

19-02-2016

## Deliverable D8.2

# Production Services Common Components and Processes

### Deliverable D8.2

Contractual Date:	29-02-2016
Actual Date:	19-02-2016
Grant Agreement No.:	691567
Activity:	8/SA4
Task Item:	Task 3
Nature of Deliverable:	R (Report)
Dissemination Level:	PU (Public)
Lead Partner:	AMRES
Document Code:	GN4-1-16-8D4C0
Authors:	P. Vuletić (UoB/AMRES), L. Hrboka (CARNet), I. Golub (CARNet), M. Mamalis (GRNET), I. Garnizov (DFN), D. Schmitz (DFN), J. Vuleta (UoB/AMRES), N. Ninković (UoB/AMRES), B. Jakovljević (UoB/AMRE)

© GÉANT Limited on behalf of the GN4 Phase 1 project.

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 691567 (GN4-1).

#### Abstract

In this document, SA4 T3 proposes changes to the current GÉANT Product Lifecycle Management (PLM) framework to allow the inclusion of Continual Service Improvement. Several service improvement opportunities have also been analysed, ranging from specific process improvements to the comparison and consolidation of cross-service functionalities.

# Table of Contents

Executive Summary	1
1 Introduction	2
2 Continual Service Improvement in GÉANT	4
2.1 Principles of the ITIL CSI Approach	4
2.2 CSI in the GÉANT project	5
2.2.1 Information sources for GÉANT services	5
2.3 GÉANT CSI Register	5
2.4 CSI in future GÉANT projects	6
2.4.1 Inputs into the CSI stage from other lifecycle phases	7
2.4.2 Outputs from the CSI stage to other lifecycle phases	7
3 perfSONAR and Service Quality Management	8
3.1 Current status	8
3.2 Issues and improvement opportunities	9
3.3 Improvement proposal	9
4 MDVPN Service Inventory Improvement	11
4.1 Current status	11
4.2 Issues and improvement opportunity	11
4.3 Improvement description and proposal for further improvements	12
5 MDVPN service configuration improvement	14
5.1 Current status and issues	14
5.2 Improvement proposal	14
6 Conclusion	16
Appendix A GÉANT CSI Register	17
Appendix B MDVPN Service Inventory	19
B.1 MDVPN Service Inventory – architecture and software stack	19
B.2 MDVPN Service Inventory – functionalities	21
Appendix C Example of MDVPN Service Configuration	32
C.1 Building a MDVPN service (L3VPN) with Ansible	32
C.1.1 Building the variables file	33

## Table of Figures

Figure 2-1: Proposed position of CSI in the Service Delivery Process	6
Figure B.1: MDVPN Service Inventory – architecture	19
Figure B.2: MDVPN SI Web application – interactions	21
Figure B.3: MDVPN SI welcome page	22
Figure B.4: New Network Service Provider Workflow: Step 1	23
Figure B.5: New Network Service Provider Workflow: Step 2	23
Figure B.6: New Network Service Provider Workflow: Step 3	24
Figure B.7: New Network Service Provider Workflow: Summary	25
Figure B.8: Network Service Provider Overview	26
Figure B.9: New VPN Service Instance Workflow: Step 1	27
Figure B.10: New VPN Service Instance Workflow: Step 2	28
Figure B.11: New VPN Service Instance Workflow: Summary	29
Figure B.12: SSP Report Export as an Excel File	30
Figure B.13: SSP Report Export as a PDF File	31
Figure B.14: SSP Report Export as an XML File	31

## Table of Tables

Table A.1: GEANT CSI Register log	18
-----------------------------------	----

## Executive Summary

Continual Service Improvement (CSI) is one of the key stages of the service lifecycle, which focuses on increasing the efficiency and decreasing the cost of services in the underlying service management processes.

GN4-1 SA4 T3 represents the first time that service improvement has had its own dedicated task in a GÉANT project. In this deliverable, the Task proposes changes to make CSI a full part of the GÉANT Product Lifecycle Management (PLM) framework. Before adoption, the introduction of these changes would need to be fully considered in the GN4-2 period.

Besides the general proposal for fitting the CSI stage into the GÉANT service architecture, several service improvement opportunities have been analysed, ranging from specific process improvements to the consolidation of cross-service functionalities. These improvement opportunities target the MDVPN service, which was in the early stages of its operations phase, with processes still to be introduced or improved towards a higher level of automation. The focus in this deliverable is on developing the MDVPN Service Inventory, enabling scalable MDVPN Service Quality Management (SQM) and automating service configuration. Specific technical solutions and appropriate recommendations are presented.

# 1 Introduction

The main objective of the GN4-1 SA4 T3 (Production Optimisation and Continuity) Task is the continuous improvement of production services and infrastructure. SA4 T3 analyses and proposes improvement recommendations to services that are either in production or in transition towards production within the SA4 activity. This Deliverable describes the results of an analysis performed for the services that are currently in production or in the service validation and testing phase.

Potential service improvement opportunities and recommendations range from specific process improvements to the comparison and consolidation of cross-service functionalities. The improvement process is not straightforward. Improvement opportunities may result from subjective judgement by some stakeholders in the service lifecycle, though they may be resisted by others, even after an external analysis is performed. Therefore, SA4 T3 investigated the use of current best practices in order to propose a methodical and objective approach to service improvement. During the course of the work, ITIL Continual Service Improvement (CSI) [[CSI](#)] was used as a reference model for the structured approach to all the activities within the task.

This is the first time that service improvement has had its own dedicated task in a GÉANT project. One of the objectives for this task is to eventually establish a service improvement procedure in the GÉANT service environment, for full consideration in the GN4-2 period. One of the key components of CSI is a CSI register, which is presented in this document as Appendix A. The task analysed several improvement opportunities which, as an example, target the MDVPN service. The MDVPN service is at the beginning of its service operations phase, and there are several processes that still need to be introduced or improved to provide a higher level of automation.

The first improvement opportunity is the design and development of the MDVPN Service Inventory: a data repository of all service instances, key service resources and service actors. Without the Service Inventory, relevant data would be scattered at different places, stored on paper or in a spreadsheet, which makes the execution of typical operational processes error-prone and time consuming. For example, resolving a problem in a particular MDVPN instance requires finding out about the resources and actors relevant for that service instance right from the start.

The second improvement opportunity relates to the Service Quality Management (SQM) process for the MDVPN service. In GN3plus, SA4 T3 developed an SQM supporting tool for the MDVPN service. It has a similar architecture to the existing perfSONAR monitoring system, but is more scalable for multi-point multi-instance services. It is capable of tracking Service Level Agreement (SLA) parameters and making a distinction between the measurements coming from different service instances. Having multiple similar systems (the SQM tool and perfSONAR) is not optimal, since it requires additional

knowledge to use both tools, additional hardware costs and additional maintenance effort. Therefore the task analysed the opportunities for integrating the SQM capabilities with perfSONAR to solve these issues.

The third improvement opportunity concerns the configuration management process for MDVPN. Each new MDVPN service instance involves multiple domains: the GÉANT network and NREN networks involved in that particular service instance. Setting up a new service instance requires the configuration of the relevant network elements in each of the involved domains. Service configuration is carried out manually, with the communication between the actors typically done by email, with each domain creating its own configuration from scratch. This complexity results in longer service delivery times, and is prone to configuration errors. The task analysed the opportunities for automating the service configuration process as much as possible, taking into account the constraints imposed by the multi-domain environment and the autonomous management of all the domains involved.

In this document, Section 2 gives an overview of the main ITIL CSI concepts in the GÉANT environment, and proposes the integration of the CSI phase with the Product Lifecycle Management framework. Section 3 describes the improvement opportunity of integrating SQM capabilities with perfSONAR. Section 4 describes the MDVPN Service Inventory and Section 5 describes the potential for automating the configuration management process for the MDVPN service.

## 2 Continual Service Improvement in GÉANT

Service design and development within the GÉANT project is well organised, monitored and documented through the GÉANT Product Lifecycle Management (PLM) Framework [[PLMportal](#)]. Key service lifecycle phases (Strategy, Design, Transition and Operations) are properly mapped onto specific actors in various tasks and activities, and key procedures for the transition between service lifecycle phases are well defined. However, in previous GÉANT projects there were no tasks and activities entirely devoted to Continual Service Improvement (CSI). The improvements and changes to the existing services were made typically through the design of new features in tasks devoted to service design. Due to the inherent problems with the adoption of the service improvements mentioned in the Introduction, SA4 T3 recognised that the current framework has room for improvement by adding and formalising the role of the Continual Service Improvement stage.

### 2.1 Principles of the ITIL CSI Approach

Service improvement focuses on increasing the efficiency and decreasing the cost of services and the underlying service management process. The service improvement approach starts with answering a set of questions about the overall vision of the service, current and target status and positioning, improvement opportunities and steps, and service metrics to measure the achievement and success of the implemented improvement. Such an approach, with its reliance on service metrics, ensures that objective improvement opportunities are obtained and the gains are properly predicted and assessed.

The Information Technology Infrastructure Library (ITIL) CSI approach is based on a seven-step process that guides each improvement cycle. It includes identifying the strategy for improvement, defining what will be measured, gathering, processing and analysing data to get valuable information, followed by the presentation and use of the information towards improving implementation. The process is supported with methods and techniques including assessments and benchmarking. Best practices outline the importance of service measurement focusing on technology, process and service metrics.

The role of CSI Manager is of central importance. The manager is accountable for the successful implementation of the improvement programme and communicates the CSI vision across the organisation. The CSI Manager works with service owners, service managers, process managers and practitioners to identify improvement opportunities and carries them out by following the CSI process and ensures that knowledge gathered from the CSI process is managed properly.

