31-01-2018

Deliverable D8.1 Proposed Integrated Services Framework and Network Services Development Roadmap

Deliverable D8.1

Contractual Date:	31-12-2016
Actual Date:	31-01-2018
Grant Agreement No.:	731122
Work Package/Activity:	8/JRA2
Task Item:	Task 1, 2, 3 and 4
Nature of Deliverable:	R (Report)
Dissemination Level:	PU (Public)
Lead Partner:	NORDUnet
Document ID:	GN4-2-17-142848
Authors:	Jerry Sobieski (NORDUnet), Ivana Golub (PSNC)

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 731122 (GN4-2).

Abstract
The Integrated Services Framework (ISF) proposes a set of generic, fundamental service classes that are needed in the emerging network ecosystem and defines a common service model that allows such services to invoke one another and to build incrementally more complex services. This Integrated Services Framework incorporates work carried out in GÉANT4-2 Network Services Development JRA2 Tasks 1-4.

Table of Contents

Table of Figures

Executive Summary

Virtualisation technologies and agility in service delivery is changing the way network services are being engineered. Defining basic network services or functions and using them to compose more complex and bespoke services is the approach taken by GÉANT, in order to utilise advanced capabilities of the network infrastructure and at the same time deliver customisable services and control to the users.

For the evolution of an integrated GÉANT services framework, several principles of engineering are employed: virtualisation, service abstraction, technology agnosticism, object-oriented inheritance, composition and construction.

The work references the TM Forum ZOOM architecture [TMFZOOM1], [TMFZOOM2], TMF Frameworx [TMFFx] when related to OSS/BSS processes and TMF SID for the information framework [TMFSID] the ETSI NFV architecture [NFV] regarding the service modelling, orchestration and function chaining. The work also references leading work from the R&E networking community such as the Network Services Interface (NSI) standard, the Generalised Virtualisation Model and draws on considerable work from prior future internet research programmes, such as FIRE and GENI. The approach impacts the work of four GN4-2 JRA2 tasks: T1 Consolidated Connection Services (CCS), T2 Service Provider Architecture (SPA), T3 GÉANT Testbeds Service (GTS), and T4 Performance Monitoring and Verification (PMV).

GÉANT is currently evolving its service framework in parallel to relevant standards and therefore, this document only presents the initial principles of the framework. For example, the ongoing work between TM Forum and ETSI, to converge the next generation service delivery ecosystem developed by TM Forum with the requirements for NFV imposed by the ETSI standards. is reflected in a suite of specifications[1] constantly updated and evolving (e.g. on hybrid network management and information models for virtualisation) which need to be taken continuously into account by the work carried out in JRA2.

The document presents the principles for GÉANT services' evolution as well as the roadmaps of JRA2 network services development for Task 1 to 4.

[1] For example 'Hybrid Infrastructure Platform (HIP) Implementation and Deployment Blueprint', TMF070A, Release 17.0.1 , November 2017 where HIP is intended to be an example of how to use the various TM Forum artifacts (e.g., APIs and data models) to manage an infrastructure that consists of virtualized, physical and mixed resources or 'Connectivity Patterns for Virtualization Management', IG1147, Release 16.5.1, March 2017,

1 Introduction

The rapid growth in the user base, requirements, and reliance on our networks has also increased network complexity. Incremental enhancements and upgrades with additional hardware pieces or new services to address this evolution have resulted in complexity in service delivery. Taking also into account the recent evolutions on virtualisation an evolved architecture approach is needed for next generation services.

The service evolution includes identifying basic functions and services which then can be combined to create more sophisticated services. Also the service evolution includes functions and services integrating at the information, system and process level, to allow common cross-function processes, reduce duplication and complexity, and thus increase their efficiency. The approach is to identify basic building blocks of services and operations upon them.

This document presents the concepts of network services evolution in GÉANT in Section 2. Section 3 presents an overview roadmap of the associated developments conducted in JRA2 Tasks 1-4. Section 4 provides insight on relevant best practices and standards. Finally, Section 5 consolidates key points. Further details on the roadmaps for Task 1 to 4 are included as Appendices.

2 Network Services Evolution

Traditional network services evolve to offer more than just fixed transport or a single infrastructure that supports a single, fixed, forwarding protocol. Network service providers still perform basic transmission of data between points, but they also offer users the option to customise their network service environments, to define their networks to most effectively pursue their specific objectives. Of particular interest to network service providers is a reliable means of continuously introducing and supporting these innovations within their infrastructures such that they can function safely and properly without interfering with each other or existing production services. Composable services – services constructed of virtualised functional and/or infrastructure resources - offer a compelling approach to address these requirements.

Virtualisation, service composition and service function chaining are gaining traction within the advanced networking community as a promising means to address network evolution. Flexible construction of services to meet specific customer needs, and doing so in a deterministic manner is an increasing requirement of users. .

User facing services need to be treated as objects, decomposed to a set of more basic components or service objects. These objects can be modelled via recursive service and sub-service decomposition, down to the level of a basic set of atomic (core) services. Modularisation and composition allows for the creation of flexible new service offerings and new service instantiation is simplified. At the same time, decomposition modelling and resource orchestration provides the framework to validate and analyse the performance of these services..

Basic services and support functions can be orchestrated to produce more sophisticated service constructs that can be easily replicated and customised, reserved, activated, and then operationally monitored and verified throughout the lifecycle of the entire service.

2.1 Evolution Principles

Evolution of network services requires careful analysis of the principles of abstraction and virtualisation i.e. disaggregating the user facing service definition from the underlying physical infrastructure, technologies, or implementation that may be used to realise the services. This design methodology is technology agnostic – service instances are measured against what they were defined to do, not how that function is implemented, and so are not dependent upon specific products or technologies.

2.2 Virtualisation

Virtualisation is the abstraction of a service object to define what that service does, or how it behaves, independently from the underlying physical implementation of that service capability. It decouples the *service object* from specific physical infrastructure, platform, or technology upon which it may be modelled, or upon which it is realised. By virtualising a physical object, a virtual instance can be mapped to any underlying physical context that can deliver the defined behavioural characteristics.

In the context of emerging network virtualisation, there are virtualised infrastructure analogs such as virtual machines, virtual circuits, virtual switches and routers, virtual storage, etc. Virtualisation also extends to functional capabilities that may have traditionally been integrated as features in particular hardware, such as encapsulation, encryption, policing/shaping, filtering, etc. (see also the ETSI NFV specification, as discussed in Section 4.2.)

By delivering services in the form of virtualised service objects, the behaviour of those service objects can be defined independently of any particular physical platform or implementation technology. Service providers are then able to factor large traditionally monolithic and increasingly complex services into a set of basic common subservices, “atomic” services, that can then be used as building blocks to compose improved or customized services. These common atomic services are simpler to develop and validate, and can be assembled, or “composed”, into more sophisticated services in a manner that extends reliability, availability, and maintainability in predictable and deterministic fashion. Further, because these virtual service objects are decoupled from specific hardware, these services can be implemented using different technologies, or different vendors and different products.

Virtual objects can be grouped together and their data flow topology explicitly defined to create “composable services” or virtualised *composite* service objects. These composite objects can themselves be manipulated/orchestrated to create more sophisticated and customised user-facing services.

A rigorous virtualisation model efficiently bounds a service object’s behaviour, i.e. it defines and constrains the object’s ability to interact with other objects. This isolates the object to prevent it from interfering with other service objects, and simultaneously insulates the object from interferences from other objects. This isolation and insulation of individual service objects extends to composite objects, and thus to higher-level service environments. (This extension of atomic service objects’ behaviour, like insulation and isolation, to larger complex composite objects thru construction is a core tenet of “composable” services.) This does not prevent separate service instances or environments from communicating or interacting with one another, but requires that such interaction be explicitly defined on the part of both service objects.

Virtual service objects can, by definition, be hosted on any hardware context able to deliver the required functional attributes. Thus this virtualisation model can extend to objects not traditionally considered network components. For example, a radio telescope can be virtualised by defining a service object that behaves ... like a radio telescope. That virtual radio telescope (VRT) will define its attributes and data flow ports in the same manner as other virtual service objects (e.g. a VM or virtual circuit) and can be orchestrated using the same lifecycle primitives and topology descriptions as other virtual network objects, and can thereby be dynamically assembled into a large-scale distributed real time science facility.

To the degree that other radio telescopes and observatories have similar capabilities, the VRT instance could potentially be mapped to any one of those participating physical facilities as long as the requested user attributes for the VRT instance can be met at the other locations. Another example includes virtualising mobile cellular spectrum to allow new mobile service models or even software defined radio formats to be instantiated and piloted.

Virtualisation offers greater potential than simply more flexible network service construction. Its innate ability to define the objects that make up the service environment and ability to manipulate those objects using a common model enables us to construct highly sophisticated global service environments that are not presently possible with conventional network engineering and/or traditional services.

2.3 Service Building Blocks

For the design and engineering of evolved services, building blocks such as hierarchical service models, physical and logical resource elements, and control and management operations are required. These are discussed in turn in the following sections.

2.3.1 Service Models

Services should be modelled in terms of what they deliver and/or how they behave, regardless of the technology that may be used to realise them. This is service “abstraction”.

Abstraction allows the service objects or functions to be clearly understood (i.e. “bounded”) as to what they do – or what they do not do – before they are mapped to particular hardware technology or products. It also enables the services to be technology agnostic, which allows different network service providers to implement them with different vendors, or different technologies.

The services delivered by a provider employing this framework delivers the service functionality by constructing a machine that performs the desired function. For instance, GÉANT builds a machine called an “IP service” to move IP packets from one location to another. In this context, GÉANT produces virtual “objects” or resources that are instantiated in the GÉANT infrastructure and placed in service to the user. These objects may be simple common virtual service objects, such as a virtual circuit or a virtual machine, or they may be more sophisticated service objects that are constructed from many other service objects. An example of the latter might be a multi-point L3 VPN. We will use this object-based model to describe future services. Thus we begin to conceptualise “services” as functions performed by “objects” the provider allocates to, or on behalf of, the user.

Object-based models for virtualised services are still evolving across industry and research communities. JRA2 Task 3 GÉANT Testbed Service has developed a Generalised Virtualisation Model (GVM) to support the dynamic creation/orchestration of testbeds (or more generally “virtual network environments”) comprised of many different virtualised objects. The GVM implementation appears to be most advanced realization (working code) for wide area networking to date, but certainly it is not the only possible approach. GVM continues to evolve and still requires community input and thinking. In recent years, work relating to network function virtualisation (NFV) bears further investigation as NFV and GVM appear to be converging. The integrated service framework should take into consideration approaches used in different industry standards and initiatives, for example, modelling network functions in line with ETSI VNFs.

We begin identifying basic service objects/functions that are "hardware analogs", i.e. these service objects reflect common conventional infrastructure components. Basic user-facing *infrastructure* services can fall into one of the following categories of objects:

- **Transport service objects** move information, unmodified, from one geographical location to another.
- **Switching and forwarding objects** sort information among transport service objects. Ingress information is inspected and processing performed to decide how to direct the information on egress. (These could also be considered a subset of computational objects.)
- **Computational service objects** are general-purpose compute objects that translate information according to software function.
- **Storage objects** hold information, unmodified, for later retrieval.
- **I/O service objects** introduce/extract information into/from the system. For example, a scientific instrument or sensor, or a display.
- **Composite services** provide the ability to group the above objects into larger objects.

