29-04-16

Deliverable D7.2
Performance Monitoring and Verification Framework

Deliverable D7.2

Contractual Date:	30-04-2016
Actual Date:	29-04-2016
Grant Agreement No.:	691567
Activity:	7/SA3
Task Item:	Task 3
Nature of Deliverable:	R (Report)
Dissemination Level:	PU (Public)
Lead Partner:	SWITCH
Document Code:	GN4-1-16-4A66F
Authors:	K. Baumann (SWITCH), A. Delvaux (PSNC), A. Oslebo (UNINETT), A. Wilson (HEAnet), B. Mortensen (NORDUnet), H. Yu (NORDUnet/DTU), J. Healy (DCU), J. Hertzberg (NORDUnet), J. Melnikov (CESNET), K. Stamos (GRNET), N. Nikovic (AMRES), N. Kanakis(GRNET), R. Lapacz (PSNC), T. Appel (DFN), T. Kulkarni (GEANT), V. Kokkinos (GRNET)

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
The aim of this document is to introduce the basic elements of Performance Monitoring &Verification (PM&V) framework through five essential use cases. An inductive, bottom-up approach using existing PM&V tools, such as perfSONAR(pS), Circuit Monitoring (CMon) and Service Quality Management (SQM) have been chosen to define generic architecture elements.

Table of Contents

Table of Figures

Table of Tables

Executive Summary

Increasing requirements to a Performance Monitoring and Verification (PM&V) framework caused by virtualisation of physical network layers, place an "urgent" need on research, technologies and standards. Network performance monitoring and verification is identified as one of the essential process in network operations since the formal network management framework FCAPS [FCAPS]. Consequently, providing a network service without proper performance verification is unthinkable today.

Within the GÉANT network, there is a broad need, not only for a global monitoring and measurement framework, but also for a PM&V. The need stems from a large number of new GÉANT-based services, such as virtualised services and emerging transport services, where current performance monitoring tools and verification processes no longer apply.

The following discussion relates to an PM&V over wired, wireless and virtualised computer (campus) networks. As in the Wireless Crowd Source Performance Monitoring and Verification document (M7.1), the PM&V is, or can be, particularly challenging, because a unique tool does not exist that covers all aspects on performance monitoring and verification [M7.1]. Therefore, the concept of performance verification becomes important for two reasons:

- Elaboration of problem solving procedures to ensure an appropriate response when faced with performance issues.
- Strategically, by defining short-, medium- and long-term goals/update/improvements of wired/wireless network topologies.

Therefore, the aim of this document is to introduce the basic elements of a PM&V framework through five essential use cases. An inductive, bottom-up approach using existing PM&V tools, such as perfSONAR(pS), Circuit Monitoring (CMon) and Service Quality Management (SQM) have been chosen to define generic elements of PM&V framework.

It is important to note that wireless, crowd-sourced performance verification and measurement is dependent on valid data-sets collected from a huge number of end-users working on NRENs' campus wireless networks. When trying to understand the end-user's point of view, a dialog about privacy, collecting end-user data and transparency must be initiated. In July 2015, a discussion at the eduPERT monthly call came up with a number of questions, from transparency, tracking and end-user behaviour on campus, to the authorised entities having access to the RADIUS accounting, and the authN/Z log files [eduPERT]. An extended policy discussion on privacy and end-user data is essential to inform future work and to provide a productive-ready service for the NRENs and the GÉANT community.

Examples of end-user transparency can be found in the Milestone M7.1 Document, Section 3.5 [M7.1].

The deliverable is divided into four main parts, and, as mentioned above, Section 2 includes PM&V proposals on essential use case scenarios: eduSAFE as a lightweight PM&V, and on an extended MD-VPN, CMon on the GÉANT Testbed Service (GTS), CMon on Alien Wavelength Service (AWS), SQM on Multi-Domain VPN (MD-VPN) and Wireless Crowd Source PM&V, all as designed and implemented. A qualitative assessment of the PM&V solutions against six technology agnostic questions in Section 3 allows identification of common software architecture elements (functionality) of used tools, described as pS-SQM and CMon-SQM (Section 4). Section 5 concludes the Deliverable, with recommendations extracted from Section 3 and 4 as a PM&V stack, a generic PM&V architecture framework with a future scope (steps) based on NIF-P15_227 [NIF-P15_227]. It is also important to note the collaborative cross-Activity work undertaken, as the implementation forms of the PM&V proposals were passed between eduSAFE devOps team, SA2,T3 (GTS) and JRA1,T2 (AWS) and the WiFiMon team (SA3,T3).

All used Open Source Software for deployments of the PM&V solution proposals are listed in the Appendix B.

1 Introduction

Performance Monitoring and Verification (PM&V) is becoming increasingly important for the (N)REN community. Computing network evolutions and network virtualisation brings new challenges, as network abstraction increases network complexity (e.g. by virtualised network functions) and also network management systems. When it comes to network performance issues, new ways of automatic performance control, verification, and provisioning of measurement/monitoring services to the network providers and the end-users, such as researchers and academic ICTs, are needed.

This has created a need for better understanding of the behaviour of network performance issues on wired/wireless/virtualised networks, their localisation and verification.

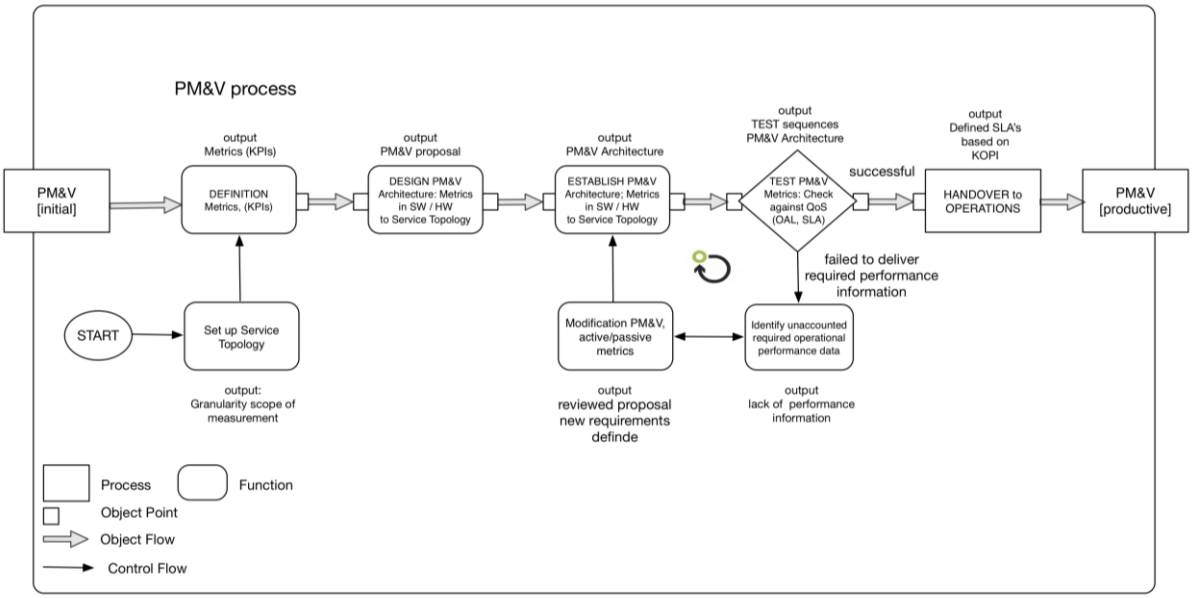

Figure 1.1: Generic PM&V process

Today's wireless/wired campus networks allow academic users and users from the private sector to manage their daily workload on laptops and powerful smartphones. Thus tremendous increase of (business relevant) data volumes implies the need to define/introduce Quality of Services (QoS) or to update Service Level and Operational Level Agreements (OLAs) by network providers.

Discussions with (N)RENs, eduPERT members, universities' NOCs and conference organisers, helped to define a generic PM&V process, from the definition of the Key Operational Performance Indicators (KOPIs) to the verification process depicted in Figure 1.1. This process can be implemented manually or can be incorporated into automated verification procedures of the (modified) KOPIs, see in Figure 1.1. This process also reflects the working procedure to figure out the elements of a generic PM&V framework (see Section 5).