## 2.2 CSI in the GÉANT project

Service assessments or benchmarking and the whole CSI stage itself can be performed as self-assessment by actors that are already involved in some service lifecycle phases or by some external actors including external bodies. Self-assessments and internally-organised CSI, although usually cheaper, can suffer from the influence of internal political pressures, and might result in a biased interpretation of the assessment results and a loss of objectivity. Assessment often requires a fresh set of eyes. In GÉANT projects, a dedicated task which is not involved in the other service lifecycle phases could be the optimal solution, achieving both cost effectiveness and an objective service assessment.

In the GN4-1 project, SA4 T3 has the role of CSI manager for services in production in SA4. For each service, SA4 T3 participants can personally take the role of CSI managers. Since SA4 is a new activity in GN4-1, and T3 a completely new task, it can work in parallel on the establishment of the service improvement process in accordance to the current best practices, and the plans for the next phases of the project.

### 2.2.1 Information sources for GÉANT services

SA4 T3 analysed available information sources for four GÉANT services: MDVPN, FaaS, eduPKI and perfSONAR. The starting point for the data gathering was the GÉANT Product Management Portal [[PLMportal](#)] which contains typical Service Strategy outputs like: Service Catalogue, Service Business Cases, Cost Benefit Analyses, Service descriptions, etc. Some services have a dedicated website (e.g. [[eduPKI](#)]), or a website maintained outside of the GÉANT realm (e.g. [[perfSONAR](#)]) where some additional information about the service can be found, while other services still don't have dedicated web sites. The GÉANT Product Management Portal also maintains dynamic monitoring of some service Key Performance Indicators (KPIs) that aim to show the progress and success of the service.

Documents that are the common output of the design or operations phase which describe key process flows (e.g. a new service request, change request, problem resolution process, etc.) and operational procedures that are typically not available to all project members or are otherwise difficult to find. eduPKI has publicly available only the service registration process [[eduPKIreg](#)]. The FaaS service has some of these documents stored at the GÉANT wiki pages, but these can only be accessed by members of SA5 T4. The MDVPN operational cookbook is also not available publicly to all project participants, but is shared among the members of the Task. SA4 T3 has gathered and used all these documents to analyse improvement opportunities.

## 2.3 GÉANT CSI Register

SA4 T3 organised a basic CSI register, a record of all service improvement opportunities, their descriptions, size, priority, timescale and other relevant information about the key stakeholders involved in each particular service opportunity. The register is presented in Appendix A of this document. Three of the improvement opportunities are described in detail in sections 3 to 5. The

fourth opportunity came in the second half of the project and will be analysed in the final months of the project, following the submission of this deliverable.

## 2.4 CSI in future GÉANT projects

The GÉANT PLM portal [PLMportal] defines the GN4 service delivery process. It presents the workflow from innovation to development and production phases and defines responsibilities of particular GN4 Activity. Note that the current PLM process ends with service production/operations. All IT services in today's dynamically changing technology and business environment require constant adjustment and improvement according to the Deming circle<sup>1</sup> and therefore implementing CSI is the expected norm for a successful product.

SA4 T3 recommends the amendment of the Service delivery process flow between CSI and other lifecycle stages in the GN4 PLM environment. The workflow proposal is shown in Figure 2-1.

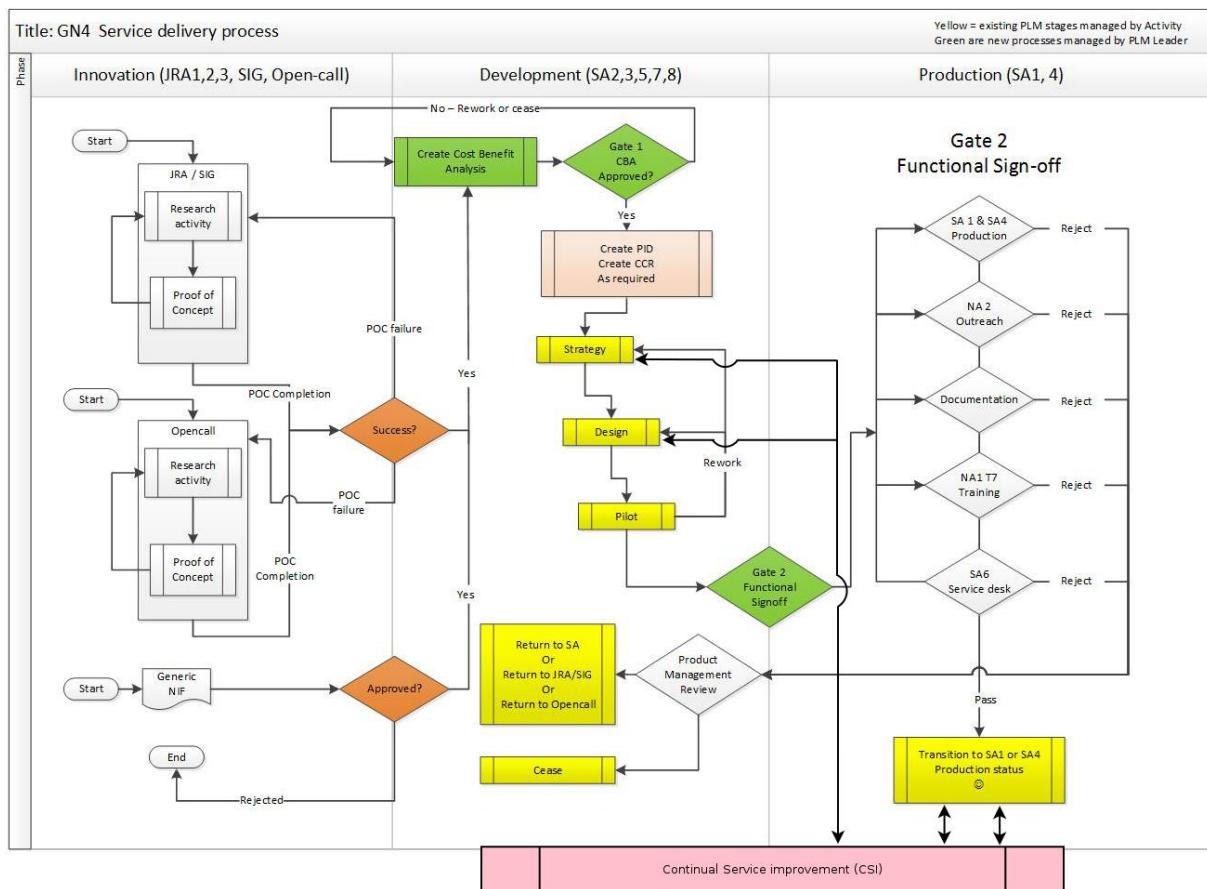


Figure 2-1: Proposed position of CSI in the product lifecycle

<sup>1</sup> Or Deming cycle: PDCA (plan–do–check–act or plan–do–check–adjust), an iterative four-step management method used in business for the control and continuous improvement of processes and products.

Continual Service Improvement needs to be carried out proactively in every stage of the product lifecycle and requires constant interaction between CSI and the other service lifecycle stages. The set of inputs and outputs between CSI and other product lifecycle phases is described in sections 2.4.1 and 2.4.2.

## 2.4.1 Inputs into the CSI stage from other lifecycle phases

Service improvement requires all the relevant data from all service lifecycle phases to be available. The list of the documentation relevant for the service improvement includes:

- From strategy: vision and mission, service portfolio, policies, strategic plans, priorities, achievements against metrics KPIs and Critical Success Factors (CSFs).
- From design: Service catalogue, Service Design Package (SDP), design of services, measurements, processes, infrastructure and systems, SLA, OLA, underpinning contracts.
- From transition: test reports, change evaluation reports.
- From operations: operational performance data and service records, proposed problem resolutions.

## 2.4.2 Outputs from the CSI stage to other lifecycle phases

This section defines some of the major outputs from the CSI stage to other lifecycle stages as recommended by ITIL:

- To Strategy: results of customer and user satisfaction surveys, input to business cases and the service portfolio, feedback on strategies and policies, financial information regarding improvement initiatives for input to budgets, data required for metrics, KPIs and CSFs, service reports, Requests For Change (RFCs) for implementing improvements.
- To Design: results of customer and user satisfaction surveys, input to design requirements, data required for metrics, KPIs and CSFs, service reports, feedback on service design packages, RFCs for implementing improvements.
- To Transition: results of customer and user satisfaction surveys, input to testing requirements, data required for metrics, KPIs and CSFs, input to change evaluation and change advisory board meetings, service reports, RFCs for implementing improvements.
- To Operations: results of customer and user satisfaction surveys, service reports and dashboards, data required for metrics, KPIs and CSFs, RFCs for implementing improvements.

These are outputs that should be provided to the PLM team and the PLM process defined in Figure 2-1, in order to bring embed CSI results in the GÉANT service management structure, as well as processes, service and products.

## 3 perfSONAR and Service Quality Management

### 3.1 Current status

The Service Quality Management (SQM) supporting tool was designed, and developed in the GN3plus project as an SLA performance-monitoring system, following specifications in ITU-T Y.1540 and Y.1541, tailored primarily for the MDVPN service. SLA parameters measured are packet latency, latency variation (jitter) and packet loss rate (PLR). The system includes common components for active measurement architecture: measurement agents, measurement data collector and controller (which is in this case service inventory as well).

Multi-Domain VPN (MDVPN) is a service designed in the GN3plus project for the establishment of L2 and L3 VPNs across the GÉANT and NREN environment. It is the primary use case where the SLA monitoring of SQM was applied. The nature of the service required availability and operational monitoring in different VPN instances from the participating partners. Such monitoring is technically possible by the service providers themselves, but in a traditional approach and with the current state-of-the-art, it would require a dedicated measurement appliance for every VPN at every partner.

The approach taken by the SQM development team was to build a versatile measurement agent SQM-MA, which could be applied by each partner and provide performance measurement in multiple VPNs, where they are present. The benefit of this, besides the simplicity and efficiency in measurement agent deployments, is the ability to centrally manage and monitor performance across all established channels of the MDVPN service. The measurement tool chosen for performance monitoring is OWAMP<sup>2</sup> and the technology that is used to switch in the different isolated VPN networks is Linux Network Namespaces [NN].