These basic objects (components) can form the base for other, more-specific functional objects. For example, a "rate limiter" service object could be implemented using a more general "container" computational object instantiated with software that effects the more specific "rate limiter" service object behaviour.

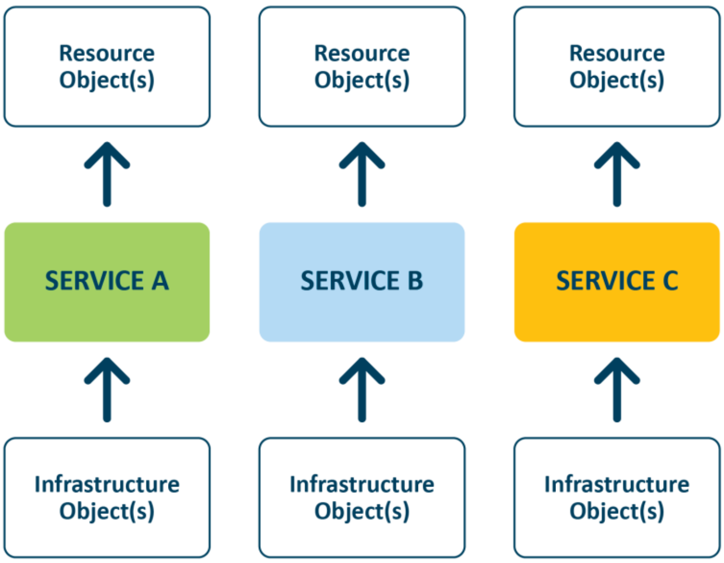

Figure 2.1: Modular approach to service construction

Several independent services are depicted in Figure 2.1 as a generic software process that takes as input a set of "infrastructure objects", and produce a user-facing output service object: a "resource object". This output resource is the object placed in service for the user. These input objects may be real physical infrastructure devices in an inventory pool managed by the service, or may be objects produced by other services. The “user” in this case is not assumed to be a GÉANT customer, but may be any [authorised] requesting agent, even another service. In this diagram, and in a generalised virtual services model, the term “infrastructure” and “resource” refers to the inputs and outputs, respectively, relative to a particular service and should not be construed to mean physical infrastructure or the ultimate resource delivered to the end user. This allows us to speak generically of virtualised services without regard to where a particular function might reside in a stack or service chain and without assumption on the nature of the input infrastructure object or the output resource object.

This dynamism is particularly relevant to virtualised services where a particular service (e.g. a VM service) may rely on other service(s) (e.g. a bare metal server (BMS) service) to provide the input “infrastructure” they depend upon, presented in Figure 2.2 The underlying BMS infrastructure service is generalised, and so can serve many other clients beyond the VM service. The generalised virtualisation model enables this dynamic interplay among services and allows the capital investment of base-level services to be applied more efficiently across a wider range of use cases, and reduces duplication of support functions required of the other services. This also introduces the ability of services to acquire/release infrastructure, as needed, to be more agile and efficient responding to capacity fluctuations.

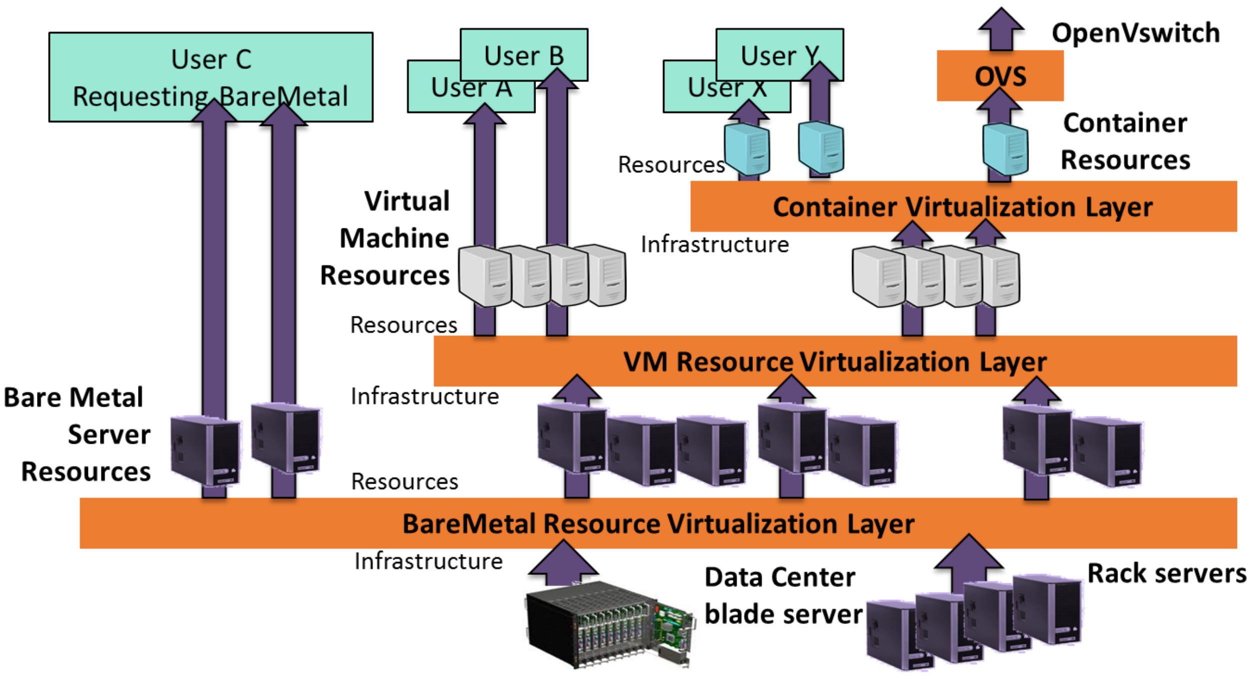

Figure 2.2: Layered resource virtualisation model

Figure 2.2 depicts how services use other services to dynamically acquire/release the [relative] infrastructure components required to produce their respective output service object. It also shows how such service factoring can define services that more efficiently cover many use cases.

Virtualised services offer great prospects for responsive dynamic/elastic interoperation and novel service concepts for both users and network service providers. Complexity, however, must be managed in a rigorous, scalable, and deterministic manner.

The model expressed here, where abstracted service objects are defined in a well-bounded manner, are composable within specific rules and are individually verifiable, allows complexity to be effectively managed. It also enables automated processes to provide services, intelligently analyse performance, and identify faulty components using deterministic, automated processes that extend from base-level atomic services to very complex envisioned service constructs.

2.3.2 Composite Services

Composite services produce service objects made up of other virtual service objects, assembled to address specific user needs. An implicit requirement of composite services is the ability to define the data plane relationship among the constituent service objects. The data plane arrangement of the constituent objects, combined with the functional behaviour of those objects, produces the emergent behaviour of the composite service object itself. A common control plane model is necessary to generalise the service interfaces to allow them to be assembled using common lifecycle primitives.

The ability to functionally and recursively decompose high-level, user-facing service requirements into a set of sub-services whose service objects that can be assembled – or composed – to produce the desired high-level functionality is the basis for composable services. The resulting service object composed of these constituent sub-service objects is a “composite” service object.

Within the existing JRA2 services, the GÉANT Testbed Service (GTS) has implemented a basic composable service architecture that enables GTS to construct wide-range composite structures from basic virtualised infrastructure services. These composite service objects are, in fact, the testbeds created by GTS.

An example of this modular approach to the creation of more complex services is depicted in Figure 2.3. One or more virtualised service objects can be combined using composability rules to assemble the multi-domain resource object – an Ethernet-framed connection object – which is provisioned and placed in service for the user.

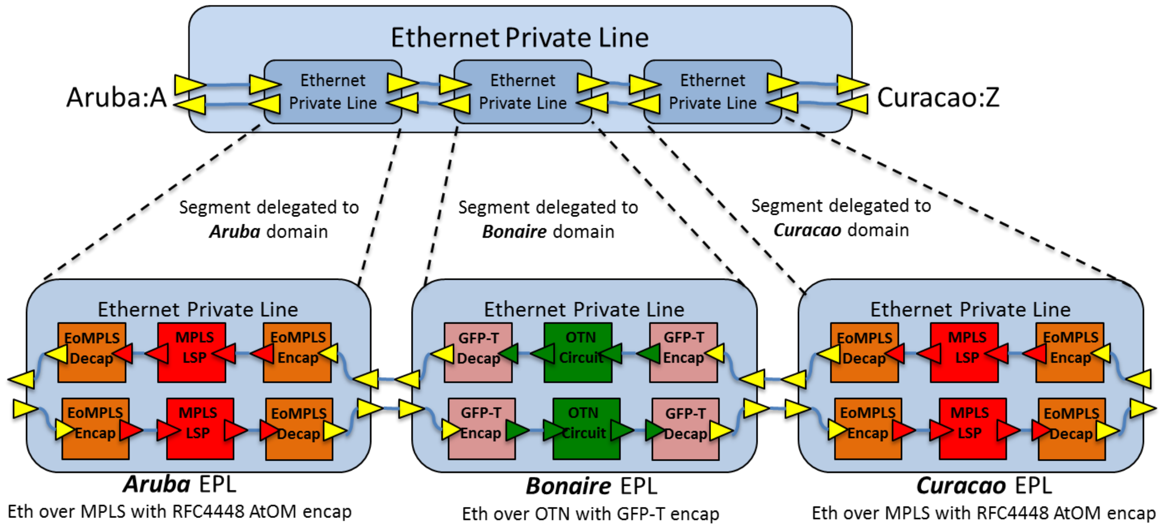

Figure 2.3: Service decomposition to atomic service objects: Recursive 'tree' segmentation of end-to-end EPL connection service

It should also be noted that a rigorous composability model is essential to address the Fault, Configuration, Accounting, Performance, and Security (FCAPS) requirements of real networks. Simply stated, FCAPS characteristics of the atomic service objects are defined such that those characteristics will extend and combine into the composite structure in the same manner as the service objects themselves are constructed to create the composite.

Another example of an integrated service model is when services request services. Several services may be incorporated into a single, abstracted computational platform – the latter decides which specific type of object to produce, based upon the user's specified requirements.