The four main functions of the PM&V process can be described as follows:

1. **Define Metrics** – INPUT: Granularity of measurement scope / OUTPUT: Key Operational Performance Indicators (KOPIs), Metrics – The granularity of the measurement scope is the input for defining metrics to the service topology. The KOPIs will be defined according the service topology. A minimal set-up of KOPIs will be used in SA3, T3. Metrics will be grouped into active[1] and passive[2] monitoring:
    - Active Monitoring: Latency, jitter, delays between the service relevant elements and bandwidth, the expected/available bandwidth between service relevant elements.
    - Passive Monitoring: Availability that answers the question "Are all service-relevant elements reachable?", traffic volume that answers "What will be the current/expected traffic volume?", and load, which indicates the load of the active service elements.

    The number of KOPIs can vary from use case to use case and priorities (see Section 2). The definition of metrics is essential for collecting appropriate measurement data, but also as a basis for an agreement between Service Owners (SO), the Operations (Ops), and Service Consumers (SC).

2. **DESIGN PM&V Architecture** – INPUT: Metrics (KOPIs) / OUTPUT: PM&V proposal.

    Here the PM&V architecture will be designed. How performance is measured (see Section 2), means using the metrics defined in the Step 1. The output, the PM&V proposal, describes the design of the measurement architecture building blocks, collecting needed measurement data as negotiated with the Service Owners (SO), Operations (Ops) and Service Consumers (SC).

[1] **Active Network Monitoring:** Active Network Monitoring is used to simulate certain conditions in the network and check parameters of data traffic. In order to verify how the network would behave it is a common approach to generate traffic (applying a hardware tool like Spirent generator or a software solution like iperf) and observe metrics, for example the bandwidth, loss or delay. This kind of monitoring is very useful to tune network nodes (parameters related to timing, buffers and protocols) and find possible issues in configurations but at the same time it needs to run carefully not to disrupt the operational data traffic in the network being verified.

[2] **Passive Network Monitoring:** Passive network monitoring focuses on observable study rather than an experiment (see active monitoring). It collects the information about users' traffic in the operational network. A common approach is to use the SNMP protocol to fetch data from network devices (switches, routers, etc.) and applications and then visualise them in a GUI. Network monitoring platforms deployed in Network Operation Centres (NOC) also set up thresholds for collected data values, which can trigger some defined actions (alarms, notifications, executions of scripts, etc.). Passive monitoring is the simplest way to get the information about the current state of the network as well as the view from the past (usually data are collected in dedicated databases). It is worth to mention that there are also other ways to collect network traffic information, more sophisticated that SNMP. One of them is NetFlow.

Essential indicators can be Measurement Points (MPs) and Archives (MA), scalability and robots for testing purposes. This function can be an iterative process with the SO, Ops and SC.

3. **ESTABLISH PM&V Architecture** – INPUT: PM&V proposal / OUTPUT: PM&V Architecture

   Establishing PM&V architecture includes all engineering and deployment tasks based on Step 2, in discussion with the SO, Ops and SCs. The implementation form of the defined metrics (see Step 1) can be pure-SW (e.g. JavaScripts and analysis procedures on web resources) or HW probes (see Section 2). This function can be an iterative process with the SO, Ops and SC.

4. **TEST PM&V metrics and** – INPUT: PM&V architecture / OUTPUT: Test sequences PM&V architecture – This step shows as output sequences, tested and validated PM&V architecture metrics. Test sequences can be implemented manually or automated, and verify the accuracy of the metrics to negotiated KPOIs. Test sequences can be fully automated by robots. This process step respect metrics revision, regarding changes and/or modifications of the service delivery: TEST SUCCESSFUL is the input to HANDOVER process, which is not a part of this document. TEST FAILED needs to identify and verify unaccounted REQUIRED operational performance data, modification of used metric set-up (e.g. extension of KOPIs), and implementation of revised metrics onto the PM&V architecture. This function can be an iterative process with the SO, Ops and SC.

The outcome of PM&V process allows the introduction of an operative performance measurement and verification service. The metric definition set-up on existing measurement/monitoring tools such as perfSONAR, CMon and SQM for network services [perfSONAR], [CMon], [SQM].

The focus of the deliverable continues with a bottom-up approach describing five essential use cases with a future perspective (Section 2), and an assessment of their PM&V solutions (Section 3) on:

- perfSONAR on eduSAFE in a light-weight version and as an extended MD-VPN use case.
- CMon on the GÉANT Testbed Service (GTS), and the Alien Wave Length Service (AWS).
- SQM on MD-VPN, demonstrated as a prototype.
- A Wireless Crowd Source PM&V, a clean slate approach [M7.1].

This expertise makes it possible to figure out generic PM&V architecture/framework elements, which results in the first steps of a PM&V stack (see the New Ideas Form (NIF), P15 227 for further development, as noted in Section 5 [NIF-P15_227]).

All of these use cases and PM&V proposals were implemented and tested. Thus, the main outputs of the deliverable are the PM&V solutions and assessments of implemented proposals (Section 2 and 3) against six technology-agnostic questions addressing: integration, reusability, extensibility, control and data management, deployment scalability and application performance, and standardisation. As a consequence of this assessment, common software architecture elements (functionalities) of used tools with CMon/SQM and perfSONAR/SQM are described for operational, respective research purposes in Section 4.

The generic PM&V architecture/framework depicted in Section 5 is the starting point for elaborating new future ideas/approaches in automation on detecting performance issues and their verification on virtual and physical network layers. Future steps/ideas as introductory steps to a PM&V framework will be described and recommended to the GEANT community.

Regarding details of the Wireless Crowd Source PM&V use case, the Milestone Document M7.1 Wireless Crowd Source PM&V (WCSPMV) exists. There a 'clean-slate' approach with recommendations for the GÉANT audience (the (N)RENs)to introduce and extend the WCSPMV to a service for future work is demonstrated [M7.1].

2 Performance Monitoring & Verification – Use Cases and Solutions

This section describes PM&V proposals for essential use cases using existing measurement/monitoring tools pS, CMon, SQM and the concept Wireless Crowd Source Performance Monitoring and Verification (WCSPMV). The selection of these use cases was discussed within eduPERT, the (N)RENs, and SA3 T3 WiredMon group, with focus on existing tools and future perspectives from the GÉANT Community's point of view of chosen services: eduSAFE, MD-VPN, GTS, AWS and WCSPMV.

For all these use cases, PM&V proposals were elaborated, installed and tested.

A qualitative summary of experience for each use case may be found in Section 3.

2.1 eduSAFE – pS on eduSAFE

The eduSAFE service is a web-portal that helps the end-users (researcher, stuff and students) to connect with each other using VPN, and to work in a secure network environment [eduSAFE]. eduSAFE hides the technical aspects of creating such connection from the end-user.

2.1.1 Overview

Implementation of performance monitoring and verification in the eduSAFE remote access (RA) Virtual Private Network (VPN) is particularly challenging, given the wide range of end devices with varying levels of security policies, different operating systems and applications setup. Such deployment of eduSAFE RA VPN tends to make conventional monitoring approaches (SNMP, Ping, Traceroute, etc.) useless, since the lack of end device unification and administrative control prevents standard tools from being used. Bearing in mind the aforementioned limitations, and considering the strict confidentiality requirements posed by eduSAFE, the proposed monitoring approach must take this security aspect into account.

2.1.2 Use Case Description

In this section, monitoring solutions for **pre-production stage and production stage** are proposed. Pre-production stage encompasses a monitoring approach where end device (remote users) are administratively controlled by eduSAFE administrators, which allows a much greater flexibility when

deploying monitoring tools. Conversely, the production stage assumes that the end devices are under the control of users, whereas eduSAFE does not have the liberty or power to make specific end devices comply with any specific monitoring tools. Consequently, pre-production and production stages should be considered by the eduSAFE team to verify targeted performance prior to production state, and performance monitoring deployment once the eduSAFE RA VPN is put into production state, respectively.