OWAMP itself is part of the perfSONAR development effort, where GÉANT participates in a joint partnership with ESNet<sup>3</sup>, Internet2<sup>4</sup> and Indiana University with a dedicated team. perfSONAR is a performance monitoring system with multiple supporting services around it to maintain an open and collaborative environment for performance measurements and issue diagnosis.

Both perfSONAR and SQM prototypes have a similar architecture, use similar measurements (SQM uses a subset of perfSONAR's set of measurements) but it handles measurements and data storage in

---

<sup>2</sup> <http://software.internet2.edu/owamp/>

<sup>3</sup> <http://www.es.net/>

<sup>4</sup> <http://www.internet2.com>

a slightly different way. Having more than one similar network and service monitoring systems is not optimal: it requires additional maintenance effort, additional knowledge to use both tools and additional costs for hardware. Based on these facts and feedback from the community, SA4 T3 decided to explore the integration of the two systems. This should improve the perfSONAR performance monitoring system of GÉANT by bringing versatile connectivity to the perfSONAR MP instances and extend the service portfolio with the service quality monitoring of SQM.

In the course of this effort, both systems have been analysed to identify the components that deliver similar functionality in an attempt to avoid duplicate development effort. Besides the low-level instrument for performance measurements, OWAMP, which is part of the measurement agent component, the team looked into the measurement archive module, central measurements' management system and the visualisation systems component. An additional improvement to perfSONAR's measurement initiation and operation was also identified and reported.

## 3.2 Issues and improvement opportunities

This improvement opportunity does not target new performance metrics, but rather looks at the optimisation of the common components and processes in existing tools. The team researched the possibilities of integrating the reviewed tools into a more versatile and efficient product.

Initially two general scenarios were investigated. The first scenario proposed the loose integration of the two systems which aims to keep the performance measurement scheduling for perfSONAR and SQM separate in attempt to avoid the complexity of the data structure introduced into perfSONAR services. What would be shared in this scenario by the two systems are measurement agents and measurement collectors. The second scenario was aiming for a tighter integration, where even supporting services would require metadata processing.

The detailed review showed that although perfSONAR 3.5.x share the same measurement tool as the one used in SQM, the current version of perfSONAR does not enable central coordination of measurements initiated for multiple isolated virtualised interfaces (i.e. per-VPN measurements), and thus cannot fully support the MDVPN measurement use case. This presents one possible area for improvement.

An improvement opportunity with perfSONAR and SQM integration was also found in the visualisation of the measurement results and status. perfSONAR features two distinct types of graphical user interface. One is the integrated perfSONAR toolkit and MADDASH GUI; the other is the perfSONAR UI service, implemented as a central measurement manager and viewer of the perfSONAR archive, which also allows extensions through modules. Neither of these GUIs was ready to support the MDVPN functional requirements.

## 3.3 Improvement proposal

The two SA4 teams – T2 (Production and Support) and T3 (Production Optimisation and Continuity) – analysed the two options and concluded that tighter integration between perfSONAR and SQM

solution would be better in the long term, although it might require more time to complete initially. Integration involves the following elements of perfSONAR:

- BWCTL daemon and interface sharing – The role of the BWCTL daemon is to become the *de facto* coordinator of the measurements in the perfSONAR suite. It will coordinate the initiation of the tests across multiple instances and gathers and manages inquiries from them. The difficulty here comes from the fact that there are multiple services running in an isolated process context as it works with Linux namespaces.
- BWCTL client – The BWCTL client is the tool used to initiate performance measurements. Since MDVPN operates in different VPN namespaces, the goal is to be able to initiate those measurements in different VPNs. In order to achieve this, it needs to instruct the client to initiate the correct VPN/Linux namespace context before it requests the test.
- Measurement archive – perfSONAR MA is the component that stores and delivers the measurement data along with its metadata. The perfSONAR development team confirmed that the metadata can be extended to allow procession of additional identifiers in order to separate the measurement results in the distinct VPN spaces. It is also important that the different visualisation and data analysis services are able to request and retrieve the data using these identifiers.
- Measurements configuration management – In the current perfSONAR 3.5 release, there are two different channels for measurement configuration management – the configuration\_daemon [[pSCD](#)] and the central “mesh configuration” [[pSCMC](#)]. The first one is dedicated to local toolkit management and the second one is responsible for centralised service management. Both of these services incorporate their instructions for measurements in the regular\_testing\_service [[pSRTS](#)]. With the MDVPN use case in mind, the team has proposed extending the test parameters to reflect the new schema.
- Visualisation – The perfSONAR development supports two different user interfaces. One is integrated with each perfSONAR toolkit instance and provides a direct view on the measurement results stored in the local measurement archive. The other is the centralised management service of perfSONAR UI, which additionally allows for remote initiation of performance measurements between two toolkit instances. The SA4 T3 team found the centralised solution of perfSONAR UI to be more appropriate for two reasons: it fits more natively to the requirements of the MDVPN use case and the development effort is less. The analysis of the SQM implementation identified that the following features should be sought from a possible future implementation.
  - Retrieval of MDVPN administrative data and SLA thresholds from the service inventory SQM component.
  - SLA verification procedures.
  - A dashboard representation of the current status.

In terms of the expected impact, the global community would be offered easily extensible performance monitoring of the core monitoring infrastructure deployed by GÉANT.

A rough estimate of the effort needed from the development team for this initiative is about 6 PM, and is planned to be delivered with the next major release of perfSONAR 3.6.

## 4 MDVPN Service Inventory Improvement

### 4.1 Current status

MDVPN is a new service which was designed during the GN3plus project. It provides L2 and L3 VPN networks which connect multiple end institutions attached to different NRENs. The service is based on a carrier supporting MPLS VPN technology. The MDVPN service operation is distributed among multiple actors: the GÉANT network operations team and operations team of every NREN that participates in the MDVPN service delivery contribute to service support. In such an environment, effective coordination and communication between all partners NOCs is crucial and affects overall service delivery performance. To achieve seamless collaboration and communication, MDVPN providers must have a common view on MDVPN service infrastructure and existing resources. During the design and transition phase, a basic MDVPN resource and service instance registry was implemented using several spreadsheets and was shared among MDVPN task members.

### 4.2 Issues and improvement opportunity

An MDVPN distributed model of work, with many different administrative domains requires a shared high-level view of infrastructure from all partners and good coordination between them. In order to achieve these aims, there must be a MDVPN centralised repository for storing technical information of all service components important for service delivery. This information is necessary for the smooth service operations and provisioning (e.g. information related to VPN service providers, involved persons, configured devices, activated service instances, connected sites, involved projects etc.).

The existing spreadsheet-based storage was non-scalable as the number of MDVPN service instances and the amount of data continued to grow. The absence in the registry of key MDVPN resources, and the lack of possibility to easily search, report and update the information about the service instances was recognised as major difficulty for service maintenance during the transition phase. A proper service inventory is a crucial database which enables the automation of various service operation processes through providing the relevant data for other OSS components supporting service lifecycle processes (e.g. MDVPN Service Quality Management).

In order to help NOCs to maintain MDVPN service easier, to collect accurate and reliable information about infrastructural components and to demonstrate service achievements and usefulness, the development of an MDVPN Service Inventory (SI) is proposed. The MDVPN task within GN3plus decided to switch to a relational database, and a data model for the PostgreSQL DMBS was designed

within the GN3plus project. PostgreSQL implements a role-based access control model with a data model that is access-control agnostic. Although relational databases have numerous advantages over Excel files and spreadsheets that were in use (such as concurrent multiuser access, referential integrity, database normalisation, various transaction isolation levels, etc.) using SQL queries for data management on a daily basis is still uncomfortable and requires at least basic knowledge about relational databases.

The next step towards more efficient and user friendlier data management was to develop a multi-tiered Service Inventory system with a Web user interface that can handle concurrent user requests. Such a system provides all the improvements brought by DB introduction, as well as a GUI and user input validation. It can be augmented with policy-based access control and integrated with eduGAIN via SAML.

## 4.3 Improvement description and proposal for further improvements

The MDVPN Service Inventory developed in SA4 T3 supports data management related to the MDVPN infrastructure, VPN service instances, VPN instance endpoints, and the development of a reporting module. The detailed description of the system developed is given in Appendix B. When the tool passes the validation tests in SA4 T1 (Service Validation and Testing), MDVPN SI will be integrated into the GÉANT Partners portal.

Some further improvements of the tool are possible through the changes of the persons' data model and the improvement of the person management elements. The MDVPN SI data model was designed from the perspective of resources and MDVPN service instances and the abstraction 'person' was treated as a placeholder for contact data without a clear distinction of the scope of the information. Furthermore, the data management workflow neglected the possibility of having the same person involved in several projects or being member of several sites or NSPs. These omissions led to "person duplication". Further MDVPN SI improvements must include access control management and continuous integration server activation. Access control management will be achieved by deploying an open source security framework, e.g. PicketLink<sup>5</sup>, once the MDVPN task specifies requirements regarding access control management. PicketLink is a JBoss umbrella project for security and identity management for Java Applications. It provides Single Sign On (SSO) for web applications and includes an XACML<sup>6</sup> policy evaluation engine and a number of Federated Identity standards such as SAML, WS-Trust and OpenId. Adding authentication to MDVPN SI, i. e. SAML interconnection to eduGAIN, should not require more than 1 PM, while the time needed for writing authorisation policies will depend on requirements.

MDVPN SI should also provide relevant data for other OSS components supporting service lifecycle processes such as MDVPN Service Quality Management (SQM). Integration with an SQM tool must include data model harmonisation and inter-system interface design and implementation. The effort

<sup>5</sup> <http://picketlink.org>

<sup>6</sup> [https://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=xacml](https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml)

required for MDVPN SI and SQM integration mostly depends on the technologies used for interface development but should not exceed 2PM.