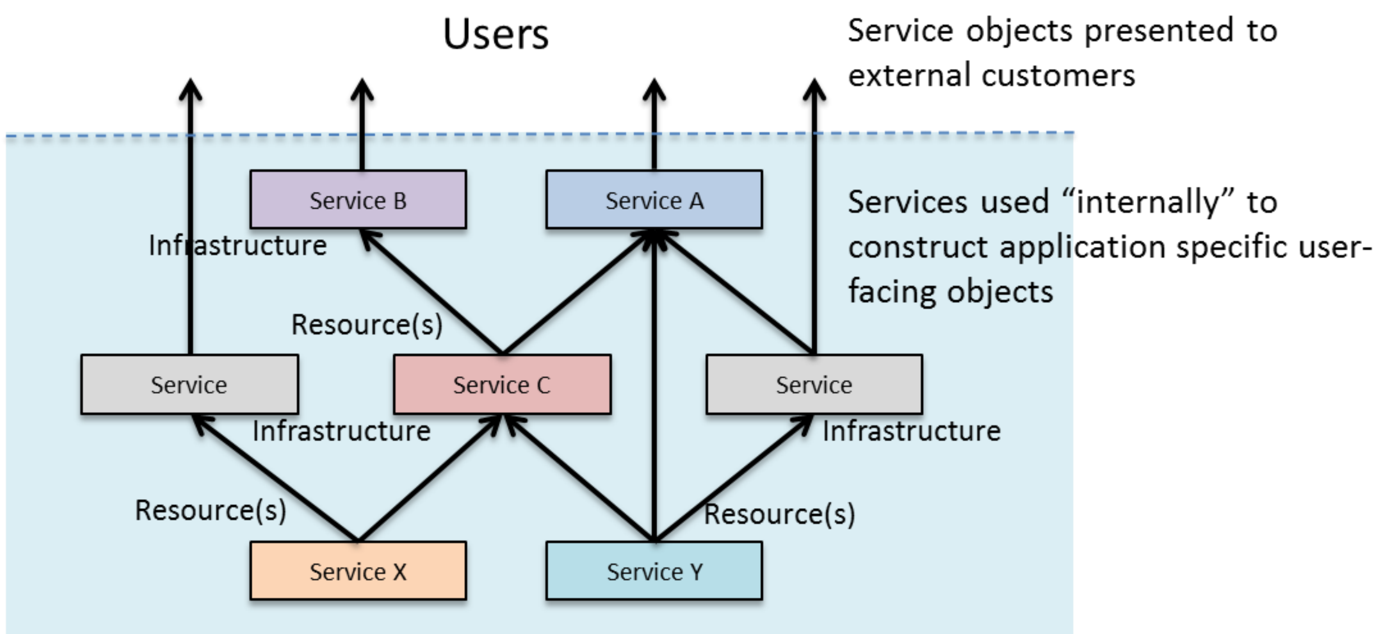

Figure 2.4: Composable services model

The general form of these 'services requesting services' implicitly means that the 'user' of a service is no longer an actual human or organisational entity, but is very likely to be an agent performing some other higher-layer service. As shown in Figure 2.4, the resource object created by the lower-layer service becomes the 'infrastructure' object(s) utilised by the higher-level service.

This standard model of a network service has interesting implications. Given that the common, semantic lifecycle primitives used to interact with these services describe the requesting agents' required resources, and that the resulting resource object is input to other service(s) as infrastructure, it follows that the form used to describe the requested resource can also be used to describe the infrastructure for these services. This means that there is no finite set of user-facing services – every service faces a 'user', and the 'user facing' service objects are also the 'infrastructure' objects that make up the inventory.

Composite construction, hierarchical construction, object-oriented construction, etc. are common paradigms in software engineering and modern programming languages. These concepts have been applied to many other areas, such as VLSI design, DSP applications, and even software development as a means to manage complexity.

The model described for creating composite services can be linked to such recursion in service modelling, as presented in [TMFSID], the ETSI NFV Management and Orchestration [MANO], IETF draft on Hierarchical Service Function Chaining [HSFC], and in the GTS Architecture [GTS].

2.3.3 Management Operations

A set of operations for participating services and resources are also required. Such operations can follow the lifecycle primitives as described in the Generalized Virtualization Model [GTS VIRTUALISATION] upon which the GTS is based, or they could follow the TMF Open API specifications such as those of the Activation and Configuration API [TMF Open], or Management and Orchestration operations as defined by the ETSI NFV specifications [ETSI NFV SPEC] Other models may also be implement such as a Slice Federated Architecture (SFA).

The operational requirements include:

- Allocation of the necessary infrastructure to support a resource instance of the type requested with the specified attributes.
- Activation of reserved resources in the infrastructure and their placement as a service to the user.
- Deactivation of a resource – taking it out of service while retaining the reserved infrastructure, as might be required for maintenance operations.
- Release of the resource instance and all allocated infrastructure back to the available pool.
- Querying a resource for the attributes associated with the resource instance.

Resources can then be managed through their lifecycle from initial inception until they are no longer of use.

2.4 Defining Network Objects and Network Functions

The JRA2 team has identified additional, functional attributes to map onto basic virtual objects to define more specialised virtual network objects. Applying specific attributes to an object and defining that flavour to be a new object can simplify the user's task of building the networks they need. For instance, an SDN controller could be created by the user allocating a "host" resource and then the user going about installing ONOS software [ONOS], and setting up the scripts to run that software. Alternatively, the provider could define a service object (say, an "ONOS-controller" object) that is constructed of a VM host with attributes that specify an ONOS boot image and user-specified attributes such as IP addresses of controlled SDN switch devices. The user sees an ONOS controller – not just a VM with some software.

Virtual Network Functions (VNFs) are also included in this context. Functions such as encapsulation/de-encapsulation, encryption/de-encryption, packet filters, rate limiters, etc., can be implemented in virtual containers and can be spun up very quickly and many (possibly hundreds) could be placed on a server, or load-balanced appropriately across (virtualised) infrastructure.

There is potential for future work by the R&E community and/or industry collaborators in the area of Network Function Virtualisation and Network Service Chaining (NFV/NSC) that can take advantage of GÉANT's advanced virtual infrastructure management via GTS. The notions of composability and construction already presented extend to such logical network service chaining.

There is substantial work in service virtualisation evolving within ETSI, IETF, TMF, and other industry specification groups. Significant And there continues to be a great deal of research work being undertaken in these same areas. While there are similarities among these groups in many respects, there is not yet unanimity on such standardisation. As example, the IETF has numerous RFCs describing Network Function Chaining, Service Function Chaining, composite objects, and hierarchical decomposition of services using these concepts to address complexity and varying administrative or responsibility roles. RFC 7665 Service Function Chaining Architecture [IETF RFC 7665] is recommended as a starting point for FFI. IRTF draft Network Virtualisation Research Challenges as of July 2017 [draft-irtf-nfvrg-gaps-network-virtualisation-06] identifies areas such as QoS, service composition, multi-domain, network function placement, privacy and security, network slicing, and device virtualisation among the topics needing further research.

JRA2 will continue to monitor the evolution of these industry specification groups with a view to bringing these GÉANT tools and services into conformance and interoperability with standards, as they emerge and as appropriate to meet the needs of the GEANT user community.

2.5 Service Construction

Construction of services in GÉANT is expected to happen through orchestration, as in [ETSI-NFV]. "Orchestration" in this context may cover a wide range of activities from assembling a few virtual resources into a testbed (ala GTS), constructing a service chain (e.g. to provide load balancer to a web service), to coordinating complex workflows. Orchestration becomes even more complex if services in one domain are likely to act as agents to assemble composite services spanning many domains.

The orchestrator can invoke multiple service primitives to build a complex service instance on behalf of any user, including other orchestrators. Such a recursive service model is present in NSI [OGF NSI], GVM, ETSI NFV, and TMF SID service models.

In cases when a service software agent must issue a service primitive to an agent that does not conform to a common model for service composition and orchestration, all that is required is the construction of wrappers that convert the common service/resource lifecycle primitives to appropriate command sequences in the old system. This is how GTS incorporates OpenStack and OpenNSA software into its service.

A service provider only knows about, and is only responsible for, those services it delivers. It is not responsible for other services, upstream or downstream, that it did not construct – indeed, it is questionable whether the local agent will know if such upstream or downstream services even exist. If we assert that a "parent" service is responsible for delivering a working service to its requesters, then the parent need only verify that its children service objects are functioning properly. If the root orchestrator agent acts accordingly to verify that its children all conform to service specifications, and each child does likewise recursively, then we can – by construction – assure that the entire service complex functions properly.

There is an important corollary to the above hierarchical decomposition theory: if a child service object is found to be faulty, then the parent should, by construction, also be considered to be faulty. A performance analysis of the parent performance would result in a performance analysis of each child. Similarly, it is the child service's responsibility to correct the fault.

As a result, there needs to be a protocol to a) ask that a child service instance be verified, and b) notify a child service that it has been found to be faulty, and c) notify a parent service that the fault has been corrected. The hierarchical and multi-domain construction of composable services means that the constituent sub-objects are assumed to be opaque, i.e. their internal structure is not visible (only external interfaces and expected performance attributes are known).

In order to verify such opaque service objects, there must be an architecture that places test points at the edge of each service object that allows that service object to participate in a deterministic performance verification process. The protocol required will manage the process recursively to identify failing components and to notify the appropriate agents of the results. The appropriate agents are not necessarily the "user" but more likely the responsible service provider with a suspect service object. This Performance Monitoring and Verification (PMV) processing is the subject of work currently being developed as part of JRA2 Task 4.

The architecture and protocol are essential to the FCAPS aspects of hierarchical decomposition and verification. Any authorised agent, for instance, a high-level orchestrator, or a OSS/BSS process, can leverage this capability to conduct automated fault analysis. It should be noted that such hierarchical authorised decomposition of the fault analysis process is critical to global scaling of PMV across administrative domains.

The end-to-end scenario is relative to each service object – not relative to some global absolute end point(s).

It should be noted that service selection is often done at different levels – each service object being the product of a particular agent that might do its own service decomposition and provisioning. It is often the case that the process of service selection and specification will take place on multiple levels, depending on the scoping of each domain agent.

One example of such an integrated service environment would be the establishment of an applications-specific network tailored to address the needs of a particular user community.

3 Network Services Development Roadmap

Network services evolution, as presented in this document, is already infusing the development of several tasks within JRA2. Many of the principles outlined here were results of work directly related to the GÉANT Testbeds Service, as it was recognised that the best way to build widely distributed and multi-purpose testbed environments and isolate them from one another, was through an object-oriented approach that incorporated techniques such as virtualisation, service abstraction, composite construction, rigorous performance management, etc.

Four tasks in JRA2 (Task 1 to 4) are related to the evolution principles already presented - Consolidated Connection Service, Service Provider Architecture, GÉANT Testbeds Service and Performance Monitoring and Verification. Further roadmap details are provided in Appendix A (Task 1), B (Task 2), C (Task 3) and D (Task 4).

3.1 Consolidated Connection Service (Task 1)

The Consolidated Connection Service (CCS) developed in JRA2, Task 1 provides an abstracted service model for connections that can harmonise and consolidate existing GÉANT connection services under a single coordinated service development effort and a single integrated, intelligent/automated service provisioning agent, thus creating a fundamental building block of more advanced network services.

The CCS approach is based upon the NSI abstraction of a "connection" to be: A logical conduit between two locations through which user data is transported, unmodified, from ingress to egress.

This simple technology-agnostic service abstraction allows for various service definitions (or object classes) to be defined, as needed, by the broader user community and implemented locally by each participating NSI service domain. Currently, only the Ethernet framed transport service is implemented, although this covers a large class of common use cases. The only requirements this service class asserts on the infrastructure is that Ethernet frames be transported, unmodified, from ingress to egress. Such a connection could be implemented over Ethernet infrastructure, or an MPLS infrastructure, or an OTN structure, or even over raw spectrum. The user perceives the defined behaviour of a logical conduit that moves their Ethernet frames from A to Z. This type of abstraction is essential for technology-agnostic virtualisation.