2.1.3 Feasibility Considerations and Choice of Tools

A basic premise in analysing the feasibility of any monitoring approach is that the deployed solution is not too invasive to the service itself. That means that the proposed tools and approach should not impose any limitations to the service features. It is also important to highlight the requirement of avoiding unnecessary burden to the users with additional installation of tools that could potentially enable performance monitoring associated to the end device. This requirement is justified in terms of security and more importantly, absence of users' technical background. The use case, which is going to be used for performance monitoring proposals is depicted in Figure 2.1.

Lack of the possibility to modify end device configuration diminishes flexibility of monitoring and limits the tools that can be used. As mentioned, monitoring flexibility is only affected in the **production stage**, when administrative control of end devices is out of reach of eduSAFE. On the other side, the **pre-production** stage offers complete flexibility in monitoring deployment with consideration of confidentiality, the basic parameter and the starting point of eduSAFE RA VPN service. Therefore, in that stage, tools such as perfSONAR (pS), OWAMP, Ping, Nagios can easily be used with minimal effort to provide a detailed, accurate and complete set of performance measurements that should provide insight to performance perceived by users.

The production stage prevents the aforementioned monitoring tools to be used, as not all tools are compatible with every OS that can be found on end devices. Also, there are inherent problems with security posture[3] of end devices which can often prevent the operation of specific tools, e.g. Ping. For instance, most end users have antivirus and/or anti-spyware software installed on their system. Such software is very restrictive in allowing unsolicited packets to pass. Therefore, in order to accept these measurement packets, firewall and security policies should be modified, which is a challenge to automate, and is even more difficult to delegate that responsibility to the users.

[3] Security posture: The security approach from plan to implementation, the overall security plan.

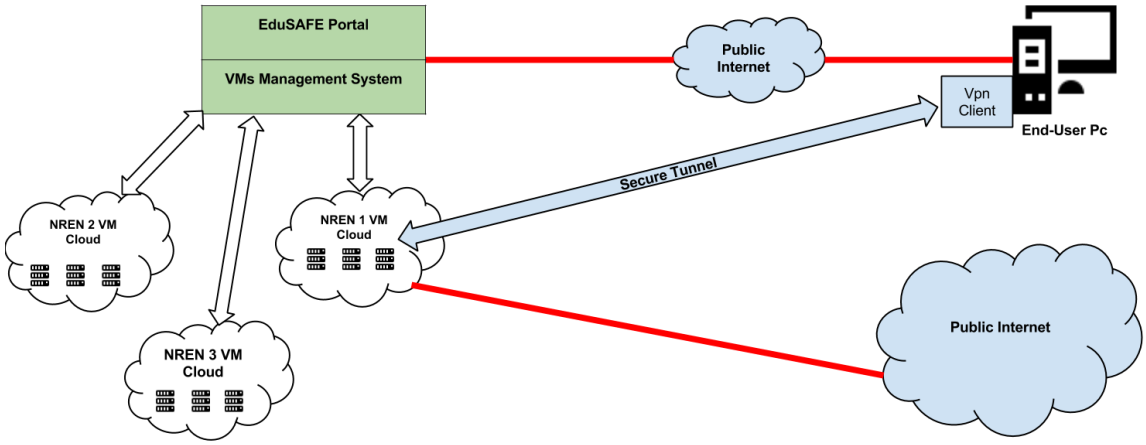

Figure 2.1: eduSAFE use case

Some consideration needs to be taken regarding the use of each measurement tool. The measurements need to be reliable for the monitoring to be useful but also need to avoid to disrupt the service provided to the end users. The following section presents a summary of key points.

2.1.4 Performance Verification Metrics

It is important to list the performance metrics that need to be measured inside the eduSAFE VPN and between some elements of the eduSAFE infrastructure (see Figure 1.1, Section 1). Two types of metrics on passive and active network monitoring (defined on page 4) may be distinguished:

- Metrics for passive monitoring:
  - **Availability**: Are all elements part of a VPN reachable?
  - **Traffic**: What is the current traffic on a VPN link at any point in time?
  - **Load**: What is the load on the eduSAFE active elements? (VPN gateway and web portal)
- Metrics for active monitoring:
  - **Latency**: What is the latency between any two elements of a VPN?
  - **Throughput**: What is the available throughput between two points of a VPN?

Availability refers to the availability of the VPN servers, as well as the availability of the underlying infrastructure. To monitor server availability, CMon Agent can be installed on each VPN server and constantly check the keep-alive messages from the central HQ. To monitor the underlying infrastructure, SNMP can be used to query the interface status. However, there are other, off-the-shelf tools that can be used to achieve the same purpose, e.g. Nagios, Zabbix, Incinga, Cacti etc. Although it is possible to use CMon to monitor availability.

Traffic refers to the received or transmitted number of bytes on a specific virtual interface within a time period. This metric can be monitored using SNMP. CMon Agents need to periodically query the

logical interfaces and calculate the value on the fly as they work. In addition, CMon will need to change the databased structure in order to maintain the size of the database, e.g. changing from MongoDB to RRDtool. Alternatively, this metric can be monitored by Cacti/Nagios, which already has the desired monitoring features.

Load refers to the workload of VPN server, including memory statistics, CPU statistics, and disk statistics. It is possible for CMon Agent to use SNMP query periodically on each VPN server with specific OIDs (1.3.6.1.4.1.2021) and forward this data to the central HQ. However, this can also be achieved by using the more powerful Nagios.

Latency measurements are very sensitive to clock accuracy. Such accuracy can be obtained through the Network Time Protocol (NTP) synchronisation to reliable time source. However, a virtualised environment such as the one used by the eduSAFE infrastructure (VM) is inevitably introducing delays in the packet's time stamping process.[4] Latency measurements performed in such an environment can be unreliable and should be considered with a bit of caution.. The tool of choice for latency measurements is OWAMP, but Ping can also be considered.

Throughput measurements can disrupt service if done without control, as traffic is generated and can saturate the network. To avoid that situation, only scheduled throughput measurements should be run at very specific moments, 4 or 6 times a day, at most, and for a maximum duration of 30 seconds. Depending on the throughput performance advertised to the end users, generated throughput can be close to the link capacity (in order to verify this advertised throughput) or be a fraction of it (in order to verify if a minimum throughput is available). The tool of choice of throughput measurements is iperf.

2.1.5 Performance Monitoring & Verification Solutions

Two types of implementation have been chosen for improving network performance functions: pre-production and production stage.

2.1.5.1 *Pre-Production Stage Monitoring*

Pre-production stage refers to the eduSAFE environment used by the eduSAFE developers and administrators, and where they implement and test new functionalities.

Performance monitoring and verification at this stage is important, as it gives eduSAFE administrators the opportunity to compare different design and implementation choices. In pre-production, administrators control the end devices and are able to implement a wide range of tools given that security posture and configuration of end devices is established according to the investigated performance aspects.

A high-level performance monitoring proposal using perfSONAR is depicted in Figure 2.2, showing, on one hand a eduSAFE lightweight version proposal (see segment 1), and on the other hand an eduSAFE extended MD-VPN one (see segment 4 and 5).

4 This happens because of the scheduling of the different VM by the hypervisors and because of conflicting access to shared resources (NIC). Virtualisation implementations vary and delays are different in each.