## 5 MDVPN service configuration improvement

### 5.1 Current status and issues

The MDVPN service uses the same infrastructure as GÉANT IP service<sup>7</sup>, but with the addition of two VPN route reflectors. In general, the technology involved is considered to be more complex than an IP service and NOC staff may be less familiar with this compared to an IP service.

This complexity may result in delays to service delivery time (a key service metric), but this can be improved by introducing configuration automation to the provision stage of each new service. Configuration automation can provide an easily configurable new service from each new request, making the MDVPN service more appealing to the end user.

### 5.2 Improvement proposal

Currently, each new service request is initiated and completed through emails. A typical example of a MDVPN new instance configuration workflow, extracted from emails, might be:

1. The service customer applies for a new service to the operations manager.
2. The operations manager gathers all data and communicates with all NREN NOCs.
3. The NREN NOCs gather all data from the operations manager and configures the service.
4. The service is delivered to the service customer.

Configuration automation is applied in step 3 where each NOC is responsible for translating the data gathered to a configuration specific for the local PE router to deliver the service.

The typical use case for configuration management and automation is to provide consistency across a large number of devices. In the scope of the MDVPN service, configuration management definition will have to be modified in a way that helps the task without directly interfering in the NRENs' administrative domains and which respects the full autonomy of the domains involved in the service instance. This means that the final configuration will be automatically produced and presented to the NOCs, but without any interaction with the actual network devices.

---

<sup>7</sup> [http://www.geant.org/Services/Connectivity\\_and\\_network/Pages/GÉANT\\_IP.aspx](http://www.geant.org/Services/Connectivity_and_network/Pages/GÉANT_IP.aspx)

The proposal may require changes in the existing MDVPN SI database. MDVPN SI has to store all the necessary data and variables needed to build the configuration automation templates. The actual implementation of configuration automation can be based on one of several open source configuration management tools already available. One such configuration management tool is Ansible<sup>8</sup>: this application works by executing management scripts, written in a data format language while making use of (configuration) templates. The templates themselves are based on vendor-specific configuration procedures, but since no interaction with the end device is needed the automation task itself can be vendor agnostic.

One point of concern is the level of exposure of domain-specific MDVPN data from the MDVPN SI database, given that there is no requirement for a separate database view for each MDVPN administrator. The MDVPN service is based on a well-established model of trust where each PE router is considered secure and managed by NREN NOC personnel. In that way exposing specific data from the MDVPN SI, like PE hostname/IP could be thought of as a security risk from the MDVPN task. To alleviate this concern, different views of the database will have to be build, with user authentication at its core.

Finally, each database user, after completing all the relevant fields, will acquire an automatically generated configuration for final review before committing it to the PE router.

Appendix C presents a simple example on the use case of a new L3VPN service from the MDVPN portfolio. The estimated size of the proposed change is medium (the number of templates to build depends on the MDVPN service portfolio and the diversity of PE routers vendors). The size indicates that some effort will have to be made in building templates for each PE router. Also, each time that a new addition is made to the MDVPN service, the templates will need updating to keep in sync.

---

<sup>8</sup> <http://www.ansible.com/>

## 6 Conclusion

GN4-1 SA4 T3 Task is the first GÉANT task fully dedicated to continual service improvement. Since Product Lifecycle Management framework created in the previous GÉANT project defined key service management procedures and workflows without the task dedicated to CSI, a change to the PLM framework was proposed in order to recognise the role of CSI in service lifecycle and to define the flow of information between the tasks dedicated to the Strategy, Design, Transition and Operations service lifecycle phases and the SA4 T3. Besides establishing CSI procedures which will be continued in future GÉANT projects and CSI register maintenance, SA4 T3 analysed existing services and registered several improvement opportunities for the MDVPN service which are described in this document. SA4 T3 members were, depending on their areas of expertise, CSI managers for various services.

One of these improvement opportunities, inherited from the GN3plus project was fully implemented in SA4 T3. The MDVPN Service inventory was fully designed, developed and put into production. Other two improvements are in a status that requires the discussion with product managers about the time and place (task) these improvement opportunities will be implemented, most probably in the GN4-2 project. In the remainder of GN4-1, SA4 T3 will work towards the identification of further improvement opportunities.

## Appendix A GÉANT CSI Register

Opportunity	Date raised	Size (S,M,L)	Timescale (S,M,L)	Description	Priority (U,1,2,3)	KPI metric	Justification	Raised by	To be actioned by	Date required by
1	24.2.2015.	L	L	MDVPN Service Inventory database: Design and develop a service inventory for the MDVPN service.	U	time from service request to fulfillment, time from problem report to resolution	MD-VPN distributed model of work with many different administrative domains that participate in delivery of service require unified high-level view of overall infrastructure from all partners and some level of coordination between them. In order to achieve these aims, the existence of MD-VPN central repository for storing technical information of all service components important for service delivery is mandatory. This information is necessary for the smooth service operations and provisioning (e.g. information related to VPN service providers, involved persons, configured devices, activated service instances, connected sites, involved projects etc.)	X. Jeanin (SA3T2 TL, MDVPN product manager)	J. Vuleta, D. Schmitz	Jan 1. 2016

2	1.6.2015.	L	L	Resolve overlapping between SQM supporting tool and perfSONAR	1	no KPI metric - optimize network monitoring portfolio of the project	Service Quality Management supporting tool was designed in GN3 plus project as an SLA performance monitoring system tailored primarily for MDVPN service. It has a very similar architecture as perfSONAR and uses the same measurement agent (OWAMP). Unlike perfSONAR, SQM supporting tool is capable to monitor multiple service instances from a single measurement point and to store and distinguish the measurements from different service instances. However, having two such similar systems is not an optimal solution. Therefore the most optimal way to integrate the two has to be found	P. Vuletić	I Garnizov, N. Ninković	Feb 29th 2016	
3	1.6.2015.	M	M	Automate the MDVPN Configuration Management process	3	service delivery time	reduce the time of the flow from service request to fulfillment	M. Mamalis	M. Mamalis, B. Jakovljević	Feb 29th 2016	
4	28.8.2015.	M	M	There is no methodology for the software version change management for FaaS components. Current practice works only for new service instances - new service instances get the latest software version, but installed instances remain with old software versions. That makes the system difficult to be managed as different service instances will have different software versions and are susceptible to different issues and problems	2	change request processing time	Automated change management process, more reliable software updates	A. Scicchitano (SA4T2 TL), FaaS operations manager	Lj. Hrboka	Apr 30th 2016	

Table A.1: GEANT CSI Register log

## Appendix B MDVPN Service Inventory

### B.1 MDVPN Service Inventory – architecture and software stack

MDVPN Service Inventory (MDVPN-SI) is multitier system for data administration about MDVPN infrastructure, VPN service instances and VPN instance endpoints. Its architecture is shown in Figure B.1.

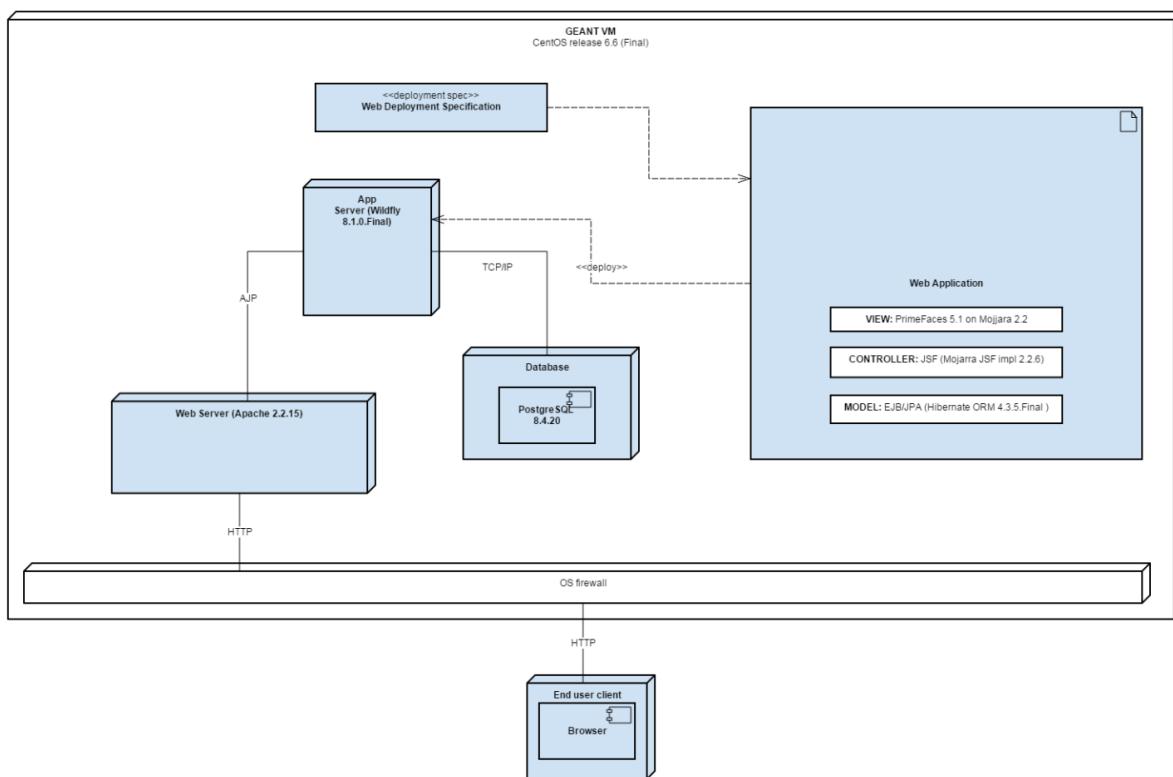


Figure B.1: MDVPN Service Inventory – architecture

Multi-tier architecture, also known as n-tier architecture, is client-server architecture with logically separated application-specific operational tiers, where each tier interacts only with adjacent tiers. Separation of tiers improves flexibility, scalability and performance as each tier can use dedicated

physical resources and can be maintained separately. Three-tier architecture is the most common architecture in web-based systems. MDVPN SI tiers are:

- Data (storage) tier – PostgreSQL<sup>9</sup> database server.
- Data tier stores, manages and retrieves data and exposes it to Logic tier. It must provide API for managing and storing data which is independent of underlying data storage mechanisms.
- (Business) logic tier – Wildfly<sup>10</sup> application server front-ended by Apache HTTP server<sup>11</sup>; Web application deployed on Wildfly application server is part of Logic tier.
- Logic tier is also known as middle tier. It contains all business logic and controls applications' functionalities. It acts as a server to the Presentation tier and as a client to the Data tier.
- Presentation tier – Web browser on enduser client machine; supported browsers are IE7, IE8, IE9, non-legacy versions of Safari, non-legacy versions of Firefox, non-legacy versions of Chrome and non-legacy versions of Opera.