Further, the NSI service model poses a virtualised object-oriented semantic that deals with the service object's *lifecycle*. Thus a connection is instantiated when it is Reserve()'d and the necessary infrastructure (whatever that may be) is selected, and allocated/scheduled, as necessary. The service instance is Provision()'d (placed in-service to the user) according to the reserved schedule, and it can be Release()'d and Cancel()'d by user control as well. The reservation process for the requested service instance recursively decomposes the request into appropriate multi-domain path segments and recursively Reserve()'s those segments using NSI protocol [NSI]. This lifecycle model and the interface semantics disaggregate the user facing service *model* from the underlying per-domain implementation at both the hardware and software/information architecture layer.

The recursive decomposition of the connection request is typically referred to as path computation, path finding, or path aggregation (NSI). However, this path selection and reservation process is also a facet of orchestration. Orchestration is generally taken to mean a broader coordination of heterogeneous services (e.g. NFV, or coordination of network resources with other e-Infrastructures (such as work undertaken in JRA1 Task 3), but fundamentally this difference is subjective. The pathfinding function of NSI-based connection services does not conflict with other Orchestration functions or approaches. The end-to-end view of the NSI connection object is defined solely by the end points specified by the requesting agent and the requested performance attributes. If that requesting agent is a separate orchestration agent doing its own path planning and reservation, this is allowed within the NSI service model.

The point to note here is that resource selection and reservation may take place at multiple levels that may hide or expose complexity or that have a different "end-to-end" scope. Each service planning agent (orchestrator, PCE, etc) is able to be as detailed as is authorised or can delegate the more detailed aspects to other appropriate agents. The resulting composite service object will still meet the requested user specifications.

As a starting point, CCS service is piloted to deliver connection objects that GTS requires. GTS is a client of CCS. As CCS is an independent service, it can also serve other clients or purposes, and is currently preparing to incorporate the GÉANT BoD service capability and the OTN based Wave service. All of three of these services deliver the same abstract connection object: Ethernet framed connections with varying performance attributes. In addition, the CCS is now being integrated with SPA as part of the ELINE project.

CCS/ELINE v1 results should be ready for pilot testing early in 2018-Q1. A subsequent release (version 2) will provide additional needed functionality, and should be ready for pilot in late CY2018-Q4 [M8.1].

3.2 Service Provider Architecture (Task 2)

As explained in [M8.2] Service Provider Architecture (SPA) in JRA2 Task 2 "is taking a unified approach to the typical working of Communication Service Providers (CSPs). The Task 2 vision is to set up the groundwork and building blocks for the implementation of a holistic framework that will bring together all available services and user/business/operations supporting activities under a single umbrella. In this way, a consistent user (customer)-centric view [CCV] of the GÉANT network and its services can be provided in an automated, self-service manner to all relevant stakeholders."

Service Provider Architecture (SPA) in JRA2 Task 2 builds common underlying information systems and supports common OSS/BSS processes and is aligned with TM Forum Frameworx [TMFFx]. The initial services portfolio consists of abstracted service models that are technology agnostic in terms of their physical implementation.

SPA focuses for the GÉANT4 Phase 2 project period is on the fulfilment and assurance operations procedures, as defined in TM Forum Frameworx [TMFFx]. This constitutes about 30% of the component APIs that comprise the Frameworx architecture. Implementing the remaining APIs and porting existing services to interoperability and then ultimately into full integration will take several years and substantial and ongoing software support. The benefit of this investment will be a well-vetted information architecture serving GÉANT, and forming a comprehensive, unified underlying information base.

The ELINE Project is a "pathfinder" project that aims to empirically evaluate the Service Provider Architecture (SPA) and to more clearly understand the service implementation and transition process. ELINE employs an initial version of SPA, and will migrate an existing GÉANT service to the SPA. In Phase 1 of the ELINE pilot, the Consolidated Connection Service (CCS) will be covered by SPA as the test case service. ELINE will also integrate the service performance monitoring function (PMV) for CCS into the SPA environment. The CCS service class implemented will conform to the MEF Ethernet Private Line service specification [MEF ELINE].

In Phase 2 of the initial pilot, the ELINE business process as defined for the SPA based CCS will be evaluated with a small set of opt-in use cases. This will provide feedback and tuning of the ELINE service delivery process.

ELINE includes following aspects of SPA, which is in accordance with TMF Frameworx [TMFFx]:

- The Resource Inventory – to keep records of the infrastructure elements and the topology.
- The Service Inventory – with the information of customer connections (circuits).
- Monitoring of the established circuits.

As explained in [M8.1], together with more details about the ELINE project and the service, ELINE builds upon the existing NSI-compatible code base of the OpenNSA software package, and adapts the three initial SPA components mentioned above to leverage the existing OpenNSA/NSI code base [OpenNSA].

The ELINE project combines standards and best practices from MEF, TMF, OGF, IETF, IEEE, and ITU. OpenNSA and presents a model for future consolidation of GÉANT connection services.

3.3 GÉANT Testbeds Service (Task 3)

GÉANT Testbeds Service (GTS) in JRA2 Task 3 (and previous work undertaken in GN4-1) has defined and developed an integrated virtualised service capability based upon a Generalised Virtualisation Model (GVM). GVM was defined as part of the GTS design effort and drew upon prior work in FEDERICA, NOVI, GEYSERS, OFELIA, GOFF, ExoGENI, DRAGON, MANTICORE, and the NSI standards working group. GVM was positioned as a potential extension and enhancement to the NSI service abstraction model. The GTS development task is focused on developing a viable, scalable service model to support a broad range of network research and distributed applications/service development and piloting. GTS was not missioned with developing an SDN, or NFV, or virtualisation focused testbed, but rather to develop a flexible and reliable capability supporting a wide range of network research in the WAN. The GVM was a necessary step to this goal and has served as the guiding design principle for GTS to date. The incremental refinement of GVM, and the extensive experience gained in the development of GTS, has resulted in a virtualised service capability in GÉANT that now informs this integrated services vision described in this document.

In parallel, the ETSI NFV industry specification group virtualisation model implements many of the principles discussed in this document, including a number of atomic services that create virtual objects, such as point-to-point connections, computational elements, switching elements, etc. These are complemented by a “composite” service object that constructs larger more complex objects from these basic services, and is fully in accordance with the Integrated Services Framework.

GTS services already support an open API that is necessary for integrated service construction. This API (the lifecycle primitive set) allows other services (not just the GTS) to be part of the ecosystem and provide or acquire the GTS service objects.

The latest GTS version in production is 4.0, while the next version (5.0) is in planning, including service provider features such as policy engines, resource allocation management, migration and traffic grooming, as well as checkpoint restart. More details about the GTS current status and future plans is provided in the Milestone document [M8.3]

3.4 Performance Monitoring and Verification (Task 4)

The formal objective of Task 4 Network and Service Monitoring (including Performance Monitoring and Verification (PMV) defined in [M8.4] is "to develop a generalised, but comprehensive, network monitoring capability that will: measure the performance of GÉANT network services, provide real-time feedback to network operations personnel or users", as well as "determine whether the network services are performing to specification, and if not, initiate an automated analysis to localise the fault, and notify the appropriate agent(s) to take corrective action."

The PMV Task has defined an architecture for performing performance analysis of physical and/or virtual service objects by which such services can be monitored, verified, and problems localised across a global landscape of aggregated services.

The monitoring, verification and problem localisation processes are being integrated with the SPA, such that PMV can be applied generically to any atomic service and recursively, to composite services. Such monitoring and verification must be tailored to each service, for example, a virtual circuit would be monitored and performance verified quite differently than a virtual machine or a testbed or an encryption VNF. PMV is currently addressing Connection services as part of the ELINE Project. PMV is planned to provide monitoring and verification for other services as well – e.g. computational elements, switching elements, etc. focusing on monitoring of key network service performance parameters of a service in production, as defined in Y.1540 and Y.1541 specifications [ITU-1540], [ITU-1541], [MEF-10.3], as presented in the milestone document [M8.4].

4 Related Work/Supporting Standards

The key work presented here correlates with the work of three standards and/or best-practices oriented working groups: the Open Grid Forum for the Network Services Interface (NSI) standard and Network Markup Language (NML) inter-domain topology descriptions, the European Telecommunications Standards Institute (ETSI) [ETSI] in ETSI NFV work, and TM Forum [TMF] with TMF ZOOM and TMF Frameworx..

4.1 Open Grid Forum

The Open Grid Forum (OGF) has standardised the Network Services Interface (NSI v2) inter-domain connection provisioning protocol and service architecture [OGF]. NSI offers abstracted and technology agnostic connection services, and its service-orientated lifecycle primitives formed the basis of the virtualised services of GTS.

The OGF Network Markup Language (NML) specifications have been broadly accepted within the R&E networking community to describe network topology [NML]. NML provides both a global inter-domain, service-oriented topology consistent with the NSI service abstraction, as well as a standard means for describing intra-domain infrastructure topology.

The NSI protocol is now deployed in almost 30 R&E networks worldwide, including most Open eXchange Points. GÉANT has been a key participant in the NSI specification and standardisation process, and continues to co-chair the OGF NSI Working Group. Other GÉANT partners (in particular SURFnet, PSNC, NORDUnet, and I2Cat) have been active in both the NSI and the NML definition and standards process.

4.2 ETSI NFV

Perhaps the most relevant recent development around virtualised network services, and NFV in particular, are those offered by the ETSI NFV industry specifications group [ETSI NFV]. This group was established and began working on a reference architecture for Network Function Virtualisation in 2012 (shortly before GTS development was commissioned.)

ETSI NFV was established specifically to create a coherent approach for the industry development of emerging NFV technologies.

Although the early stages of GTS began at roughly the same time as ETSI NFV, GTS was missioned to deliver user-defined and controlled wide- area testbeds for network research. Despite distinct starting points, GTS and ETSI NFV have evolved in convergent directions over the last four years. The GTS architecture and the ETSI NFV architecture presently overlap in several key functional areas: virtual infrastructure management (VIM) components and some orchestration aspects (as shown in Figure 4.1), and there has also been convergence of the ETSI architecture to the current GÉANT Testbeds Service architecture with respect to virtualised environments.

Ongoing analysis and comparison is part of JRA2's future work. It is possible that GÉANT could offer test environments for the ETSI NFV efforts – their "hive" is a collection of modest distributed test stacks used to carry out interoperability testing. GTS experience and working code might also influence the emerging models, while becoming more interoperable with them.