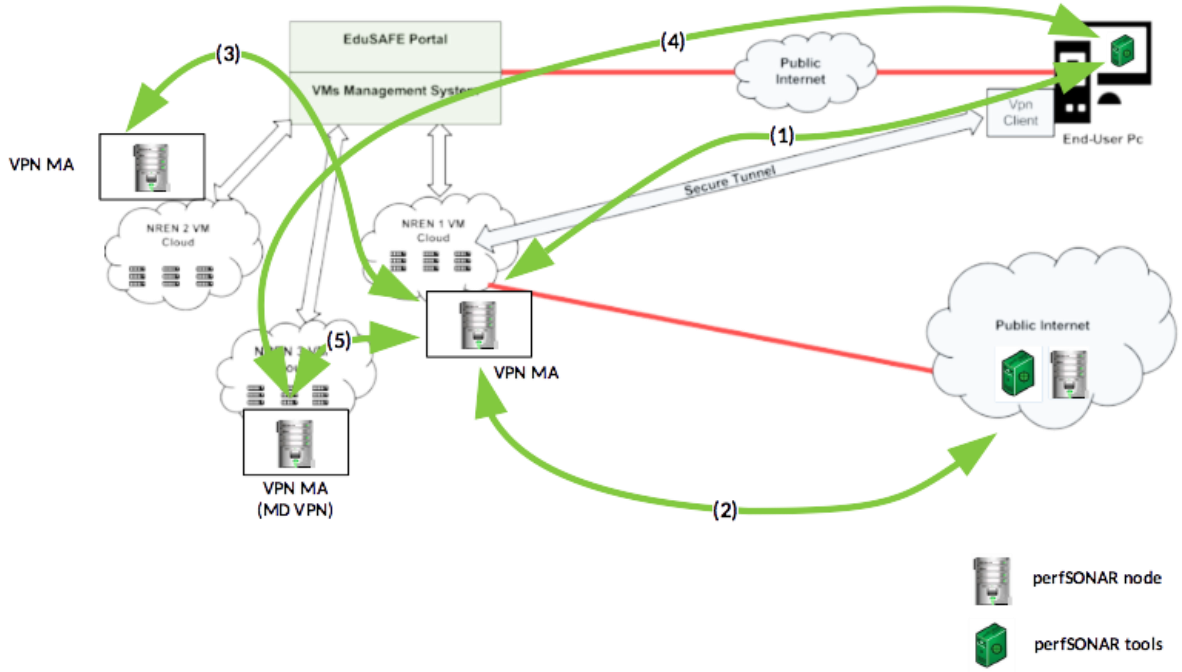

Figure 2.2: Performance monitoring in the pre-production stage of eduSAFE RA VPN

Installing a perfSONAR node in each NREN cloud, as part of the eduSAFE infrastructure, should enable the eduSAFE developers to obtain a complete set of performance monitoring data. The perfSONAR node will be the central component delivering a complete monitoring solution that will cover the active measurements aspect in the Remote Access (RA) VPN environment. This node will serve as the orchestrator of the measurements, it will serve as a measurement target, it will collect all measurement data, and it also allows the deployment of measurement visualisation tools.

When a new VPN is requested and setup by an eduSAFE user, automatic configuration of the central pS node should happen in a way that also connects the pS node to the newly created VPN. Then the end-user machines, with basic pS tools installed (iperf, OWAMP and BWCTL), can target the pS node located in the NREN1 VM cloud, as it is also a part of the VPN where the end-user PC is located (see Figure 2.1/Figure 2.2).

All the pS nodes distributed over the different VM clouds can also be used to monitor inter-cloud performance of other VPN servers. This will become useful once the eduSAFE VPN is distributed and relayed amongst different NRENs.

Therefore, the pre-production stage may be observed in five segments:

1. Intra-VPN performance monitoring will be observed in the segments designated (1) in Figure 2.2. Active Measurements are taken between the end user machine and the pS node located on the NREN cloud. As this node will be deployed as a separate VM, RA VPN features are not diminished when the measurements are correctly scheduled.
2. Internet access VPN performance monitoring will be observed in the segments designated (2) in Figure 2.2. Active measurements will be carried out between the pS node in the VPN and some chosen pS instances located on some specific location on the Internet. A set of different

pS instances can be used to provide different measurements to different part of the Internet (i.e. some close to the VPN gateway and some further away). These measurements should provide information regarding the performance pertaining to the Internet access and its gateway, and can deliver information of potential performance issues with outside links that may impede VPN-user Internet access.

3. Inter-VPN performance monitoring — active measurements between pS nodes in different NREN VM clouds, as depicted (3) in Figure 2.2. Necessity of inter-VPN performance monitoring is yet to be confirmed by the eduSAFE team depending on the proposed VPN use cases and the general eduSAFE architecture. Nevertheless, implementation of inter-VPN monitoring enables obtaining complete set of performance monitoring information for any observed use case.

4. Intra-MD-VPN performance monitoring will be observed in the segments designated (4) in Figure 2.2. Active measurements between the end user machine and the pS node located in the MD-VPN. As this node will be deployed as a separate VM, RA VPN features are not diminished when the measurements are correctly scheduled.

5. Inter-VPN performance monitoring, between eduSAFE and MD-VPN (5) in Figure 2.2. Active measurements are carried out between pS nodes in different VPN types. These measurements can assess the performance of the interconnection of the two VPN types.

By having the pS node taking part to each VPN and separate pS installation on the end user PCs, it is possible to customise the measurement tests, depending on the requirements of eduSAFE.

Limitations

The proposed performance measurement architecture has one major limitation that the eduSAFE team must be aware of. The pS node that will take part in each new VPN must support source-based routing, which is how multi-homing (i.e. having multiple network interfaces on the pS node) can be done with OWAMP and iperf. This also implies that the IP addressing used in each VPN must be different, they cannot overlap (i.e. two different VPN cannot use the same 10.0.0.x/24 subnet).

The pS node will also need to receive the VPN connection settings, as would any user taking part in a VPN, and this VPN setup needs to be automated on the pS node. Such functionality would need to be configured by the eduSAFE administrators, with support from the perfSONAR team.

If separated IP space cannot be assured between all the different VPNs, a pS node would need to be deployed for each new VPN. This could be automated in some way depending on the cloud environment, but scalability questions will arise if there is need to resort to such a setup within the GÉANT network.

Another limitation is the use of Network Address Translation (NAT). When monitoring from the VPN to the outside Internet, as with segment (2) in Figure 2.2, the monitoring solution described will not work correctly if NAT is used on the Internet gateway. This is a current limitation of the BWCTL component of perfSONAR that might be lifted in the future.

The last limitation is the OS that the perfSONAR nodes can be deployed on. perfSONAR packages are available for CentOS-6, Debian-7, Debian-8, Ubuntu-12.04 and Ubuntu-14.04.

Visualisation of measurements

By deploying a perfSONAR toolkit instance as the perfSONAR node, the eduSAFE administrator will gain access to a useful set of measurement scheduling and visualisation tools. On top of that, a perfSONAR web UI (psUI) instance can also be installed to provide a front end capable of running on-demand measurements.

2.1.5.2 *Production Stage Monitoring*

As previously mentioned, performance monitoring for the production stage of eduSAFE RA VPN cannot rely on end users installing the required software. In order to continue measuring the same metrics as those during the pre-production phase, a second pS node must be deployed to take part in the VPN. Deployment of this second pS node in a situation similar to an end-user machine, i.e. at the edge of an NREN network is recommended.

This second pS node can be provisioned with an automatic configuration feature similar to that of the first pS node. This way it would automatically take part in any new VPN set up by the end user. The central pS node can then schedule measurements towards this node as it would have done with end-user point in the pre-production stage. This setup covers the first (1) and the fourth (4) performance-monitoring segment.

Production stage active monitoring performed by this second pS node can be considered intrusive by end users.[5] If such a situation is not acceptable, and also to enable the end users to measure the performance as seen from the VPN, users can be provided with a Network Diagnostic Tool (NDT) server. This setup is part of the regular pS toolkit that is to be deployed on the pS node inside each NREN cloud. With this tool, the end users can initiate active measurements from their end devices, more specifically, using a web-socket-enabled web browser or a Java applet.

Relying on user-initiated monitoring will provide performance measurements that are not similar to the pre-production stage, as the tools are not the same. It will also prevent the eduSAFE administrator to collect data from measurements performed by the end users. However, it will provide the end users a means to test for themselves the quality of the VPN and can be considered an acceptable level of performance verification.

The second (2), third (3) and fifth (5) monitoring scenarios can be identical to the pre-production setup, described above.

2.1.5.3 *Performance Monitoring & Verification Environment*

For the eduSAFE use case, the Task 3 team agreed with the eduSAFE devOps team to focus our work on the “pre-production” stage implementation of the proposal; a lightweight and a MD-VPN version. For that, the following measurement points are built in three different VPNs:

- An eduSAFE safe-browsing VPN hosted by NORDUnet (see segment (1) in Figure 2.2).
- An eduSAFE safe-browsing VPN hosted by GRNET (see segment (1) in Figure 2.2).
- An eduSAFE to MD-VPN bridge VPN hosted by PSNC and connecting with MD-VPN to the GRNET network (see segment (4) and segment (5) in Figure 2.2).