The Presentation tier provides a graphical user interface (GUI) for services provided by multitier system. It partially validates user input, forwards user requests to the Logic tier and displays results of the requested action. It interacts only with the Logic tier.

The web application in the Logic tier is organised according to Model-View-Controller (MVC) architecture, named by its components. MVC is widely used in Web application development as it applies separation of concerns principle (SoC) and improves reusability. MDVPN Web application consists of:

- The Model component – EJB<sup>12</sup>/JPA<sup>13</sup> (Hibernate ORM<sup>14</sup>). The Model component manages data that is retrieved or stored according to Controller commands and displayed in View.
- The Controller component – JavaServer Faces<sup>15</sup> (JSF). The Controller component sends commands to Model regarding needed data modifications. Controller also sends updates to View. There can be several Controllers.
- The View component – PrimeFaces<sup>16</sup> and JSF. The View component generates output in a user-friendly manner according to changes in the Model. There can be several Views.

The user interaction with the MDVPN SI Web application is shown in Figure B.2.

---

<sup>9</sup> <http://www.postgresql.org>

<sup>10</sup> <http://wildfly.org/>

<sup>11</sup> <https://httpd.apache.org>

<sup>12</sup> <https://jcp.org/en/jsr/detail?id=318>

<sup>13</sup> [http://download.oracle.com/otndocs/jcp/persistence-2\\_1-fr-eval-spec/index.html](http://download.oracle.com/otndocs/jcp/persistence-2_1-fr-eval-spec/index.html)

<sup>14</sup> <http://hibernate.org/orm/>

<sup>15</sup> <https://jcp.org/aboutJava/communityprocess/final/jsr344/index.html>

<sup>16</sup> <http://www.primefaces.org>

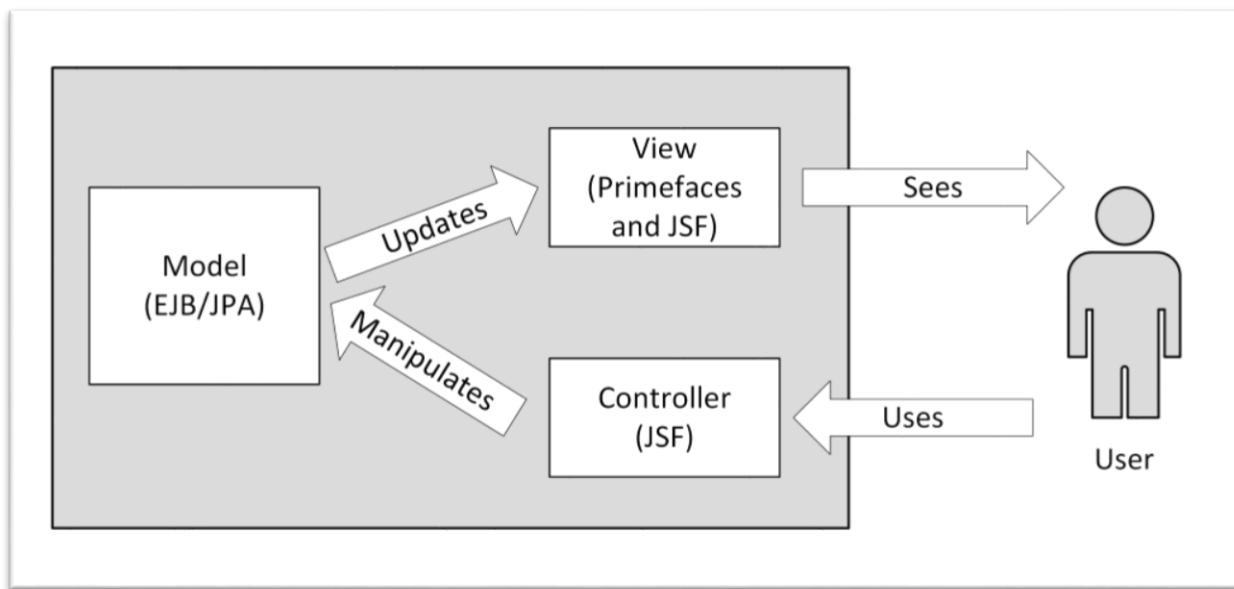


Figure B.2: MDVPN SI Web application – interactions

## B.2 MDVPN Service Inventory – functionalities

A short summary and overview of the functionality offered by MDVPN service inventory (MDVPN SI)<sup>17</sup> is given in this section. The welcome page of the tool is shown in Figure B.3

<sup>17</sup> The tool is available on the following page: <http://mdvpn-inventory.galab.geant.net/mdvpn-service-inventory/index.jsf>

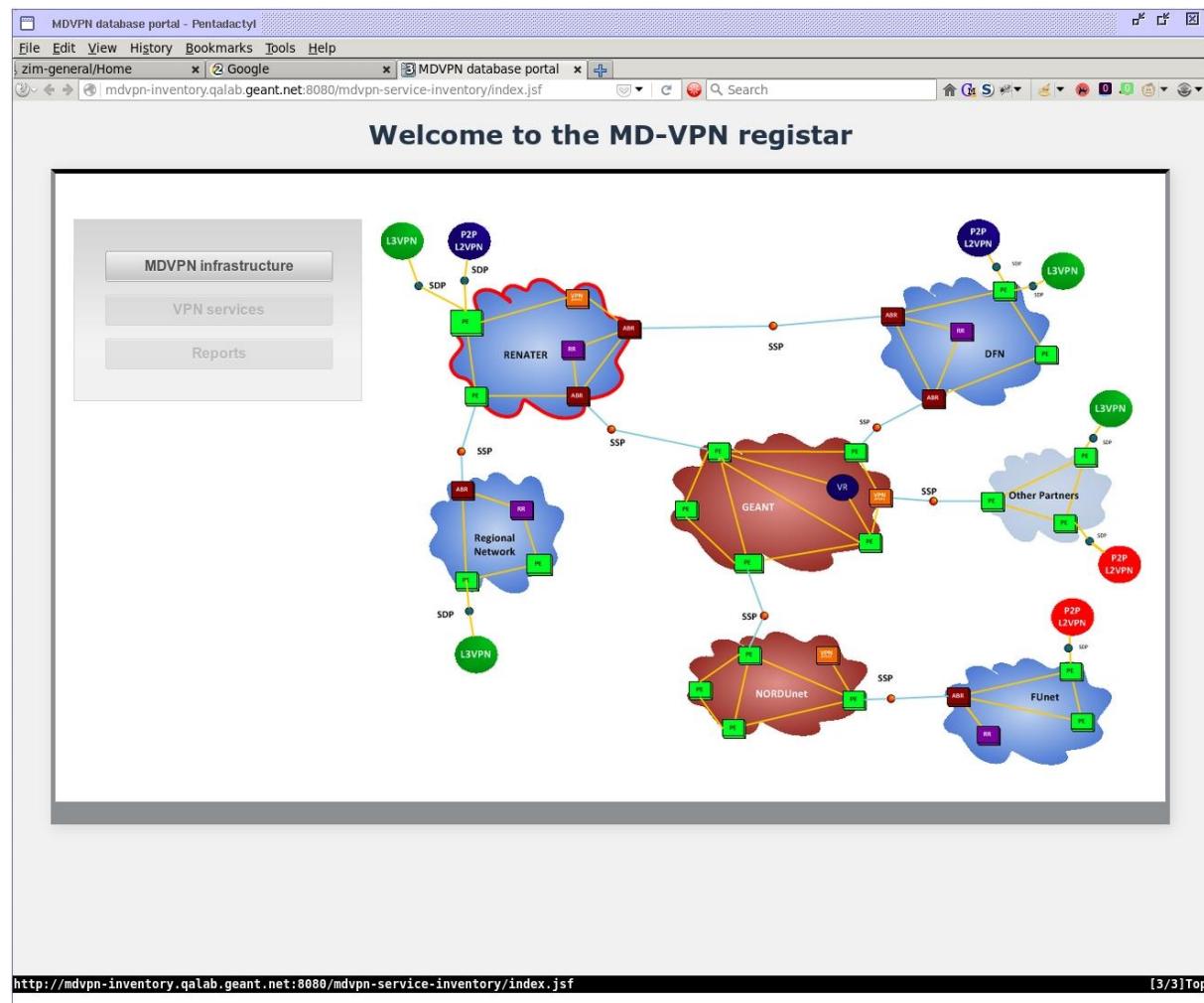


Figure B.3: MDVPN SI welcome page

MDVPN SI is designed to support all necessary types of multi-domain data concerning the configuration of MDVPN instances to be edited and managed, including the possibility of entering new data in a workflow fashion. This comprises data concerning MDVPN-related network services providers along with relevant persons and all involved equipment. It also contains all data about the actual VPN service instances realised with this equipment, along with their related projects, service stitching points, sites, and relevant people.

For both types of a specific workflow, to add a new network service provider a new VPN service instance is available. To give an impression about these workflows, the following three figures show steps to add a new network service provider are shown.