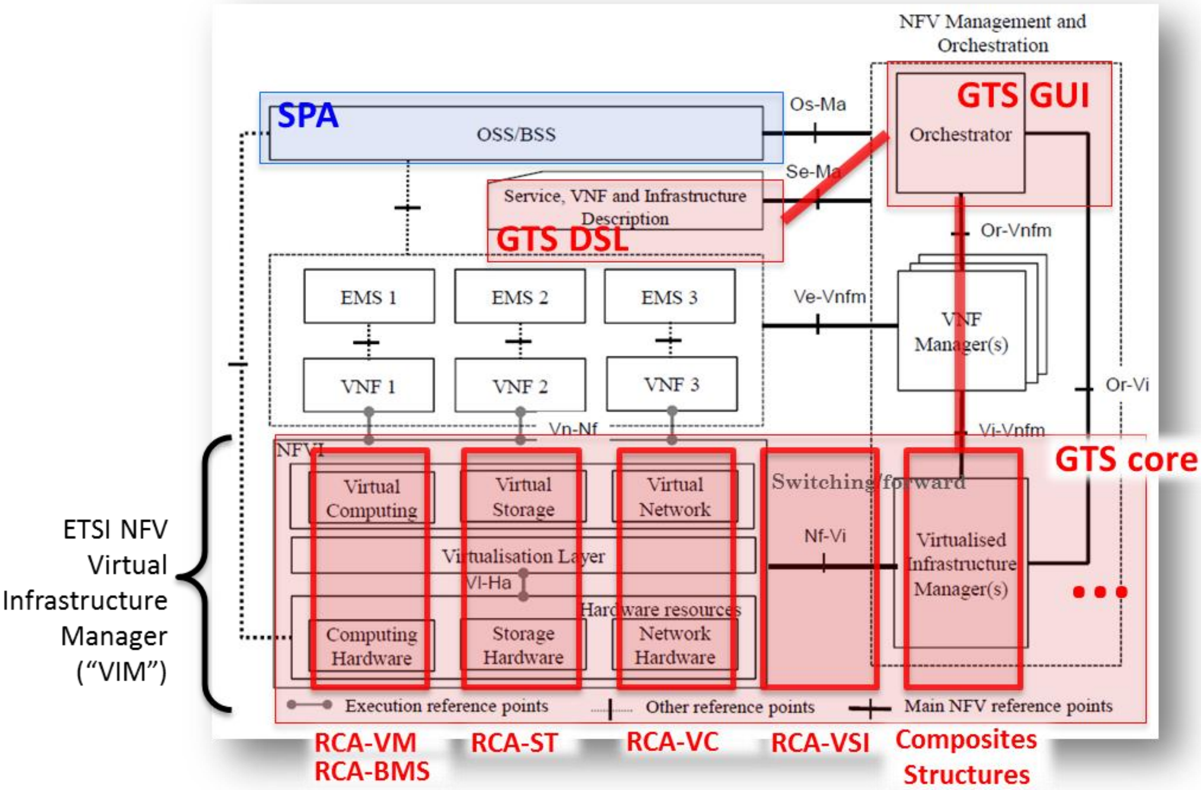

Figure 4.1: ETSI NFV Reference Architecture and JRA2 solutions: preliminary mapping

The second aspect where the two architectures, ETSI NFV and GTS, correlate is the OSS/BSS. This work, highlighted in the upper left-hand corner of Figure 4.1, is conducted within Task 2., This SPA work implements the Zoom architecture, as such, the OSS/BSS aspects depicted represent a much more substantial functional area than would be indicated by the single small box in the diagram.

4.3 TM Forum

Integrated Services Framework relates to TM Forum specification in multiple aspects, most important of which are TM Forum ZOOM architecture [TMFZOOM2], TM Forum Business Process Framework - eTOM [TMFFx] and TM Forum Information Framework - SID [TMFSID].

As per [TMFZOOM1] TM Forum Zero-touch Orchestration, Operations and Management (ZOOM) "is working to develop best practices to support both the technology and business transformation brought about by the introduction of Network Function Virtualisation (NFV) and Software Defined Networking (SDN)".

In its Business Process Framework, TM Forum provides systematic elaboration of business processes at several levels of complexity. This is already used as a starting point for the SPA and PMV work. While SPA focuses primarily on fulfilment and assurance, PMV work relates to the Service Quality Management process in its search for performance verification.

TM Forum Information Framework "provides standard definitions for all the information that flows through the enterprise and between service providers and their business partners" [TMFSID], thus helping the design, development and implementation of integration interfaces. Decomposition of individual services into atomic abstracted services, their integration with management methods and functions into the specific more or less complex customer-facing or internal services can use TMF SID as a potential source of information for easier composition and decomposition of services. TMF SID is used in the SPA work, as well as in the ELINE project.

The information architecture described by the TMF specifications offers GÉANT a well vetted model that can provide a foundation for GÉANT’s long-term information management strategy.

5 Conclusions

The potential impact of virtualisation and composability techniques on GÉANT's delivery of future network services is enormous. The integrated services framework described in this document leverages virtualisation and dynamic service construction techniques that will allow GÉANT to easily build complex, bespoke services that can meet world-class reliability, availability, and maintainability objectives. JRA2's evolution of these services has evolved from the Generalised Virtualisation Model, implemented in the GEANT Testbeds Service, and the first-hand experience of developing, deploying, and running the GTS service.

Service abstraction, virtualisation, and interoperability among services can deliver a number of advantages to system architects, managers, operators and users as it provides a framework for engineering and operating basic atomic services and subsequently for composition and orchestration of more bespoke and sophisticated services. This allows services to be tailored to specific needs of each user, and reduces duplication and complexity and improves reliability and manageability of the resulting services. In addition, and perhaps most importantly, a clear understanding of the internal logical structure of a service and its constituent elements is achieved. This enables identifying flaws in the service logic, or faults in the service instances themselves – and such rigorous fault resolution leads to greater reliability and availability.

The consolidation work is related to several standardisation and best-practices effort, including TM Forum ZOOM and TMF Frameworx, the ETSI NFV architecture, Open Grid Forum, and MEF Service Operations Specification. Due to the evolving standards and the constantly changing landscape, the work presented here is work in progress at the moment (see also the roadmaps found in Appendix A, B, C and D). Standards development has a direct impact on the work of tasks in JRA2 Network Services Development and JRA1 Network Evolution, as well as a broad impact on operational and service management activities.

Appendix A Task 1 Roadmap for Consolidation of Point-to-Point and Point-to-Multipoint Services

JRA2 Task 1, the Consolidated Connection Services, is missioned to consolidate the many-connections oriented service currently offered by GÉANT into a single, unified service offering. The initial services included are GÉANT Plus, Bandwidth-on-Demand, GÉANT Testbeds Service virtual circuits, and MD-VPN. Other similar connection-oriented services such as GÉANT Lambdas and GÉANT Alien Waves are planned to be incorporated into this unified model as the service progresses. The objective is to not only make access to these services more of a one-stop shop, but to collapse multiple product service concepts and support silos into a single, comprehensive service planning and development function that presents a simplified “connection service” model to the user. The strategy is to abstract these services in order to identify the common user requirements, and allow an intelligent provisioning system and API to map those service attributes to the appropriate underlying technologies and infrastructure elements.

Fully automated delivery of these services is a key objective. This effort includes: GUI development, hardware configuration protocols, NSI protocol, time-based path analysis and reservation, network topology specification/distribution, path finding, federated AAI, policy support, multi-domain interoperability, etc.

A.1 Work Plan

Task 1 of JRA2 is a mostly infrastructure oriented task. It will cover re-evaluation of all connection services currently offered by the GÉANT network (there are services which are very similar). Special attention will be paid to service implementation in the current GÉANT network infrastructure, the delivery and administration process, and the required level of automation. Results of such effort should be narrowed and a simpler service portfolio maintained (i.e. BoD, GÉANTPlus and GTS circuits are the same thing with different provisioning process. Both can be presented as one common service definition of connection oriented service).

An important part of Task 1 responsibility will be provisioning of service development laboratory in Prague, in cooperation with other JRA2 tasks. The laboratory is now used for the development of GÉANT Testbed Service (the Software development as well as hardware engineering, integration, and configuration interface testing.

Except the laboratory support, which will be continuing effort, the planned Task 1 roadmap will consist of the following:

A.1.1 CCS v1.0

CCS version 1.0 will focus on consolidation of point-to-point, Ethernet-framed virtual circuit provisioning. This will establish a Network Service Interface (NSI) standard circuit provisioning capability, using open source OpenNSA software as the code base. These will be best-effort circuits only, with no performance guarantees. (This is the same as existing EoMPLS circuits of BoD and GÉANTPlus). This initial CCS will assimilate the GÉANT Testbeds Service connection provisioning, the BoD service, and will provide an open and publically accessible NSI interface that other user agents can use to create “connections” across the GÉANT core.

As part of CCS v1 transition to the GÉANT core routers, JunosSpace was utilised to mediate provisioning requests between OpenNSA and the routers. (JunoSpace is a proprietary configuration management tool from Juniper.) This new southbound interface has proven to be unreliable for rapid provisioning requirements of GTS. Resolution of intermittent timeout issues and provisioning failures of JunosSpace has required more time and effort than was initially anticipated. This beta testing included GTS as the primary user application. Further beta testing has continued to resolve the JunosSpace interface reliability issues. Operations and JRA2 jointly agreed to halt any further debugging of the JunosSpace SBI. OpenNSA will be reverting back to the prior (simpler and more reliable) direct SSH interface used in GTS v3. This change should occur in early 2018. CCS v1 will also begin multi-domain peering with other NSI networks, starting with NetherLight and CESnet. Chain provisioning to direct neighbours is currently enabled in OpenNSA, NML topology processing enhancements will be added to support a more extensive “tree” style path reservations following the NSI recommendations. Other NSI domains, particularly those at Open eXchange Points, will quickly follow. This requires enabling TLS authentication between NSI agents in peering networks, which will have an impact on GTS and other client agents within the GÉANT ecosystem that do not currently utilise TLS certificates. A minor update to deal with authentication is being developed for GTS. As soon as this update it ready, multi-domain peering can begin in earnest.

As of December 2017, CCS v1.0 has been deployed at eight locations in the GÉANT core to support GTSv4.0. This deployment has fully implemented GTS Virtual Circuits as best-effort Ethernet over MPLS service, with a maximum capacity of 10GE. Additional PoPs will be supported as the BoD service termination points are migrated to the CCS.

CCS v1 is planned to be in official Pilot by 1 March 2018 (updated timeline as of December 2018). This Pilot is planned to continue until CCS v2 is ready.

A.1.2 CCS v2.0

CCS v2 is now planned to deliver the QoS performance guaranteed connections, an NSI-enabled circuit-provisioning GUI, and transition of GÉANT Plus circuits to CCS management. The QoS feature is the critical path for CCS version 2.

Quality of Service for Ethernet-framed connection services – or more broadly “performance guaranteed” connections – are a critical feature to be offered by CCS. Without per-circuit performance management, a service like GTS cannot offer researchers or developers a predictable level of service – making performance analysis impossible and many applications untenable. Further, a service like GTS without rigorous capacity management can easily overwhelm the infrastructure, negatively impacting all the GTS service environments, as well as posing a risk to other services. QoS and performance guarantees are essential for GÉANT and emerging virtualised services.

The MPLS core is currently unable to provide QoS guarantees on MPLS LSPs. As a result, it was decided the needed QoS/performance guarantees would be delivered over the Infinera-switched OTN layer instead of the MPLS core. This has necessitated the development of a new southbound interface in OpenNSA to manage the switched Infinera kit, and requires additional OTN hardware to be purchased and deployed to support the ~100 interfaces that must support PG circuits. (Update: as of Q4 2017, the OTN path planning and timeslot allocation code is complete in OpenNSA, and a southbound interface to Ciena OTN kit is working. The SBI for Infinera is expected in early Q1 2018 in the CCS software, and Q3 2018 for the additional hardware deployment.)