[5] This can be mitigated by assuring the users that all software that is part of the measurement infrastructure is Open Source software, as is the software that actually runs the VPN.

Segment (2) was not implemented, as there are limitations of pS traversing a NAT, and segment (3) is not possible per the actual eduSAFE architecture [eduSAFE].

Each VPN used VMs hosted in the SWITCHengines cloud infrastructure to do the measurements [SWITCHENGINES]. Targets of the measurement were perfSONAR nodes deployed on the eduSAFE VPN servers of each of the setup described above (see Figure 2.2).

Pre-production stage performance measurements, the lightweight version and eduSAFE on MD-VPN can be viewed on the eduSAFE confluence wiki [CONFLUENCE].

2.2 GÉANT Testbed Service (GTS) – CMon on GTS

This section addresses the GTS system as a use case and how CMon can be used to address GTS's PM&V (monitoring) needs.

2.2.1 Overview

The GÉANT Testbed Service (GTS) offers experimental space for the construction of Layer 2 network testbeds for research across different participants. This enables research projects working across different administrative domains to effectively collaborate in order to engage in data transfer that is carried out in an environment that is self-contained, robust and provides consistency and uniformity. GTS provides dynamically provisioned network environments consisting of computational servers, data transport circuits, and switching/forwarding elements. These testbeds can be reserved in advance to coordinate testbed experiments with other non-GTS activities too.

Figure 2.3 shows the design of the GTS system in detail by depicting the various layers within it.

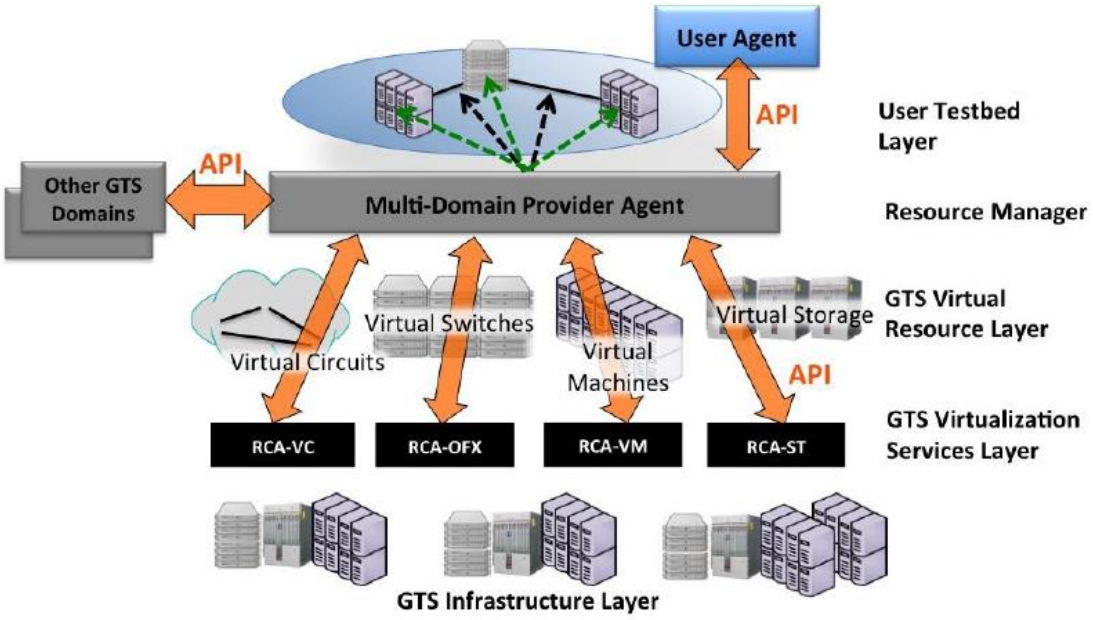

Figure 2.3: GTS Infrastructure layer *Source: [GTS]*

Detailed information about GTS can be found in *Deliverable D6.2: Architecture Description GÉANT Testbeds Service, Version 2* [D6.2].

2.2.2 Use Case Description

As seen from the GTS diagram above, the Resource Control Agent ring dual Circuit (RCA-VC) component of the Virtualisation Layer of GTS system uses OpenNSA to provision virtual circuits. In essence, RCA-VC is a provisioning entity and has the following state diagram for the creation and termination of virtual circuits [OpenNSA]:

- Define – define the resources needed.
- Reserve – reserve the resource(s).
- Activate – activate the resource(s).
- Query – query the resource(s) for usage, monitoring, etc.
- Deactivate – deactivate resource(s).
- Release – release resource(s).
- Undefine – undefined resource(s), so that they may be picked up by other testbed setups.

The virtual circuits setup by the RCA-VC component is reserved using network resources available in the testbed, and is defined by certain parameters. It allows data transfer between endpoints during its lifetime, therefore, it is essential for continued service-level use that these circuits are monitored for availability, as well as measured for performance. The selected monitoring tool should be able to perform both these functions seamlessly, present the results in a single interface that is easy to understand and interpret, and facilitate network engineers' work by making troubleshooting easy.

2.2.3 Feasibility Considerations and choice of Tools

Circuit Monitoring (CMon) is a distributed circuit monitoring application that performs end-to-end monitoring of point-to-point, Layer 2 circuits that traverse multiple domains [CMon]. These circuits may be dynamically provisioned with a provisioning tool (e.g., AutoBAHN), or can be static circuits (e.g., GÉANT Plus or GÉANT IP) [AutoBAHN], [GÉANTPlus], [GÉANTIP]. CMon is a simple, extensible tool that collects monitoring data from different domains participating in a multi-domain circuit, correlates them intelligently to appropriate circuits and reports on circuit status and useful metrics such as packet loss at interfaces. It enables easy monitoring via a single Graphical User Interface (GUI), providing a complete view of the end-to-end circuit so as to facilitate easy troubleshooting for NOC engineers.

Figure 2.4 shows the architecture of CMon:

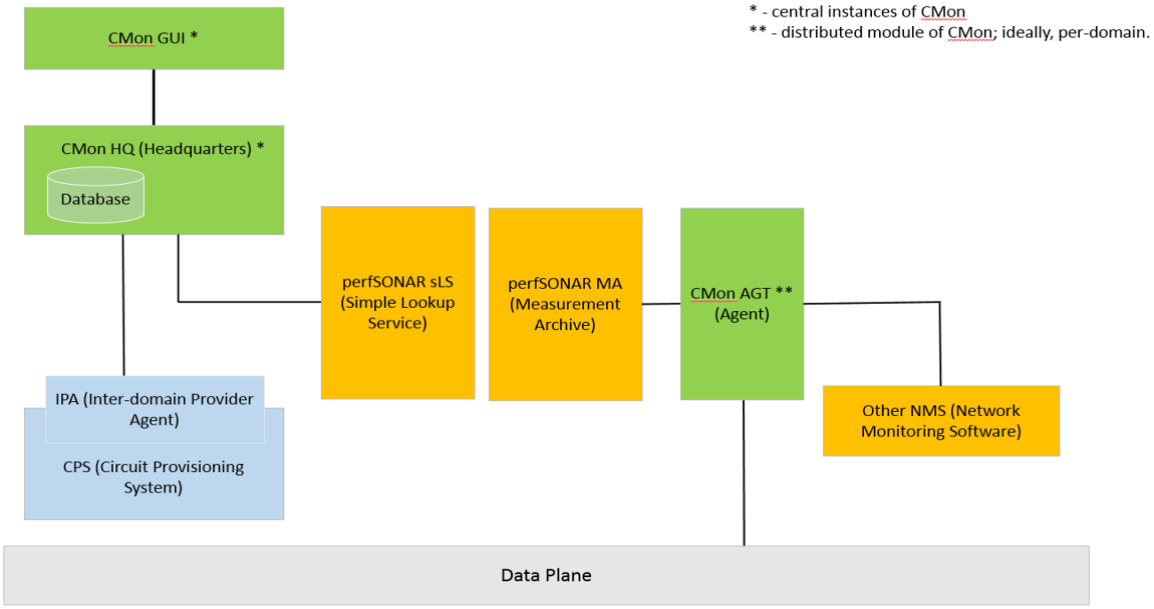

Figure 2.4: CMon architecture overview

CMon is designed in a star-like architecture, with the HQ in the centre and Agents installed in each NREN (domain), collecting data and sending to the HQ. The HQ is responsible for circuit topology reconstruction and matching the received data to the correct circuits. A central CMon webserver is responsible for hosting the GUI for users to check status of circuits, create circuit topology, and other admin tasks. The HQ interfaces with provisioning systems to receive circuit setup and termination notifications, and informs Agents, installed in each domain, to collect monitoring data. The Agents are designed to interface with different sources of monitoring data within domains, such as perfSONAR's central MA, a domain's own NMS, such as Icinga, Cacti, or bespoke NMS solution developed in-house in the NREN, etc., or even directly query the data plane via SNMP. The GUI provides a singular interface to view circuit information, and also has circuit scheduler features for static circuits, wherein each domain can independently setup circuits in preparation for it to go live.