MDVPN database portal - Pentadactyl

File Edit View History Bookmarks Tools Help

zim-general/Home x Google x MDVPN database portal x +

mdvpn-inventory.qalab.geant.net:8080/mdvpn-service-inventory/mdvpn-infrastructure [3/3]

**NSP registration - step 1**

**NSP data**

NSP name:	DFN
NSP country:	Germany
NSP web page:	http://www.dfn.de
Nsp type:	NREN
Parent nsp:	GEANT
AS number:	12345

**Autonomous systems**

NSP is not registered yet.

<http://mdvpn-inventory.qalab.geant.net:8080/mdvpn-service-inventory/mdvpn-infrastructure-nsp.jsf> [-] [3/3] Top

-- INSERT --

Figure B.4: New Network Service Provider Workflow: Step 1

MDVPN database portal - Pentadactyl

File Edit View History Bookmarks Tools Help

zim-general/Home x Google x MDVPN database portal x +

mdvpn-inventory.qalab.geant.net:8080/mdvpn-service-inventory/mdvpn-infrastructure [3/3]

**registration - step 2**

**Nsp contacts**

**Person Registration**

**New equipment in DFN**

**Persons**

No registered persons.

<http://mdvpn-inventory.qalab.geant.net:8080/mdvpn-service-inventory/mdvpn-infrastructure-add-nsp-2.jsf?cid=1> [-] [3/3] Bot

Figure B.5: New Network Service Provider Workflow: Step 2



Figure B.6: New Network Service Provider Workflow: Step 3

The following equipment-related types are supported by the tool: AS border routers, service stitching points, (VPN) router reflectors (VRR and RR), peerings with VRR, PEs, VRR clients and VRR proxies.

Here the final summary of the new network service provider is given as an overview, still with the possibility to further edits:

MDVPN database portal - Pentadactyl

File Edit View History Bookmarks Tools Help

zim-general/Home x Google x MDVPN database portal x +

mdvpn-inventory.qalab.geant.net:8080/mdvpn-service-inventory/mdvpn-inl

Search

**DFN - summary**

NSP name: DFN  
NSP country: Germany  
NSP URL: http://www.dfn.de  
NSP type: NREN  
NSP parent: GEANT  
AS: 123435

**Persons**

ID	Person full name	Person name	Person fax, tel.	Person mob, tel.	Person email	Person roles	
71	Joe	NoName	+412345678	+1234567	test@test.com	 	

**DFN - ASBR list**

ID	ASBR name		ASBR loopback IP	
264	AS border router 1		12.12.12.12	
265	AS border router 2		12.12.12.15	

**Half SSP list**

ID	Equipment name	Couple equipment name	IP peer name	Monitoring circuit name	SSP status	Creation date	
30	AS border router 1	bud.mxt.geant.net		mon circuit name 1	UNACTIVE	2015-11-06	

**DFN - VRR list**

ID	VRR name		VRR loopback IP	
267	VRR 1		12.12.12.21	

**DFN - RR list**

ID	RR name		RR loopback IP	
269	RR 1		12.12.12.31	

**Peer with VRR list**

ID	VRR name	RR name	Monitoring circuit name	Peer with VRR community	VRR learning state	Creation date	
30	lubbjanix.geant.net	RR 1	mon circuit 2	test community	UNACTIVE	11/01/15	

**DFN - PE list**

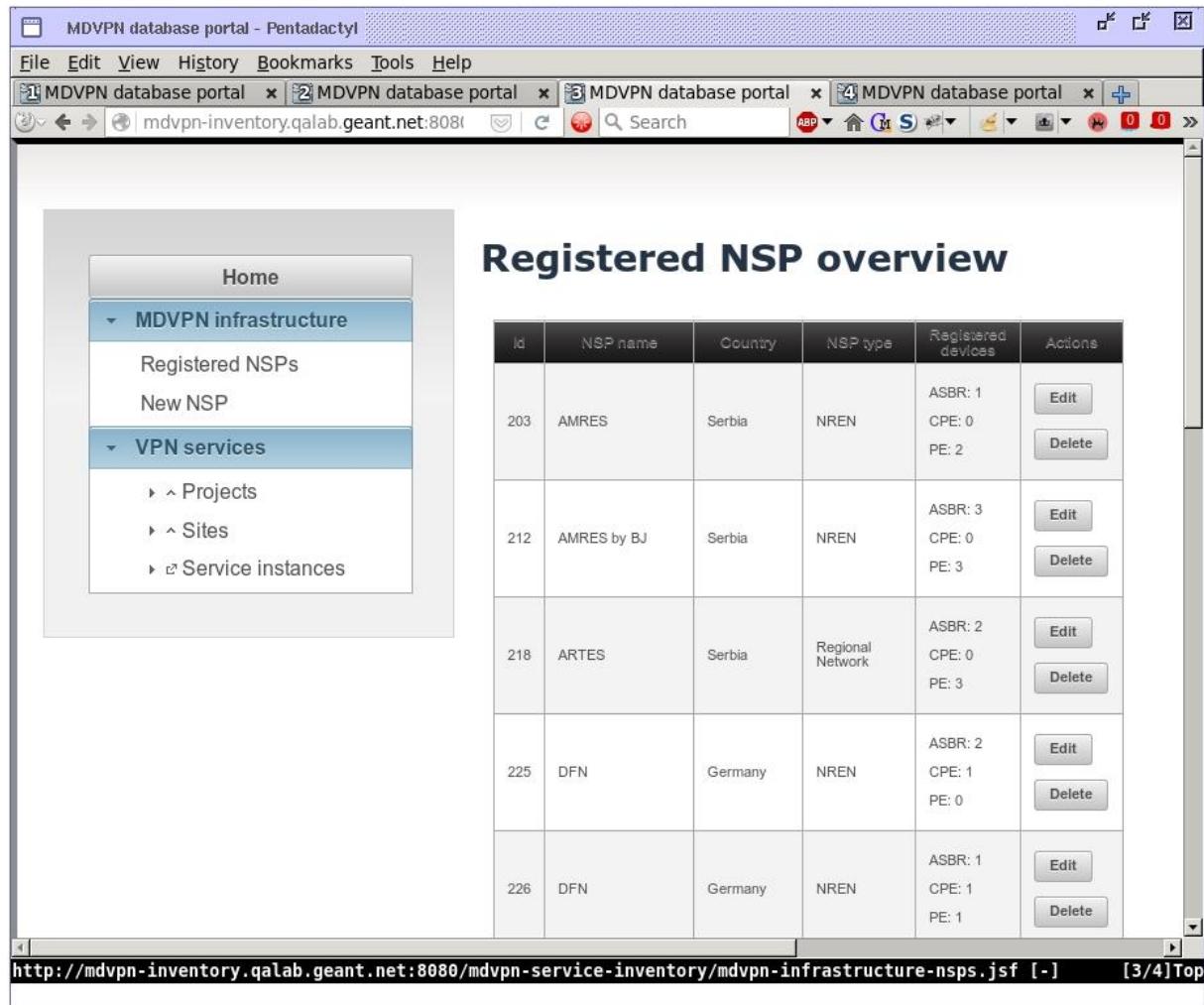
No registered PEs.

**DFN - others list**

Figure B.7: New Network Service Provider Workflow: Summary

**Deliverable D8.2  
Production Services Common Components  
and Processes  
Document Code: GN4-1-16-8D4C0**

As an example, an overview list page is shown which allows further edits to existing network service providers.



Id	NSP name	Country	NSP type	Registered devices	Actions
203	AMRES	Serbia	NREN	ASBR: 1 CPE: 0 PE: 2	<a href="#">Edit</a> <a href="#">Delete</a>
212	AMRES by BJ	Serbia	NREN	ASBR: 3 CPE: 0 PE: 3	<a href="#">Edit</a> <a href="#">Delete</a>
218	ARTES	Serbia	Regional Network	ASBR: 2 CPE: 0 PE: 3	<a href="#">Edit</a> <a href="#">Delete</a>
225	DFN	Germany	NREN	ASBR: 2 CPE: 1 PE: 0	<a href="#">Edit</a> <a href="#">Delete</a>
226	DFN	Germany	NREN	ASBR: 1 CPE: 1 PE: 1	<a href="#">Edit</a> <a href="#">Delete</a>

Figure B.8: Network Service Provider Overview

Similarly, the tools support a workflow for adding a new VPN service instance. The following three figures show two steps of this workflow and a final summary page.

MDVPN database portal - Pentadactyl

File Edit View History Bookmarks Tools Help

MDVPN database po... MDVPN database po... MDVPN database po... MDVPN database po...

mdvpn-inventory.qalab.geant.net:8080

Goto VPN data Goto SDP CE PE links Cancel

## VPN registration - step 1

New VPN Instance

VPN instance name:	Test VPN instance
VpnInstance type:	L3VPN
Initiator NSP:	DFN
Project:	Test Project
+ Add	
VPN instance status:	OPERATIONNAL
VPN instance route target:	1.1.1.1
Request date:	1/2/16
Foreseen deployment date:	1/15/16
Decommission date:	1/31/16

Save and Stay >> Next

<http://mdvpn-inventory.qalab.geant.net:8080/mdvpn-service-inventory/mdvpn-service-instances>. [3/4] Top

Figure B.9: New VPN Service Instance Workflow: Step 1

MDVPN database portal - Pentadactyl

File Edit View History Bookmarks Tools Help

MDVPN database por... MDVPN database por... MDVPN database por... MDVPN database por... +

mdvpn-inventory.qalab.geant.net:8080 Search ABP

**VPN registration - step 2**

New SDP CE PE link for Test VPN instance

Site:	Select Site
SDP PE router:	PE-2.amres.ac.rs
SDP status:	UP
SDP CE PE MTU:	1400
Foreseen deployment date:	1/11/16
Foreseen complete deployment date:	1/31/16

**SDP CE PE Links**

No registered SDP CE PE links.

http://mdvpn-inventory.qalab.geant.net:8080/mdvpn-service-inventory/mdvpn-service-instances-a[3/4]Top