In parallel with the consolidation of the provisioning aspects of these Ethernet-framed, point-to-point circuits, Task 1 will develop or adapt an interactive user/operations interface to establish and monitor standalone circuits. This is to mainly provide an operations tool for the visual monitoring of the status of service instances, and secondarily, for direct manual provisioning by operations or conventional users. Upon making this GUI available, CCS, in conjunction with Operations, will begin transitioning GÉANT Plus service instances to full CCS support. (Update: as of Q4 2017, the MEICAN graphical provisioning tool, an open source, NSI-enabled, tool developed by the Brazilian NREN RNP, is being evaluated for this role.

At this point, all Ethernet framed point to point connection services previously covered by BoD, GEANT Plus, GTS, and GEANT Lambdas will be provided by CCS. The CCS software can support GEANT Open as well. CCS will provide a public, secure, NSI API for other clients to use to request or monitor circuits. Thus, a single service and a single product development team will be responsible for the evolution of "Connection Services", and the features of these services can be managed and provisioned more effectively and efficiently.

A.1.3 CCS v3.0

CCS v3.0 will incorporate enhanced-path selection capability, additional multi-domain peering, and a simplified installation, configuration, and operations.

The enhanced path computation is needed to support multi-point services and to provide improved multi-domain scheduling, topology processing, and switching point placement for multi-point multi-domain connection services. Consider, for instance, the process required to incrementally grow a multi-point L2 broadcast domain, or L3 forwarding domain (VRF) across domains in Europe and North America. Inefficient layout or periodic re-balancing can result in highly inefficient trees.

The simplified CCS v2 is intended to ease the initial deployment of Connection Services so that NRENs, campuses, or even labs, can initially deploy CCS services using only commodity hardware (e.g. some available servers running OVS or similar software switch), that can install and configure the service quickly and easily to minimise the impact to local engineering or operations staff (the goal is less than one hour per server), and to make the operational support of the service easily integrated into existing environments.

As a facility gains experience, the configuration process yields more control of the configuration, and as performance requirements eventually increase, the servers can be upgraded to hardware switching elements. The goal is to make virtual circuit services as simple as possible to install, configure and get running to enable services to expand into the last kilometre.

CCS v3.0 is currently estimated to be available in early Q4 2018.

In June 2017, the joint effort with Task 2 has been initiated in order to design and develop the ELINE service (internally referred to as "The ELINE project"). ELINE is a prototype integration of the Consolidated Connection Services (CCS) capability, using the SPA information architecture and APIs.

The ELINE project, while using a basic CCS v1 service platform, is part of the JRA2 T2 Service Provider Architecture development effort. This first ELINE port of CCS to SPA is a high-level, wrapper-based approach sufficient to evaluate a real service within the SPA context. If this ELINE pathfinder project proves the SPA to be a viable and strategically valuable approach, a more substantive adaptation of CCS to utilise SPA information base directly will be required. The scope of effort and schedule for such a re-implementation will be addressed as part of the SPA pilot report.

Task 1 contributes to the OpenNSA open source code base. The OpenNSA has a modular, southbound interface architecture, which makes it easily adaptable to many possible network infrastructures. The NSI Connection Service specifications are general enough to be possible for other institutions/networks (member NRENs, other research networks around the world) to adopt it, and using the standardised NSI lifecycle API, allow automated provisioning of multi-domain connection instances. Task 1 can support this deployment in member NRENs by developing OpenNSA backends, as needed for network equipment used by interested NRENs.

The first QoS aware CCS implementation has been introduced and demonstrated at SC17. The implementation works with CIENA 6500 platform (OTN switch) and allows a flexible number of 1Gbps timeslots to be mapped to a connection, as needed to guarantee capacity and jitter bounds. It was demonstrated together with GTS deployment over CIENA research infrastructure in Ottawa, Canada (called the Ciena Testbeds Service). The southbound interface to cover Infinera OTN transport is work in progress, and will result in the consolidation of the GÉANT Waves service.

Another area to study in T1 is the GÉANT Lambda service [GÉANT LAMBDA]. As this is a connection service (and in this case also just Ethernet service over "big pipe"), it should be provisioned using the same tools and procedures as other circuit service. In order to achieve this, T1 will examine the potential to provision such circuits using similar service definitions, as mentioned above. This entire circuit services portfolio should be then presented to users using one common web portal.

Multipoint services such as L3VPN are by their nature, also connection services [L3-VPN]. There are two models generally considered for abstracting multipoint services: The first is to treat point-to-point (P2P) connections as a trivial case of a multipoint connection, i.e. a multipoint connection with only two service termination points. The second model treats multipoint connections as a composite construct built up from one or more "switching" objects, each connected by point-to-point connections to the user designated end points and P2P connections interconnecting intermediate switching points for large spanning trees. This latter model leverages work in optimised spanning tree construction to grow larger, multipoint connections efficiently and incrementally.

The CCS development team expects this composite model to be more flexible and dynamic, particularly in multi-domain construction and in future NFV environments. CCS plans to implement multi-point using this composite construction method. The Task 1 team will investigate and propose possible generalised service definitions to cover multi-point services in order to allow their provisioning using NSI protocol and common software elements, such as path computation, virtualised switching and control functions, etc.

Appendix B Task 2 Roadmap for Service Provider Architecture

The Service Provider Architecture aims to provide an initial OSS/BSS pilot implementation that sits on top of the virtualised services and resources as represented with ISF and referenced by NFV ETSI. It is composed of loosely coupled microservice-based components that are representing different functional components based on the TMF TAM decomposition. All components expose TMF-compliant REST APIs that use a SID data model. SPA focuses on the business/high-level operations oriented orchestration of the components based on the end-to-end eTOM based business processes.

The initial SPA v1.0(beta) pilot is implementing the Consolidated Connection Services (CCS) point-to-point Ethernet private line service within the SPA information architecture. This CCS implementation within SPA is designated the ELINE" project. The ELINE effort will implement the CCS-based P2P connections using the SPA northbound API. This pilot is scheduled to begin January 2018, and run for 120 days. This pilot has two objectives: a) to determine the actual scope of effort to port a service to SPA, and b) to test the process of business process specification and use.

The features available in the initial pilot focus on migration of the CCS service to function under the control of SPA components and offer service fulfilment to a set of alpha users that will provide their feedback when testing functionalities related to automated creation and termination of E-Line-based circuits using a self-service portal. Based on the information provided from the dynamic service catalogue on the portal, the user chooses a service and activates the related business processes. The system captures user orders and tracks their status, manages the service instance data in the service inventory and uses the logical resource inventory. For the purposes of interfacing with the T1 CCS E-Line implementation, a TMF-compliant Activation and configuration API is used.

The pilot will run until end of April 2018, with the aim to capture and document the efforts needed to migrate the E-Line service to SPA and the user experience of using the automated system. Any issues identified in terms of design, implementation or performance and scalability will be addressed before continuing with the extension of the solution. It is also presumed that the ELINE pathfinder will expose hidden complexities.). The Pilot report will inform the future planning for SPA implementation. If the decision is to continue the implementation, we can better plan the buy vs build analysis, and better estimate the manpower and timeline that will be necessary.

SPA v1.1 development will commence immediately following the conclusion of the 1.0 pilot period (April 2018). Key features of 1.1 include a) integration of the PVM module to perform verification testing and monitoring of ELINE service instances, and b) will implement the SPA “feasibility” function. The PVM module (developed by JRA2 T4 PVM) will verify the circuit is established correctly and functioning as expected after provisioning, but before placing the connection into service for the user. This module will continue to monitor the service object and will trigger alerts and/or notifications when the monitoring functions determine the service object has failed. “Feasibility” translates to path selection and reservation for the CCS point to point connection services. This function is currently performed by the OpenNSA agent of CCS as the NSI “reserve()” primitive. This function will perform an end-to-end reservation of needed path segments across multiple domains, or main be restricted to only path selection within the local domain.

This work in version SPA 1.1 will provide further in-depth experience addressing service migration to SPA. We expect to see v1.1 in beta by Q3 of 2018, if there are no substantial challenges with feasibility implementation.

As we are near the close of SGA2 and prepare for future work, the SPA will be ready for incorporation of a second service that will enable integration tests, focusing on the correctness of the service agnostic implementation in terms of component interaction and data model. The services to be ported, their sequence, and anticipated timeline will be part of the SPA 2.0 Strategy phase.

The remaining work in GN4-2 should focus on scalability of the solution, addressing procedures for micro services management and intra/inter security of the distributed system. Also, the main new feature to be added is a policy engine that will provide means to implement different levels of rights of access. Another important aspect of the initial GN4-2 efforts for SPA is the training of the service management and operations teams so that the process of migration of existing services to SPA can continue. Based on the user experience with SPA v1.x, the addition of other features will be prioritised. These aspects will be resolved by the end of GN4-2, as we strategise SPA v2.0 in Q3 of 2018.

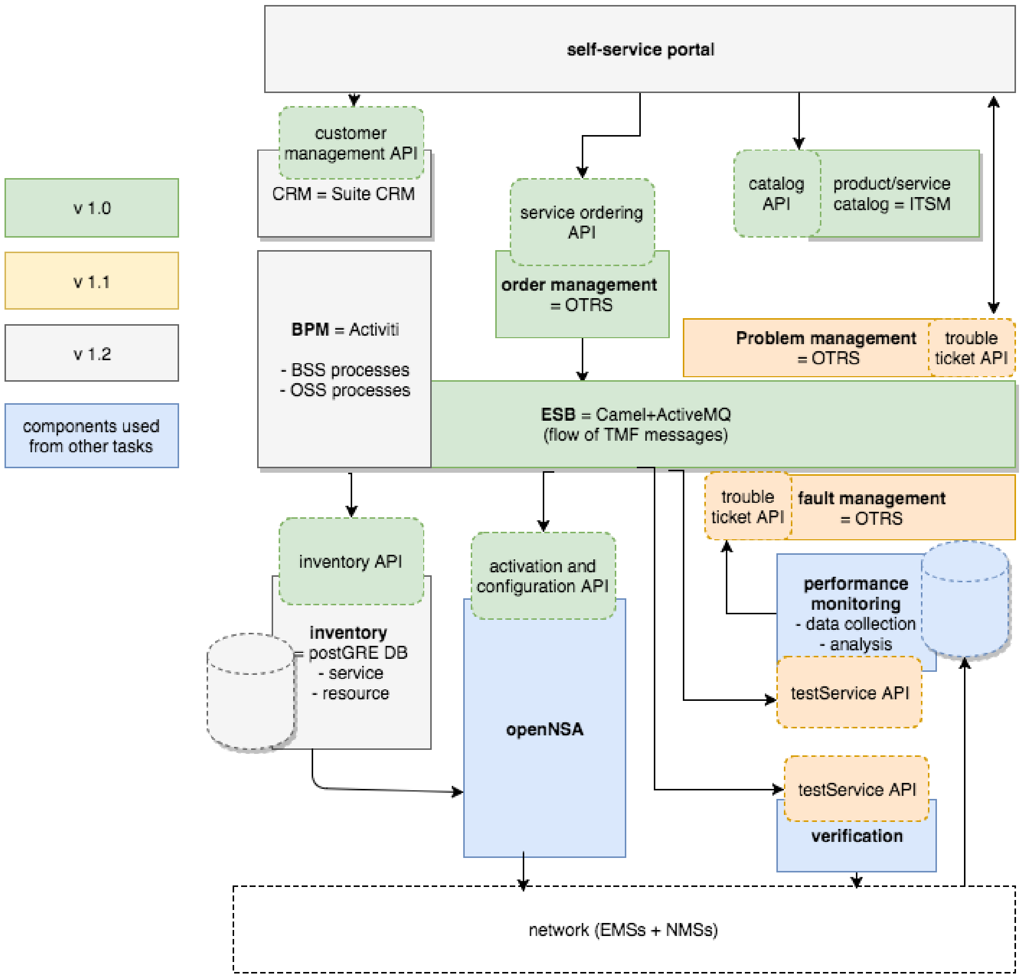

Figure B.1: SPA component view

Appendix C Task 3 Roadmap for GÉANT Testbeds Service (GTS)

GTS began design and development in GN3plus (April 2013) and made its debut in Q3 of 2014 with Version 1 launching in Bratislava, Ljubljana, Amsterdam and London. In 2015, Version 2 was deployed with three expansion sites added in Hamburg, Prague and Milan. In 2016, as GN4-1 was concluded, Version 3 was deployed and two additional sites were added in Madrid and Paris. In July 2017, GTS Version 4 was opened to users in 8 locations (Ljubljana was decommissioned due to backbone limitations). GTSv4 represents a major evolutionary step for GTS, migrating the service from its initial demonstration infrastructure to the GÉANT core MPLS infrastructure, upgrades in interface capacity to 10Gbps, introduction of advanced and fully virtualised SDN switching resources, and delivering bare metal servers as a major new user resource. As of Q4 2017, the GTS development team is completing the operational integration of the new servers, addressing reliability and performance issues associate with the circuit migration to the core routers, and preparing a number of remaining features in upcoming minor releases of Version 4.