The benefits offered by CMon[6] system is that it is a unique tool which provides end-to-end, multi-domain monitoring of circuits. It is designed in such a way that any changes within a component, whether the GUI, HQ or Agent, does not affect the other components. It uses MongoDB, a NoSQL database, and hence is capable of monitoring different metrics for different circuit types based on NRENs' requirements, and is also able to handle any resource changes in a virtual circuit, even if it occurs within the circuit's lifetime [MongoDB]. It saves the valuable time of NOC engineers wanting to troubleshoot issues, and can pinpoint exactly which part of the circuit is performing poorly, specifying the domain-level details. Such information is very useful in everyday situations as well as in critical projects, for maintaining high availability of the service.

[6] CMon: CMon is currently in 'pilot' stage in three domains: (GÉANT, HEAnet and PSNC) for monitoring Layer 2 circuits set-up on demand using AutoBAHN.

2.2.4 Performance Verification Metrics

As seen from the Use Case description (see Section 2.2.2), CMon will serve as the perfect monitoring solution for virtual circuits (VCs) provisioned by GTS system. CMon is highly flexible, and the modules are loosely coupled so that interoperability with any provisioning system only requires changing the interfacing module. The use of MongoDB makes it possible to cater to each domain's specific monitoring metrics (e.g. link status, packet loss, bandwidth utilisation) collection, as allowed by the domain administrators. The reported metrics on GTS use case availability and performance are listed below:

- Availability
  - Interface status – whether the VLAN interface is 'up' or 'down', which specifies whether the circuit that uses the interface is up or down.
- Performance
  - Packet Loss – number of packets lost during transmission, as measured at the interface.
  - Bandwidth utilisation (Tx and Rx) – interface bandwidth consumed by transmitted and received network traffic respectively, measured in bits per second.

Very little work is required in both CMon and GTS to inter-operate, as VC setup and teardown is already published in the JMS queue, and CMon Agents collect metrics, which are useful for monitoring availability and performance. Only CMon HQ needs to be enhanced with parsing OpenNSA-specific provisioning information as related to VCs.

CMon as a monitoring solution provider is also future-proof in that it works with NSI-based provisioning systems, which fits well with GTS GTS-CMon Agents col to expand to other domains, some of which use the aforementioned systems. For future plans of CMon, see Appendix B.

2.2.5 Performance Monitoring and Verification Solution

In the current implementation status of CMon, it can start monitoring VCs at the 'Reserve' can state. The following diagram and the steps in Figure 2.5, describe in detail how CMon can work with GTS system:

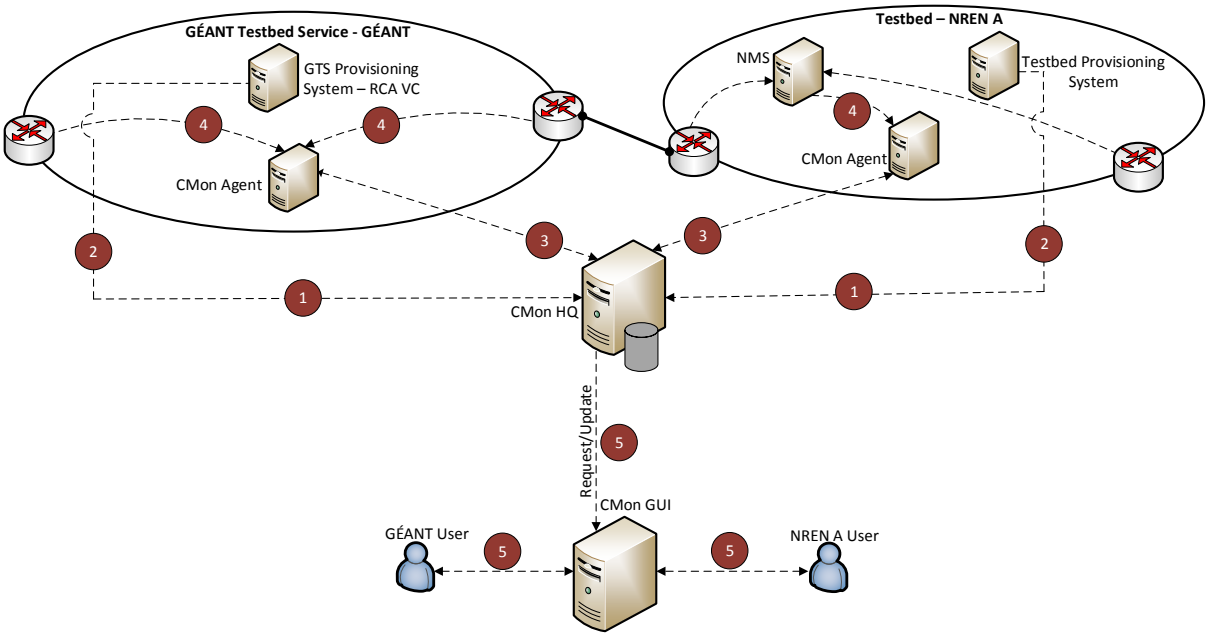

Figure 2.5: CMon for GTS – PMV solution architecture

1. When the VC is reserved, the RCA-VC component pushes this information to CMon HQ, so that it is prepared to start monitoring. This implies that the resources have been reserved and the request for a virtual circuit has been completed. At this stage, the circuit is not yet live and is not carrying any data. This step has already been implemented by RCA-VC component, in that it writes the activation information to a Java Messaging System (JMS) queue, from where external systems, such as CMon, can retrieve it.
2. The VC reservation information is pulled by CMon HQ, the single central instance, which is also, by function, an interface to the RCA-VC module from the JMS queue. The CMon HQ parses this information and stores it in the database. Since 'Reserve' state implies that the circuit is not carrying any data yet, it is not required for CMon to monitor the circuit in this state. Within the CMon HQ, this is marked as 'Inactive' state in the database.
3. When the virtual circuit is ready to go live, it enters the 'Activate' state in the RCA-VC, as described in Section 2.2.2. The RCA-VC once again pushes a message to signal this to the queue, from where the CMon HQ retrieves this information. It parses this and as a result, prepares to monitor the circuit. Internally, in the CMon system, the circuit is now marked as 'Active' in the database, implying that it is ready to be monitored.
4. To start monitoring, the CMon HQ contacts the relevant CMon's Agents installed in the domains that are partaking in that virtual circuit.
5. Depending on the monitoring method used by the domain, monitoring data (metrics such as link status, packet loss, bandwidth utilisation) is collected by the Agent and transferred back to HQ in a secure manner. The HQ collates this information and stores it in the database, aggregating in a manner such that the information is available to be displayed end-to-end for each virtual circuit passing through different domains.
6. The CMon GUI queries the database and displays monitoring and performance information for each circuit in a meaningful and easy manner. This ensures maximum clarity when debugging specifics such as which VLAN failure led to the circuit becoming unavailable, or which segment

of the multi-domain circuit is underperforming, thus reducing troubleshooting time and resolution time, to ensure steady performance.