Figure B.10: New VPN Service Instance Workflow: Step 2

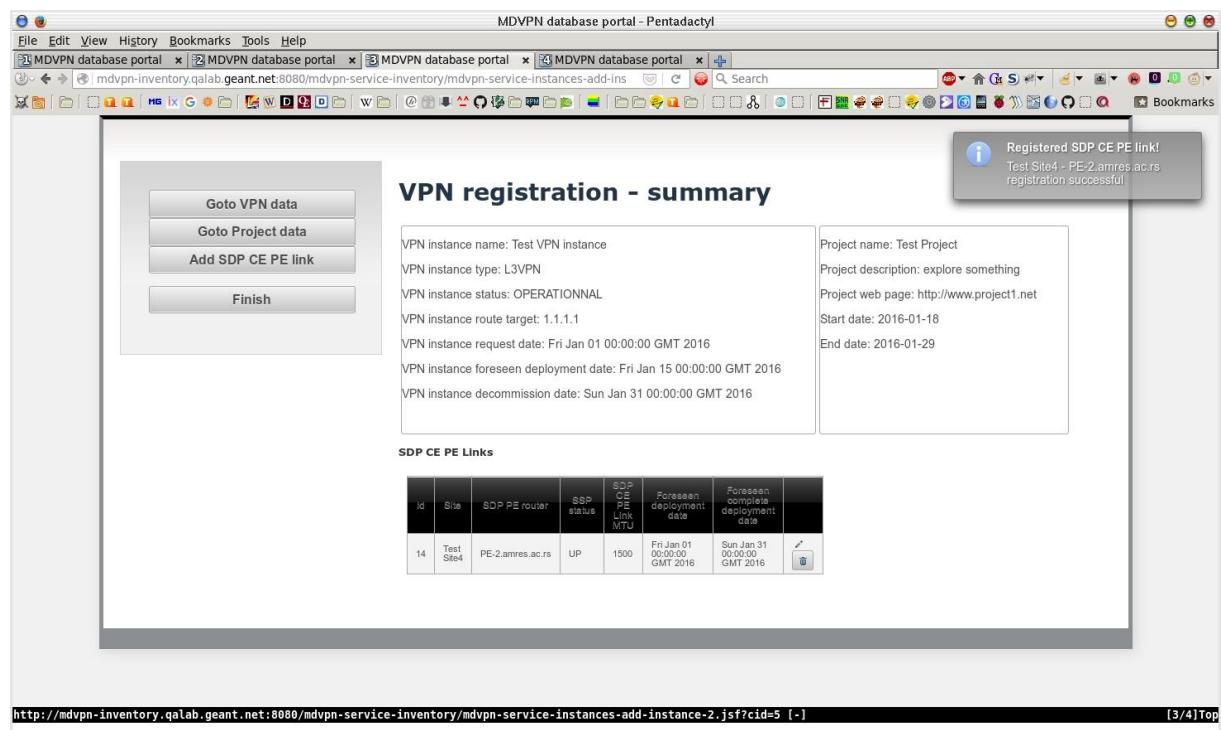
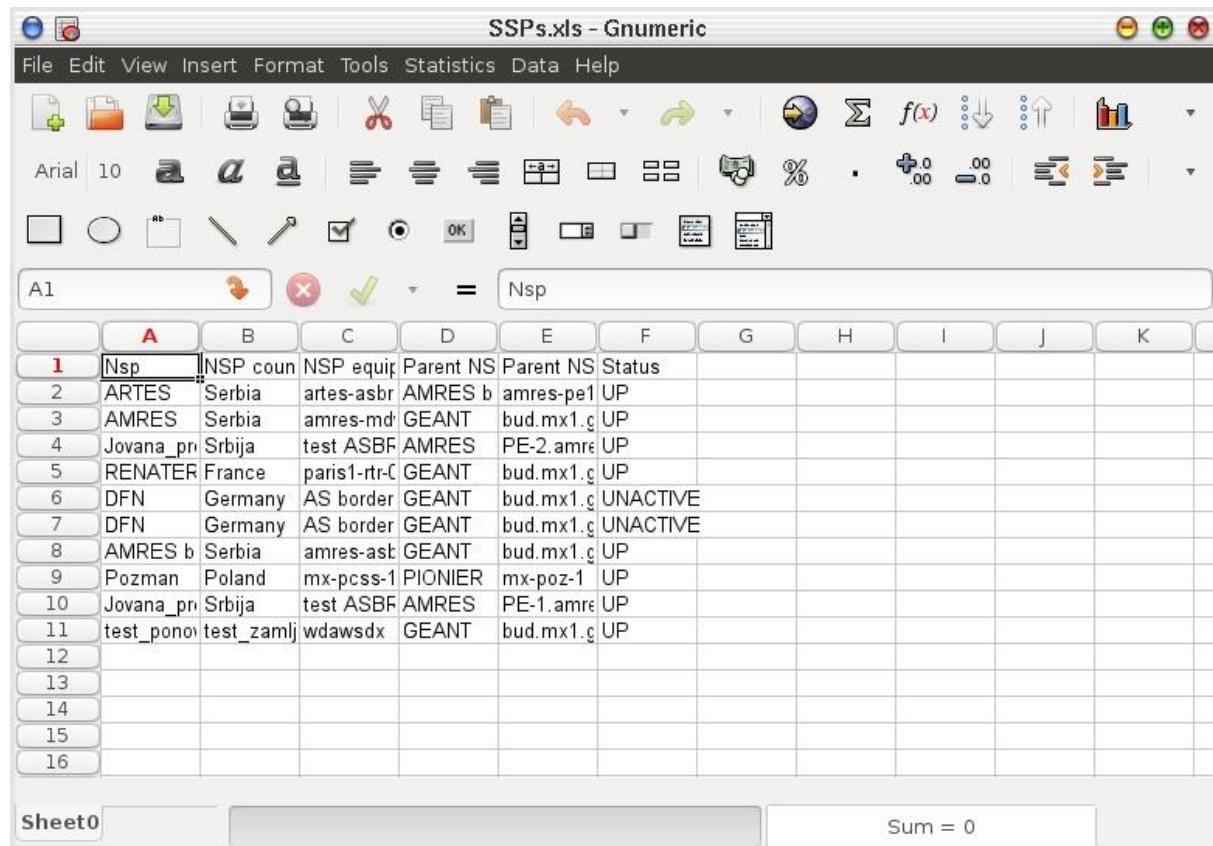


Figure B.11: New VPN Service Instance Workflow: Summary

Further editing of existing VPN service instances is also possible, including the later editing of projects along with relevant people, as well as the editing, adding or removing of associated service stitching points along with their sites.

The tool supports not only the editing and management of MDVPN-related configuration data internally, but can also generate filtered reports for various aspects, like service stitching points, VRR/RR and PE. The reports may be exported as Excel, PDF, CVS or XML files. Figure B.12, Figure B.13 and Figure B.14 show examples of such exported reports:

SSPs.xls - Gnumeric



The screenshot shows a Gnumeric spreadsheet titled "SSPs.xls - Gnumeric". The menu bar includes File, Edit, View, Insert, Format, Tools, Statistics, Data, and Help. The toolbar contains various icons for file operations, cell selection, and mathematical functions like sum and product.

The spreadsheet has a header row with columns A through K labeled: Nsp, NSP coun, NSP equip, Parent NS, Parent NS Status. The data starts at row 1 and continues through row 16. Row 1 is bolded and serves as the header. The data includes entries for ARTES, AMRES, RENATER, DFN, AMRES b, Pozman, Jovana\_pn, and test\_pono. The "Parent NS Status" column shows values like "amres-pe1 UP", "bud.mx1.g UP", and "UNACTIVE".

	A	B	C	D	E	F	G	H	I	J	K
1	Nsp	NSP coun	NSP equip	Parent NS	Parent NS	Status					
2	ARTES	Serbia	artes-asbr	AMRES b	amres-pe1	UP					
3	AMRES	Serbia	amres-md	GEANT	bud.mx1.g	UP					
4	Jovana_pn	Srbija	test ASBF	AMRES	PE-2.amre	UP					
5	RENATER	France	paris1-rtr-C	GEANT	bud.mx1.g	UP					
6	DFN	Germany	AS border	GEANT	bud.mx1.g	UNACTIVE					
7	DFN	Germany	AS border	GEANT	bud.mx1.g	UNACTIVE					
8	AMRES b	Serbia	amres-ast	GEANT	bud.mx1.g	UP					
9	Pozman	Poland	mx-pcss-1	PIONIER	mx-poz-1	UP					
10	Jovana_pn	Srbija	test ASBF	AMRES	PE-1.amre	UP					
11	test_pono	test_zamlj	wdawsdx	GEANT	bud.mx1.g	UP					
12											
13											
14											
15											
16											

Sheet0 Sum = 0

Figure B.12: SSP Report Export as an Excel File

Xpdf: /home/mathix/current/download/SSPs.pdf

Nsp	NSP country	NSP equipment name	Parent NSP	Parent NSP equipment name	Status
ARTES	Serbia	artes-asbr1.artes.com	AMRES by BJ	amres-pe1.amres.ac.rs	UP
AMRES	Serbia	amres-mdvpn-asbr.amres.ac.rs	GEANT	bud.mx1.geant.net	UP
Jovana_prob_a_test	Srbija	test ASBR	AMRES	PE-2.amres.ac.rs	UP
RENATER	France	paris1-rtr-001	GEANT	bud.mx1.geant.net	UP
DFN	Germany	AS border router 1	GEANT	bud.mx1.geant.net	UNACTIVE
DFN	Germany	AS border router 1	GEANT	bud.mx1.geant.net	UNACTIVE
AMRES by BJ	Serbia	amres-asbr1.amres.ac.rs	GEANT	bud.mx1.geant.net	UP
Pozman	Poland	mx-pcess-1	PIONIER	mx-poz-1	UP
Jovana_prob_a_test	Srbija	test ASBR	AMRES	PE-1.amres.ac.rs	UP
test_ponovo	test_zamlja	wdawsdx	GEANT	bud.mx1.geant.net	UP

Figure B.13: SSP Report Export as a PDF File

Figure B.14: SSP Report Export as an XML File

## Appendix C Example of MDVPN Service Configuration

This Appendix outlines the technical details of integrating configuration automation in the MDVPN service using a solution based on Ansible, an open source tool. As outlined in the following sections, Ansible is used to provision a MVDPN service on a Juniper logical router, but the technical solution applies to all MDVPN services and is vendor agnostic.

