFV has literally millions of candidate packages that can become VNFs, not only to supply network features but even to provide application features. We could compose a service combining hosted network features, real network services, and cloud application components like CRM.

Section 5.2 High-Level NFV Framework

1. We seek to demonstrate an end-to-end, user-to-user, cloud-compatible architecture that is fully open an integrable with network, IT, and virtual function resources can implement the NFV framework as described. We believe that only an NFV data model that can describe the total service can guide the

operational integration of NFV with other pieces of the network, out to the customer level. We can show that this is possible without compromising the NFV-specific aspects of deployment and management.

2. We seek to demonstrate that a service data model can describe services across this complete framework and integrate management processes between NFV and other portions of the service. Absent a place where a service is completely described, service management is not possible and all operations practices risk becoming siloed by technology.

3. We seek to demonstrate that this architecture can be coupled to network and IT infrastructure in a way that is compatible with public cloud principles and SDN principles, but also agile enough to support current proprietary hardware in both the network and IT sense. Our notion of a Service Model Handler, specified independently for every Node and optionally for every Interface, is an agile way to link service models to deployment and management, and can make ANYTHING an NFV resource.

Section 6.2 Virtualization of Function Blocks for Network Services

1. We seek to define a data model for the entire end-to-end service shown in Figure 2. The ISG has correctly modeled a service end-to-end but does not propose a data model or management/deployment process that can extend to that scope. We believe that such a model can be constructed, and that doing so will avoid serious management/integration problems.

2. We seek to demonstrate a modeling mechanism to describe service graphs in any suitable graphing language or technique. A graphing language does not facilitate deployment; fixation on any single approach can make it more difficult to accommodate multiple implementations with different requirements sets. In addition, it is not clear that such languages encourage connectivity definitions in a form that would actually be suitable to describe the network setup needed to link VNFs. We have examples of graphic models that would, if deployed, connect nothing successfully.

Section 7.2 Architectural Function Blocks

1. We seek to demonstrate a mapping of Figure 4 to a running implementation of NFV and to expose/address the issues related to all of the interfaces shown and to other possible points of functional interconnection and integration, including but not limited to the use of "federated" components provided by other public cloud, network operator, or NFV providers, including providers of NFV services based on other implementations. Our model allows resource pools (clouds) to be identified through a URI, so we can deploy on any cloud for which we have a compatible Service Model Handler. We can also define one or more Nodes in a service to be "Proxy" Nodes that represent a service component to be provided by another NFV implementation, and post an order to the other system to secure the resources there. We are pursuing a Federated Services integration with another NFV implementation at this time; several calls have already been held.

2. We seek to demonstrate multiple management models for VNF services, including:

    a. A "Virtual Device" model. We can reproduce a current device MIB.

    b. A "Resource-Centric" model. We can manage resources to meet resource-level SLAs and deal with FCAPS processes only on the resource pool, inferring service state from resource state.

c. A "Service-Centric" model. We can manage services using service-specific Nodal management models that have functional rather than device associations so that explicit service state can be obtained without binding management to the notion of old device models.

d. An arbitrarily composed functional management model. Any "pseudo-MIB" can be defined if it makes management more convenient and efficient.

e. The value of TMF GB922 and GB942 management principles. We believe that we demonstrate that the TMF GB922 and GB942 principles bring great value to the NFV MANO processes.

Section 7.2.4 NFV Infrastructure

1. We seek to demonstrate a data modeling of NVFI based on TMF GB922 and Logical Resource principles, that can support the entire end-to-end service and provide consistent management between NFV-created portions and other portions. We have Service, Resource, and Logical Resource concepts in CloudNFV, and we can define "Infrastructure Services" that are components of other services deployed in multi-use mode. In the Frankfurt presentation by DoCoMo, they cited the value of these multi-use services, but the current ISG model does not describe them.

2. We seek to demonstrate the value of IETF Infrastructure to Application Exposure (i2aex) principles in NFV. We use i2aex to proxy all management information so that we have the only direct connection to resource/application MIBS. This prevents stability/security problems with infrastructure control but is not required by the ISG.

3. We seek to demonstrate configurations and optimal hardware/software requirements for hosting and interconnection within an NFV data center. We have practical guidance for assembling NFV data centers and selecting components, and also for integration of these resources.

4. We seek to validate NFVI assumptions under systematic generated-load testing in parallel with conference video. We are upgrading or lab configuration to include 40G MIBs and are introducing test data generation—in a flexible "as-a-service" way.

Use Case Mappings

1. We will be demonstrating an IMS-Project Clearwater deployment. The Clearwater framework is an exceptional demonstration of the relationship between ISG goals and requirements and the interplay of VNF and MANO functionality.

2. We also plan to demonstrate DPI as a service, and show at least one step toward EPC functionality by demonstrating data-plane handling of SBC traffic. Adding SBC functionality to Clearwater provides the traffic-steering needed to address a virtual implementation of EPC, and the integration of CDN and EPC functionality. DPI as a service, like testing/monitoring as a service, are examples of the use of CloudNFV principles to compose complex services that virtualize resources normally considered to be fixed.

3. We are exploring further demonstrations in Service Chaining. We are looking for a service chaining example that crosses between devices and servers to demonstrate the interworking of legacy and NFV-specified elements.

2.4 PoC Success Criteria

We propose to report on the functionality and performance demonstrated for each of our goals and in each area of alignment. Our goal is to establish the value of the architecture and test the specific hypothesis identified in this document; success will be achieved if we have done so, even if the result proves that changes to our approach, to the ISG specifications, or to the TMF documents would be advisable.

2.5 Expected PoC Contribution

Please List of PoC Intended Contributions to Specific NFV Groups:

1) PoC Project Contribution #1: Optimum Strategy to Achieve Openness, to NFV Group TSC
2) PoC Project Contribution #2: Recommendations on VNF/VNFC Structuring to Optimize Use of Existing and Open Source Software, to NFV Group SWA
3) PoC Project Contribution #3 Optimum Data Model for Describing NFV and Combined Services, to NFV Group MANO.
4) PoC Project Contribution #4 Optimum Infrastructure Design for VNF Hosting and Management, to NFV Group INF.

We will provide an initial assessment of all four of these areas in a single preliminary "Plenary" report approximately one month after the project launches.

The following provides a higher level of detail on each contribution:

1. We will contribute a general report on the optimum way to achieve openness in NFV, both at the level of the network functions and at the level of the software used to deploy and manage these functions.
2. We will make a contribution to the SWA group making recommendations on the structuring of VNFs, the conversion of existing software into VNFs, and the authoring of new VNFs.
3. We will make a contribution to the MANO group making recommendations on the optimum data model for NFV, the best interfaces to expose to maximize openness and utility, and the ways in which NFV can be coupled to OSS/BSS/NMS systems and practices.
4. We will make a contribution to the INF group making recommendations on the reliability and performance implications of various hardware and software choices at the platform level.