Step 4 is iterated until the virtual circuit is ready for termination, i.e. the 'Deactivate' state. At this stage, the RCA-VC once again notifies CMon HQ of deactivation, at which point the HQ instructs respective CMon Agents to stop collecting monitoring data. Within the CMon system, all information pertaining to the VC is moved to a separate area in the database, so that it can be viewed later for analysis or reporting purposes.

Evidence of measurement and monitoring of circuits provisioned by the GTS system can be viewed on CMon GUI by the end of GN4-1 [CMon_MEAS].

2.3 Alien Wave Length – CMon for AWS

Alien Wavelength Service (AWS) is a network service defined with the service parameters (service level specification) based on the alien wave transport across the network [AWS]. This section address the Alien Wave Length service as a use case and how CMon can address the PM&V (monitoring) needs.

2.3.1 Overview

Alien Wave (AW) is a technical capability of an optical network to transport an optical signal of the agreed properties on a defined point-to-point path over the optical network layer without conversion to the electrical. There are three types of alien wave, defined as follows:

- Type-I alien waves refer to where optical signal is formed within the same electronic communications network, but not within the optical network layer. Examples of such alien waves are the communication (SFP/SFP+/QSFP+/CFP) modules capable of producing optical signal with the parameters (wavelength) suitable for transport over the optical layer. These modules may be installed directly into the routing/switching equipment and then transported through the optical layer as Type-I alien waves.

- Type-II alien waves refer to where the optical signal is not formed and received within the same electronic communications network. Typically such alien waves are used to connect third parties directly over the optical network layer.

- Type-III alien waves refer to where the optical signal forming/receiving equipment is not connected directly to the optical network layer within the same electronic communications network. Typically it means that the optical signal may have passed through one or more optical network layers and different communications network (without OEO transitions) before entering the particular communications network.

Alien Wave Type II and III are commonly understood as alien wavelength services.

Stakeholders:

- NRENs: currently NORDUnet and SURFnet are the major stakeholders.

- GN4-JRA1-T2 team is working on the AWS provisioning and demo.

2.3.2 Use Case Description

Alien Wavelength Service (AWS) is a network service defined with the service parameters (service level specification) based on the alien wavelength transport across the network. Alien wave Type II and III are commonly understood as alien wavelength services. Due to the nature of alien waves, the edge nodes of two domains, as shown in Figure 2.6Figure 2.5, need to share their optical parameters, e.g. transmission power, in order to correctly receive the optical signal and pass it into its own network without interrupting other optical signals or distorting the received alien wave.

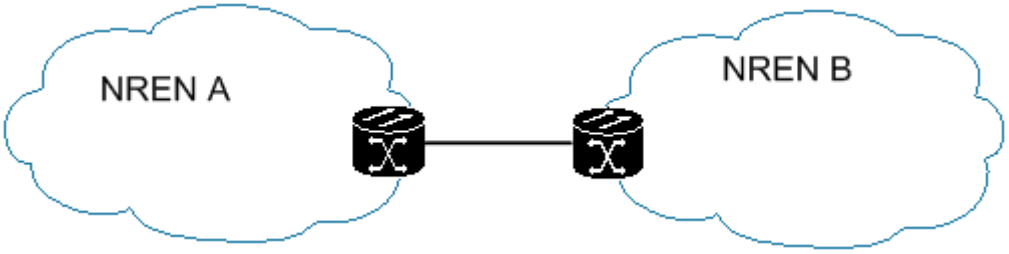

Figure 2.6: AWS use case

To provision such a multi-domain optical link, network administrators are heavily involved in sharing parameters, tuning optical transponders, and setting up edge equipment. The process is usually calculated in weeks or longer. In order to boost the provisioning process, the multi-domain optical link is required to be monitored and the information collected is required to be shared among the involving domains. Data from the monitoring system can be further used by provisioning scripts in order to automate the AWS provisioning process.

To offer AWS, an easy-to-use, multi-domain optical link monitoring, the data collector, CMon Agent, should be installed and correctly configured by each NREN. Each CMon Agent is managed by a NREN and will periodically collect data from the equipment and store the data in the central database held by the CMon HQ. By doing this, the HQ can reconstruct the link topology and map the received monitoring data accordingly.

2.3.3 Feasibility Considerations and Choice of Tools

This section describes the three main components of CMon, in terms of its feasibility for use with the Alien Wavelength Service solution.

2.3.3.1 *CMon Agent*

CMon Agent is the data collector responsible for collecting the monitoring data from individual NREN domains. Each domain can install one or more Agents in its network and configure them to collect different parameters. The Agent works in two modes, pushing and polling. In the polling mode, the Agent periodically queries the physical device for specific parameters and sends the data to the CMon HQ. In pushing mode, the Agent listens on specific ports and lets the device push data to it. The polling mode is more suitable for non-critical parameters, e.g. bandwidth, while the pushing mode is normally used by time-sensitive parameters, e.g. interface up and down status.

Current implementation of CMon Agent uses Simple Network Management Protocol (SNMP), supporting both SNMP traps and SNMP polling. This means that the NRENs must allow the Agent to perform direct SNMP query on their equipment, or enable the equipment to send or push SNMP traps to the Agent. This may not be possible for some NRENs, due to operation policy, security concerns or equipment limitations. Thus, to monitor AWS, CMon Agent is required to interface with local network management system. An adaptation module should be implemented, as shown in Figure 2.7, below. The adaptation module should be responsible for interfacing with various management systems, e.g. Icinga, Nagios, Cacti, etc. For the AWS, the provisioning team and monitoring team should collaborate on the adaptation to the local system.

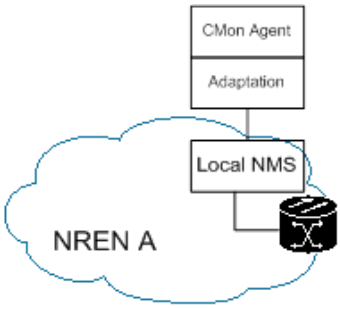

Figure 2.7: CMon agent

2.3.3.2 *CMon Headquarters*

CMon HQ is the central component of the monitoring system, as shown in Figure 2.8. The circuit topology should be sent to the HQ by the provisioning system or administrator before the monitoring can begin. The topology information is stored in the central database for later circuit reconstruction. When the monitoring progress begins, the HQ processes the monitoring data from different Agents and stores them in the database.

The current implementation of the HQ requires little modification in order to provide monitoring service to AWS.

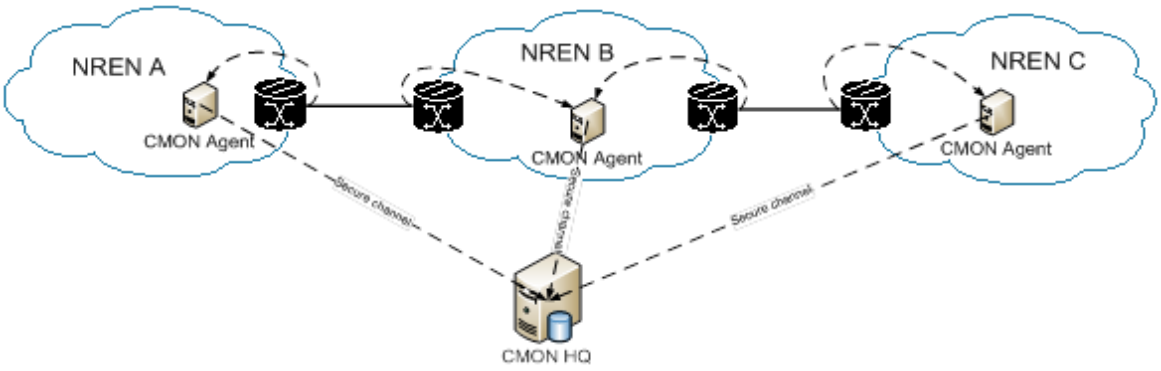

Figure 2.8: CMon headquarters

2.3.3.3 *CMon GUI*

CMon GUI is the web server that processes user requests and presents the monitoring results shown in