It is out of scope for this task to build configuration templates that cover all MDVPN services, but an example of provisioning a L3VPN is given.

### C.1 Building a MDVPN service (L3VPN) with Ansible

Ansible is organised in pieces of code, named playbooks that are written in YAML<sup>18</sup> format. For example by running the following mdvpn.provision.yml file:

```
#####
#
# This play creates the configuration for a new MDVPN service #
# by calling the mdvpn role and the relative templates #
#
#####
---  
- name: Create configuration for new MDVPN service  
  hosts: mdvpn  
  connection: local  
  gather_facts: no  
  
  roles:  
    - mdvpn
```

Ansible calls on a predefined role that is here named mdvpn. This role gathers all mdvpn service-specific variables, invokes a configuration template and builds the configuration, storing it in a local file:

---

<sup>18</sup> YAML is a data serialisation language: <http://yaml.org/>

```
$ ansible-playbook mdvpn.provision.yml

PLAY [Create configuration for new MDVPN service] ****

TASK: [mdvpn | Building new MDVPN service configuration] ****
changed: [kolettir2.grnet.gr]

PLAY RECAP ****
kolettir2.grnet.gr : ok=1 changed=1 unreachable=0 failed=0
```

The generated output is stored in a predefined folder/file as outlined in another YAML file:

```
build_dir: ./generated_configs/{{ inventory_hostname }}/tmp
junos_conf: ./generated_configs/{{ inventory_hostname }}/junos.conf
```

and is ready to be committed to the device.

### C.1.1 Building the variables file

To produce the configuration file, Ansible uses necessary variables as input to a predefined template that is presented later in this Appendix.

In this example (building the configuration for a L3VPN), the following data is stored in the `host_vars.yaml` file:

```
service:
  hostname: mdvpn
  routing_instance_name: edusafe
  customer_interface: ae2
  loopback: 62.217.100.21
  mdvpn_service_id: 111
  route_target: 9112:3066
  nren_asn: 5408
  customer_connected_ip: 10.0.0.2
  nren_connected_ip: 10.0.0.1
  connected_subnet: /30
  assigned_v6: False
  nren_connected_ipv6: 2001:ftou:kai:bgainw
  connected_subnetv6: /126
  customer_asn: 65000
  customer_vlan: 578
  customer_subnet: 10.0.0.0/24
```

```

bgp_policy_vrrs_out: mdvpn-vrrs-out
subif_description: .HOSTED-EDUSAFE
bgp_with_customer: False
  - static_with_customer: True

```

The data for the host\_vars.yaml file was populated manually according to the information exchanged in emails during the new MDVPN service request. In our recommendation for configuration automation, data input is linked with the MDVPN SI so that the host\_vars file is populated automatically from the Service Inventory system.

### C.1.2 Using service templates

The next step is building the configuration templates. Ansible has been instructed to use a JINJA file as a configuration template:

```

---
# by referencing the mdvpn role in a playbook
# this task is executed causing Ansible to generate
# configuration for a new mdvpn service
- name: Building new MDVPN service configuration
  template: >
    src=mdvpn.conf.j2
  dest={{ build_dir }}/mdvpn.conf

```

The mdvpn.conf.j2 file, is shown below.

This stage is indicative of the timescale associated with integrating configuration automation to the service, since the MDVPN team is assigned to build configuration templates that cover all the use cases and vendor equipment used to deliver the MDVPN services.

```

logical-systems {
  # this configuration builds a new L3VPN service based on a already configured
  logical system #
  {{ service.hostname }} {
    interfaces {
      # first build the interface configuration #
      {{ service.customer_interface }} {
        unit {{ service.customer_vlan }} {
          description "[{{ service.subif_description }}]";
          vlan-id {{ service.customer_vlan }};
          family inet {
            address {{ service.nren_connected_ip }}{{ service.connected_subnet }};
          }
        }
      }
    }
  }
}

```

```
% if service.assigned_v6 %}
family inet6 {
address {{ service.nren_connected_ipv6 }}{{ service.connected_subnetv6 }};
}

{%
endif %}
}

}

policy-options {
# add the service assigned subnet to the BGP export policy towards the VRR#
policy-statement {{ service.bgp_policy_vrrs_out }} {
term 1 {
from {
route-filter {{ service.customer_subnet }} exact;
}
}
}
}

routing-instances {
# configure the routing instance for a new L3VPN service #
{{ service.routing_instance_name }} {
instance-type vrf;
interface {{ service.customer_interface }}.{{ service.customer_vlan }};
route-distinguisher {{ service.loopback }}:{{ service.mdvpn_service_id }};
vrf-target target:{{ service.route_target }};
vrf-table-label;
# configure static routing in case no dynamic protocol is in use #
{% if service.static_with_customer %}
routing-options {
static {
route {{ service.customer_subnet }} next-hop {{ service.customer_connected_ip }};
}
}
{% endif %}
# or else configure eBGP routing with L3VPN customer #
{% if service.bgp_with_customer %}
protocols {
bgp {
group {{ service.routing_instance_name }} {
type external;
local-as {{ service.nren_asn }};
}
```

```
neighbor {{ service.customer_connected_ip }} {
  description {{ service.routing_instance_name }};
  local-address {{ service.nren_connected_ip }};
  peer-as {{ service.customer_asn }};
}

}

}

}

}

{ % endif %

}

}

}

}

} 
```

Ansible logs indicate the term in which each task is executed until the construction of the final configuration file:

```
$ tail ansible.log

2015-11-16 10:55:07,213 p=5239 u=mmamalis | PLAY [Create configuration for new
MDVPN service] ****
2015-11-16 10:55:07,215 p=5239 u=mmamalis | TASK: [mdvpn | Building new MDVPN
service configuration] ****
2015-11-16 10:55:07,294 p=5239 u=mmamalis | changed: [kolettir2.grnet.gr]
2015-11-16 10:55:07,295 p=5239 u=mmamalis | PLAY [Apply configuration]
*****
2015-11-16 10:55:07,295 p=5239 u=mmamalis | TASK: [assembling part configuration
files and merging to one file] ****
2015-11-16 10:55:07,352 p=5239 u=mmamalis | ok: [kolettir2.grnet.gr]
2015-11-16 10:55:21,207 p=5239 u=mmamalis | PLAY RECAP
*****
2015-11-16 10:55:21,207 p=5239 u=mmamalis | kolettir2.grnet.gr : ok=5 changed=4
unreachable=0 failed=0
```

The configuration is ready for inspection by the user via the MDVPN SI portal and for final implementation to the PE device.1

## References

- [CSI] ITIL Continual Service Improvement, 2011 edition, TSO, London 2011.
- [eduPKI] <https://www.edupki.org>
- [eduPKIreg] eduPKI Services registration process,  
<https://www.edupki.org/fileadmin/Documents/eduPKI-PMA-GÉANT-services-registration-process-1.0.1.pdf>
- [NN] Network namespaces,  
<https://lwn.net/Articles/219794/>
- [perfSONAR] <http://www.perfsonar.net>
- [PLMportal] The PLM business process,  
<http://plm.geant.net>
- [PLMprocess] PLM process for GN4,  
<https://services.GÉANT.net/plm/Pages/GN4-PLM-page-1.aspx>
- [pSCD] perfSONAR configuration,  
[http://docs.perfsonar.net/config\\_files.html](http://docs.perfsonar.net/config_files.html)
- [pSCMC] The perfSONAR MeshConfig,  
[http://docs.perfsonar.net/config\\_mesh.html?highlight=mesh%20configuration](http://docs.perfsonar.net/config_mesh.html?highlight=mesh%20configuration)
- [pSRTS] perfSONAR Regular Testing Configuration,  
[http://docs.perfsonar.net/config\\_regular\\_testing.html?highlight=regular\\_testing](http://docs.perfsonar.net/config_regular_testing.html?highlight=regular_testing)

## Glossary

<b>CSF</b>	Critical Success Factor
<b>CSI</b>	Continual Service Improvement
<b>FaaS</b>	Federation-as-a-Service
<b>GUI</b>	Graphical User Interface
<b>ITIL</b>	Information Technology Infrastructure Library. A set of practices for IT Service Management (ITSM) that focuses on aligning IT services with the needs of business
<b>KPI</b>	Key Performance Indicator
<b>MDVPN</b>	Multi-Domain VPN
<b>MVC</b>	Model-View-Controller architecture
<b>NOC</b>	Network Operations Centre
<b>NREN</b>	National Research and Education Network
<b>OLA</b>	Operational Level Agreement
<b>OWAMP</b>	One-Way Active Measurement Protocol
<b>perfSONAR</b>	Performance focused Service Oriented Network monitoring Architecture, an open source toolkit for running network tests across multiple domains
<b>PLM</b>	Product Lifecycle Management
<b>PLR</b>	Packet Loss Rate
<b>PM</b>	Person Month
<b>pS MP</b>	perfSONAR measurement point
<b>pSCD</b>	perfSONAR Configuration Daemon
<b>pSRTS</b>	perfSONAR Regular Testing Service
<b>psSMC</b>	perfSONAR Mesh Configuration
<b>RFC</b>	Request For Change
<b>SLA</b>	Service Level Agreement
<b>SQM</b>	Software Quality Management
<b>SSO</b>	Single Sign On
<b>UI</b>	User Interface
<b>VPN</b>	Virtual Private Network