After its currently deployed version (GTSv4.1) there will be a follow-up version **GTSv4.2**. Its main features will be a) QoS/performance guarantees on transport circuits, b) an improved and more scalable Internet Access Gateway (IAGW) functionality for Bare Metal Servers (BMS), and c) Multi-Domain (MD) environments. The IAGW for each project allows users to access the testbed infrastructures via a public IP address and VPN. Perhaps the most critical feature of all of GTSv4 is the ability to instantiate GTS "Links" that have real and enforced pseudowire-like performance guarantees. With GTSv4, "link" resources are established by requesting virtual circuits from the Consolidated Connection Service (JRA2-T1 CCS).

As of Q4 2017, CCS is pursuing QoS using OTN transport technology instead of MPLS hardware and it is now expected to be available in Q2 or Q3 2018. The positive control over capacity allocation provided by the QoS becomes a critical component of GTS version 4, now that GTS has an aggregate capacity of some 900+Gbps available directly to user testbeds. Other minor features will also be introduced with v4.2 and 4.3, such as simplified DSL specifications, enhanced, user-based project administration privileges, and various bug fixes.

The central theme for **GTS v5.0** will be a focus on features needed to promote GTS as a "sustainable" service of GÉANT. Two important features along this trajectory are 1) Checkpoint/Restart Migration/Grooming, and 2) Policy Engine.

Checkpoint/restart (c/r) allows a user to take a snapshot of their testbed's state, save that state to persistent storage, then at some later time remap the virtual environment to another equivalent set of infrastructure, and restart the testbed from the saved snapshot. This feature requires a notification protocol be developed to inform the testbed application of an impending checkpoint process (in order to allow the application to enter a quiescent state), and high performance distributed storage facility integrated across the GTS core in order to actually capture the state and to redistribute the state files to their new locations in preparation for restart. The GTS v5 Checkpoint/Restart feature is expected to require approximately 30–50 TBytes of high speed storage in each GTS location (about 250-400 TB in total.)

A related feature to checkpoint/restart is "active modification" (AM). AM allows the structure of an active testbed to be modified without requiring a complete release and new reservation of the entire testbed. In order to simplify a highly complex modification process, Active Modification will add or delete only a whole resource instance to/from a composite resource object. (GTS testbeds are constructed as hierarchical assemblies of composite objects, so “testbeds” are, in fact, just the root node in a tree of composite resource objects.) In order to perform these changes to a running distributed environment (i.e. a Testbed) the testbed itself must be notified of an impending change and allowed sufficient time to halt or idle the processes on affected components. It is this notification process that is also required of checkpoint restart.

From the service provider's perspective, checkpoint/restart enables resource migration and grooming. With migration/grooming (m/g) the GTS service provider can save all or part of a running testbed and re-map the testbed in order to vacate some set of infrastructure components, or to gather a set of sparsely distributed resources onto a single infrastructure component. These actions may be necessary in order to perform maintenance on hardware, or to free up partially utilised infrastructure for other purposes, or to perform green energy-saving power-down processes etc.

Testbeds – or more realistically: virtual networked environments – typically require substantial setup, configuration, and tuning over time. This makes users reluctant to release testbed resources when the testbed is not actively being used because the time and effort needed to recreate the runtime environment is substantial. The c/r feature is crucial to support user release of resources since it intends to make restart an easy process. This allows the user to apply their quota allotment more efficiently over the life of their project and allows the provider’s hardware investment to efficiently serve more users before capacity upgrades become necessary.

GTSv5 will also be delivering a policy engine feature that will enable service providers to more effectively define the rules and limits of resource allocation across their user base. The policy engine will define actors, roles, functions, and privileges associate with authorisation policy, and will define policy check functions, and relevant data object specifications on which the policy checks will be based (e.g. rollup of accumulated resource-hours.)

A further feature of GTS v5 will be a graphical drag and drop testbed editor. The initial plan is to adapt the jFED tool developed within the Fed4FIRE project to utilise the GVM API. Such a graphical editor must be able to edit complex hierarchical composite resource objects, generate the corresponding DSL, and submit that DSl file for Reservation and Activation. The graphical layout meta-data must be stored with the DSL description in order to maintain graphical rendering of the testbed from one editing session to another. Further, implementation of Active Modification – partial manipulation of a testbed service tree, graphical navigation through the testbed topology structure, real-time monitoring, rollback features, etc. result in a complex development and implementation of the graphical editing capability.

Appendix D Task 4 Roadmap for Performance Verification and Monitoring (PVM)

JRA2 T4 aims to develop a generalised but comprehensive network monitoring capability that will:

- Measure the performance of GÉANT network services.
- Provide real-time feedback to network operations personnel or users.
- Determine whether those services are performing to spec, and if not, initiate an automated analysis to localise the fault and notify the appropriate agent to take corrective action.

This network monitoring and performance verification capability must function within the modern virtualised network transport service context and support emerging non-traditional service objects, such as virtualised network service functions, and do so in a multi-domain environment.

JRA2 T4 will focus the design of network monitoring and performance verification capability to support the following types of network services: point to point Ethernet framed connections based on NSI common service definitions (e.g. CCS), Ethernet-over-MPLS, and MPLS based VPNs (multipoint L3 and L2 VPLS and MDVPN), chained service functions, non-standard SDN switched circuits and networks. It will focus on monitoring key network service performance parameters of a service in production as defined in Y.1540 and Y.1541 specifications. In the reminder of the text we will call this the network service performance verification and monitoring capability as PVM tool set.

The PVM approach will adapt and use both active monitoring methods and the approach with capturing the traffic and meta data (e.g. timestamps, payload signature, etc.) on monitoring zone border points, and near real-time performance analysis from the captured flow data. The amount of captured traffic will be balanced in such a way that system scalability, measurement accuracy, and security/privacy are achieved.

Key PVM feature sets and release roadmap and timeline consist of the following:

- PVM v1.0 – Design and development M1- M22 (February 2018). PVM v1.0 includes the following features: Performance verification and fault localisation of basic network services (L2 and L3 point-to-point and multipoint VPNs), per-service instance monitoring and SLA verification, GUI/dashboards for monitoring parameters, integration with the service inventory and service provisioning OSS components. Key milestones:
    - Milestone 4.1 – Advanced Service Monitoring/Performance Verification Architecture (ASM/PVA) – Month 9 (January 2017) - completed

    - Milestone 4.2 – Prototype for the measurement of large scale network infrastructure elements – Month 12 (April 2017) - completed
    - PVM Prototype demonstration – Month 13 (May 2017) - completed
- PVM v1.0 – Pilot Transition M22-M26 (June 2018) PVM 1.0 will be deployed in a limited set of locations – specifically those associated with the GTS service footprint. This is due to the availability of hardware consistent with the measurement point placement described in the PVM architecture.
- PVM 1.1 includes the rest of the features that are defined by the description of work: Integration with other OSS/BSS components, the implementation of 100G capturing hardware for high-speed (100G) link monitoring and monitoring chained services. This version will include a module to be integrated with SPA to address ELINE service verification and initial automated fault analysis/notification.
- PVM v1.1 – Design and development will begin M21 (Jan 2018) through M29 (Sep 2018),–
- PVM v1.1 – Pilot phase M30-M32 (December 2018). PVM v1.1 is not expected to meet production capabilities until GN4-3
- PVM v2.0 will go through a strategic review in 2018-Q3 to identify specific features, and is expected to enter development by the end of SGA2. An important feature will be enhanced verification capability for the CCS/ELINE services. CCS/ELINE will be offering Ethernet framed point-to-point and multi-point services provisioned with both performance guarantees and best effort, across both MPLS and OTN infrastructure, in both a GÉANT-only edge-to-edge mode and in a multi-domain end-to-end mode, and up to 100Gbps/flow. Support for 100G performance verification, monitoring, and fault analysis across all GÉANT CCS endpoints and legacy IP ports will be part of the strategic requirements of version 2.

References

[BOSSF] **Jithesh Sathyan, Fundamentals of EMS, NMS, and OSS/BSS, CRC Press, 2010**
Jithesh Sathyahttps://www.crcpress.com/Fundamentals-of-EMS-NMS-and-OSSBSS/Sathyan/p/book/9781420085730n, Fundamentals of EMS, NMS, and OSS/BSS, CRC Press, 2010

[CAT] **TMForum Catalyst Program**
https://www.tmforum.org/collaboration/catalyst-program/catalyst-program-benefits/

[CCV] **White Paper OSS Systems Need to Evolve From Network Management to Customer Centric Service Management, Cisco, 2012**
http://www.cisco.com/c/dam/en/us/products/collateral/cloud-systems-management/prime-fulfillment/hr_cisco_customer_centric.pdf

[CMON] **CMon: Multi-Domain Circuit Monitoring System Based on GÉANT perfSONAR MDM**
https://www.researchgate.net/publication/289344863_CMon_Multi-Domain_Circuit_Monitoring_System_Based_on_GEANT_perfSONAR_MDM?ev=prf_high

[EDA] **Programming without a call stack – Event-driven architectures**
http://www.enterpriseintegrationpatterns.com/docs/EDA.pdf

[eduGAIN] http://www.geant.org/Services/Trust_identity_and_security/eduGAIN

[eTOM] https://www.tmforum.org/business-process-framework/

[ETSI NFV] http://www.etsi.org/technologies-clusters/technologies/nfv

[ETSI NFV SPEC] "Network Functions Virtualisation (NFV); Management and Orchestration" ETSI GS NFV-MAN 001 V1.1.1 (2014–12)

[FED4Fire] https://www.fed4fire.eu/

[GÉANT LAMBDA] https://wiki.geant.org/pages/viewpage.action?pageId=53117504

[GENI] https://www.geni.net

[GMOSS] **The Guide to Modern OSS, OSS Line, 2016**
http://www.ossline.com/the-guide-to-modern-oss

[GTS] https://www.geant.org/Services/Connectivity_and_network/GTS/Pages/Resources.aspx

[GTS VIRTUALISATION] https://www.geant.org/Services/Connectivity_and_network/GTS/Documents/M6-1_Generalised-Virtualisation-Model-v2016_v02.pdf

[HAZ-2014] M. Hazlinsky, B. Pietrzak, P. Szegedi, F. Farina, J. Sobieski, SDNI: The GÉANT Testbeds Service – Virtual Network Environments for Advanced Network and Applications Research, Science and Technology Conference (Modern Networking Technologies) (MoNeTeC), 2014 International, IEEE, Moscow, Russia, 28-29 Oct. 2014, p. 62-67, ISBN: 978-1-4799-7593-8, DOI: 10.1109/MoNeTeC.2014.6995585, available from http://ieeexplore.ieee.org/abstract/document/6995585/

[HSFC] D. Dolson, S. Homma, D. Lopez, M. Boucadair, Hierarchial Service Function Chaining https://www.google.co.uk/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwjk1aSK2L7YAhXB2KQK

HZlIBCsQFggpMAA&url=https%3A%2F%2Ftools.ietf.org%2Fhtml%2Fdraft-ietf-sfc-hierarchical-05&usg=AOvVaw3CkdBOpd_N3PZKyO99uGjT

[IEEE] R. A. Dourado, L. N. Sampaio, J. A. Suruagy Monteiro, "On the composition of performance metrics in multi-domain networks", IEEE Communications Magazine, vol.51, no.11, pp.55-62, November 2013

[ITIL] **ITIL and IT Service Management**
http://www.itil.org.uk/

[ITU-1540] "Internet protocol data communication service – IP packet transfer and availability performance parameters, ITU-T Recommendation Y.1540", March 2011.

[ITU-1541] "Network performance objectives for IP-based services", ITU-T Recommendation Y.1541, December 2011.

[jFED] http://jfed.iminds.be/

[JRA2, T2] https://intranet.geant.org/gn4/2/Activities/JRA2/T2/SitePages/Home.aspx

[L3-VPN] https://www.geant.org/Services/Connectivity_and_network/Pages/VPN_Services.aspx

[M8.1] (Link for GN4-2 project participants)
https://intranet.geant.org/gn4/2/Activities/JRA2/Milestones%20Documents/Consolidated%20Connection%20Services%20Roadmap/M8.1_Consolidated-Connection-Services-Roadmap.pdf

[M8.2] (Link for GN4-2 project participants)
https://intranet.geant.org/gn4/2/Activities/JRA2/Milestones%20Documents/Service%20Provider%20Architecture%20Roadmap/M8%202_Service-Provider-Architecture-Roadmap.pdf

[M8.3] (Link for GN4-2 project participants)
https://intranet.geant.org/gn4/2/Activities/JRA2/Milestones%20Documents/G%C3%89ANT%20Testbeds%20Service%20Roadmap/M8.3_GE%CC%81ANT-Testbeds-Service-Roadmap.pdf

[M8.4] (Link for GN4-2 project participants)
https://intranet.geant.org/gn4/2/Activities/JRA2/Milestones%20Documents/Network%20Monitoring%20-%20Performance%20Verification%20Architecture%20Roadmap/M8.4_Network-Monitoring-Performance-Verification-Arch-Roadmap.pdf

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

[MEFEL] **ELINE MEF Reference Wiki**
https://wiki.mef.net/display/CESG/E-Line

[MEF-10.3] **MEF Technical Specification**
https://www.mef.net/Assets/Technical_Specifications/PDF/MEF_10.3.pdf

[NFV] **NFV Network Monitoring Framework Diamond**
https://www.ericsson.com/research-blog/sdn/nfv-network-monitoring-framework-diamond

[NML] **Network Markup Language Working Group**
https://redmine.ogf.org/projects/nml-wg

[NSI] https://www.google.co.uk/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=0ahUKEwiBrZLB8b7YAhWRGuwKHYWFCVkQF

ggtMAE&url=https%3A%2F%2Fwww.ogf.org%2Fdocuments%2FGFD.212.pdf&usg=AOvVaw0rvM-Y1U5PukD-Jyb-Jo77

[ODE] **Open Digital Ecosystem, TM Forum** https://www.tmforum.org/%0bopen-digital-ecosystem-2/

[OGF] https://www.google.co.uk/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwiDicGN-L7YAhWC3KQKHb9eDyIQFggnMAA&url=https%3A%2F%2Fwww.glif.is%2Fmeetings%2F2012%2Ftech%2Fslides%2F20121012-OGF36-NSI-AG.pdf&usg=AOvVaw0wBhfxyRmgFFtLb5E1P2zH

[ONOS] https://onosproject.org/

[OpenFlow] https://www.opennetworking.org/sdn-resources/openflow and controller options https://osrg.github.io/ryu/]

[OpenNSA] https://github.com/NORDUnet/opennsa

[OpenVSwitch] http://openvswitch.org

[perfSONAR] http://www.perfsonar.net/

[SDI-2016] Looking Beyond the Internet, March 28, 2016, available from https://lookingbeyondtheinternetblog.wordpress.com/

[SNTS] **SIG-NOC Tools Survey 2016** https://www.geant.org/Resources/PublishingImages/SIG-NOC%20Tools%20Survey%202016.pdf#search=sig%20noc

[SOA] **SOA Blueprint – SOA Practitioners Guide** http://www.soablueprint.com/practitioners_guide

[SOB-2015] Jerry Sobieski, Susanne Naegele-Jackson, Blazej Pietrzak, Michal Hazlinsky, Fabio Farina and Kim Kramaric, GÉANT Testbeds Service (GTS) - GÉANT Testbed Service - External Domain Ports: A demo on multiple domain connectivity, European Workshop on Software Defined Networks (EWSDN), Bilbao, Sept. 30 - Oct. 2, 2015

[SOB-2015a] Jerry Sobieski, Susanne Naegele-Jackson, Blazej Pietrzak, Michal Hazlinsky, Peter Szegedi and Fabio Farina, GÉANT Testbeds Service - Virtual Network Environments for Advanced Network Research, TERENA Networking Conference TNC15, Porto, Portugal, June 14-18, 2015, https://tnc15.terena.org/core/poster/4 (abstract TNC_SA2_GTS_2015_Poster.pdf)

[SQMtool] http://geant3plus.archive.geant.net/Resources/Deliverables/Documents/D8-6_DS4-3-2_Refactoring-GN3-Tools-based-on-BPM.pdf

[SUR] **2014 TM Forum Frameworx Adoption Survey Results** https://www.tmforum.org/wp-content/uploads/2014/09/Frameworx-Adoption-Survey-Report-2014.pdf

[TMFF] **TM Forum Frameworx** https://www.tmforum.org/tm-forum-frameworx/

[TMF Open] https://projects.tmforum.org/wiki/display/API/Open+API+Table

[TM Forum (TMF)] https://www.tmforum.org/

[TR244] https://www.tmforum.org/resources/technical-report/tr244-tm-forum-information-framework-enhancements-to-support-zoom-r15-0-0/

Glossary

AAI	Authentication and Authorisation Infrastructure
AFU	Automated Fault analysis Utility
AM	Active Modification
API	Application Programming Interface
BoD	Bandwidth on Demand
BMS	Bare Metal Servers
BPM	Business Process Management
BSS	Business Support System
CCS	Consolidated Connection Services
CCV	Customer-Centric View
COTS	Commercial off-the-Shelf
CPU	Central Processing Unit
c/r	checkpoint/restart
CRM	Customer Relationship Management
CSF	Communications Sub-system Failure
CSP	Communication Service Provider
CY	Calendar Year
DSL	Digital Subscriber Line
DSP	Demand Side Platform
EDA	Event-Driven Architecture
EFTS	Ethernet Flow Termination System
EPL	Ethernet Private Line
ESB	Enterprise Service Bus
eTOM	enhanced Telecom Operations Map (TM Forum Business Process Framework)
EVPN	Ethernet Virtual Private Network
FAB	Fulfilment Assurance and Billing
FCAPS	Fault, Configuration, Accounting, Performance and Security
GENI	Global Environment for Network Innovations
GN3plus	GÉANT Network 3 plus, a project part-funded from the EC's Seventh Framework Programme under Grant Agreement No.605243
GN4-1	GÉANT Network 4, Phase 1 project part-funded from the European Union’s Horizon 2020 research and innovation programme under Grant Agreement No.691567
GN4-2	GÉANT Network 4, Phase 2 project part-funded from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No.731122
GTS	GÉANT Testbeds Service
GUI	Graphical User Interface
GVM	Generalised Virtualisation Model

IAGW	Internet Access GateWay
IP	Internet Protocol
ITIL	Information Technology Infrastructure Library
ITU	International Telecommunication Union
JRA	Joint Research Activity
JRA2	Joint Research Activity 2 Network Services Development
Layer 2	In the OSI model, Layer 2 is the data link layer
Layer 3	In the OSI model, Layer 3 is the network layer
LTE	Long-Term Evolution (standard for high-speed wireless communication for mobile phones and data terminals)
LTS	Long-Term Support
M	Month
MANO	Management and Orchestration
mDC	microDataCenter
MDVPN	Multidomain Virtual Private Network
MEF	Metro Ethernet Forum
MPLS	Multiprotocol Label Switching
NFV	Network Function Virtualisation
NIC	Network Interface Card
NIF	New Ideas Forum
NML	Network Management Language
NMS	Network Management System
NOC	Network Operations Centre
NREN	National Research and Education Network
NSC	Network Service Chaining
NSI	Network Service Interface
ODE	Open Digital Ecosystem
OFX	OpenFlow eXperimental (facility)
ONOS	Open Network Operating System
OSS/BSS	Operations Support Systems/Business Support Systems
OTN	Optical Transport Networking
OVS	Open vSwitch
OWAMP	One-Way Active Measurement Protocol
OXP	Open eXchange Points
P2P	Point to point
PLE	PlanetLab Europe
PLM	Product Lifecycle Management
PVM	Performance Verification and Monitoring
QoS	Quality of Service
R&D	Research and Development
R&E	Research and Education
RCA	Resource Control Agent
RCA-VC	Resource Control Agent Virtual Circuits
SA	Service Activity
SBI	Southbound Interface
SDN	Software Defined Networks
SFA	Slice-based Facility Architecture

SGA	Single Grant Agreement
SID	Shared Information Data
SNTS	SIG-NOC Tools Survey
SOA	Service-Oriented Architecture
SPA	Service Provider Architecture
SQM	Software Quality Management
T	Task
TAM	Telecoms Application Map (application framework)
TMF	Tele-Management Forum
TRL	Technology Readiness Level
TTS	Text to Speech
VC	Videoconference
VFC	Virtual Forwarding Context
VIM	Virtual Infrastructure Management
VLSI	Very-large-scale integration
VM	Virtual Machine
VNF	Virtual Network Function
VPLS	Virtual Private Local Area Network
VPN	Virtual Private Network
VPWS	Virtual Private Wire Service
WAN	Wide-Area Network