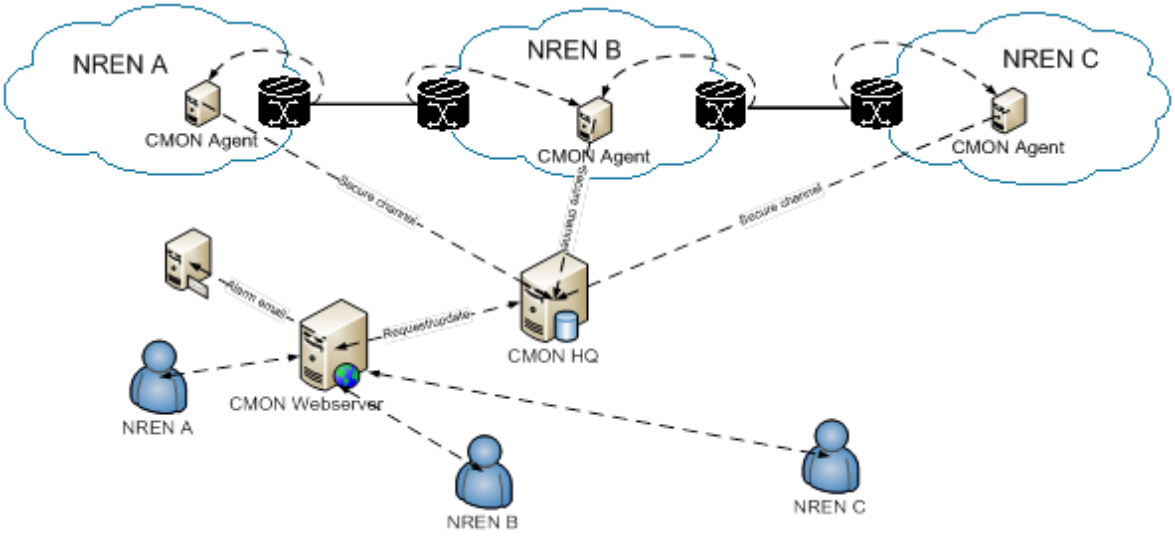

Figure 2.9. To monitor AWS, the GUI needs to add a separate category for the AWS links

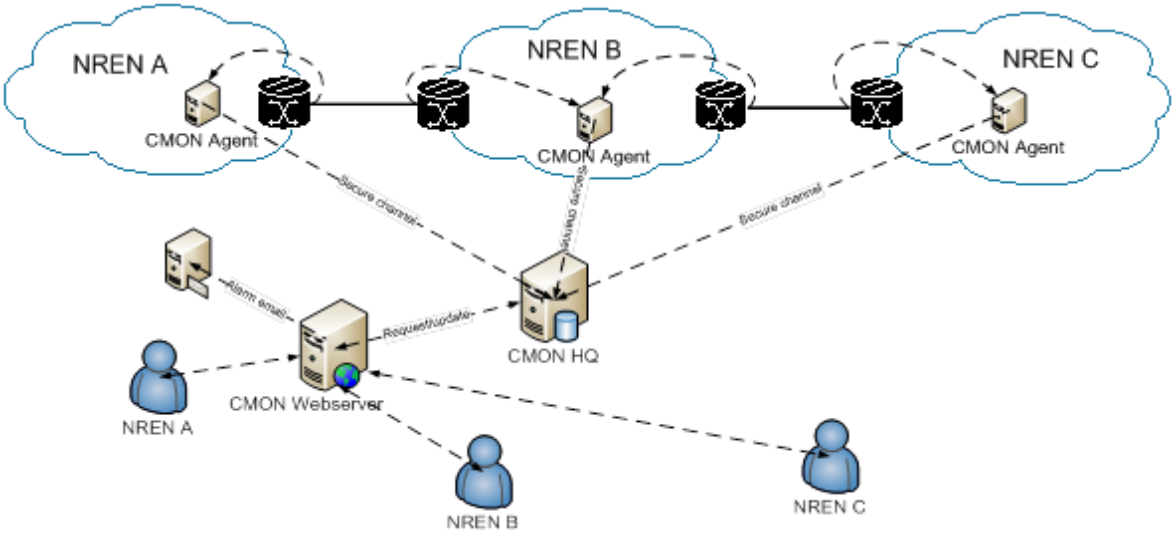

Figure 2.9: CMon GUI

2.3.4 Performance Verification Metrics

Monitoring metrics are requested and defined by the AWS NRENs, listed as follows:

- **Optical Signal-to-Noise Ratio (OSNR)**, it is defined as the ratio of signal power to the noise power, often expressed in decibels. A ratio higher than 1:1 (greater than 0 dB) indicates more signal than noise.
- **Pre-FEC BER**, the optical transport network (OTN) interfaces support monitoring the condition of an OTN link by using the pre-forward error correction (pre-FEC) bit error rate (BER). The Port

Interface Cards use forward error correction (FEC) to correct bit errors in the received data. As long as the pre-FEC BER is below the FEC limit, all bit errors are successfully identified and corrected and, therefore, no packet loss occurs. The system monitors the pre-FEC BER on each port. This gives an early warning of link degradation.

- **Input–output power level** compares the power-level specs for the optical shelf ADD/DROP ports and then compares them to the input–output power level specs on the router interface. Ensuring that power levels are compatible is important to maintain full performance. It may be necessary to adjust the equipment output power levels and/or add fixed attenuation on each port to get everything dialled in.

2.3.5 Performance Monitoring and Verification Solution

Alienwave service, especially the Type-III, can take advantage of the monitoring service provided by CMon. The workflow of CMon is described as follows (Figure 2.10):

1. **Topology construction**: AWS administrator enters the circuit topology information through CMon Webserver. The information is sent to CMon HQ and stored in the DB hosted on the same VM.
2. **Monitoring data collection**: CMon Agent hosted by each participating NREN periodically fetches the monitoring data and sends to the CMon HQ.
3. **Data storage**: Upon receiving the data, CMon HQ stores it in the DB.
4. **Data presentation**: Requested by any user, CMon Webserver reads the monitoring data from the DB and matches it with the circuit topology, so that the user can see the metrics on the multi domain circuit in seconds.

Monitoring metrics are requested and defined by the AWS NRENs, listed as OSNR, Pre-FEC BER, and Input-Output power level (see Section 2.3.4)

The architecture topology of CMon is a star topology, with CMon HQ located in the centre of the star topology and CMon Agents surrounded and installed in each NREN. With different configurations, NRENs can set up their Agent to monitoring different metrics using different method. Several solutions are proposed for the AWS monitoring as follows:

- **Solution A (SNMP Query)**: With the architecture described in Figure 2.10, CMon Agent should include the MIB OIDs of the aforementioned metrics of the equipment. The NREN then can install the CMon Agent on a virtual machine in their domain following the CMon Installation Guideline [CMon_INSTALL]. The NREN needs to configure the equipment to accept SNMP query from the Agent.

  After correct configuration, the CMon Agent installed in the NREN can periodically query the metrics from the device and further send the monitoring data to the HQ.

  The administrator the AWS should input the link topology through the GUI into the database so that monitoring data can be correctly mapped into the link skeleton.

- **Solution B (SNMP Traps)**: As shown in Figure 2.10, in case direct SNMP query is not possible in the NREN domain, SNMP traps can be configured in the equipment and sent to the CMon Agent periodically.

The NREN should install the CMon Agent on a virtual machine in its domain, following the CMon Installation Guidelines [CMon_INSTALL]. The NREN needs to configure the equipment to send SNMP traps to the Agent with a fixed interval so that data is pushed to the Agent periodically instead of being polled as in Solution A.

This solution can ensure that the address of the provisioning equipment of the NREN is invisible to the CMon Agent in case of security policy concerns.

- **Solution C (Interfacing with NMS)**: As depicted in Figure 2.10, for NRENs where neither SNMP query nor SNMP trap is possible, the CMon Agent is required to interface with the local NMS of the NREN. This requires collaboration of both the CMon team and the NREN, which could result in a longer deployment process.

  The NREN should install the CMon Agent on a virtual machine in its domain following the CMon Installation Guideline [CMon_INSTALL]. In order for the Agent to interface with local NMS, an extra module should be developed by the CMon team, if it does not already exist. For popular NMS, such as Nagios, Incinga, or Cacti, the CMon Agent can use the provided RESTful APIs to fetch monitoring data and send to the HQ. This requires the NREN give NMS access permission to the CMon Agent.

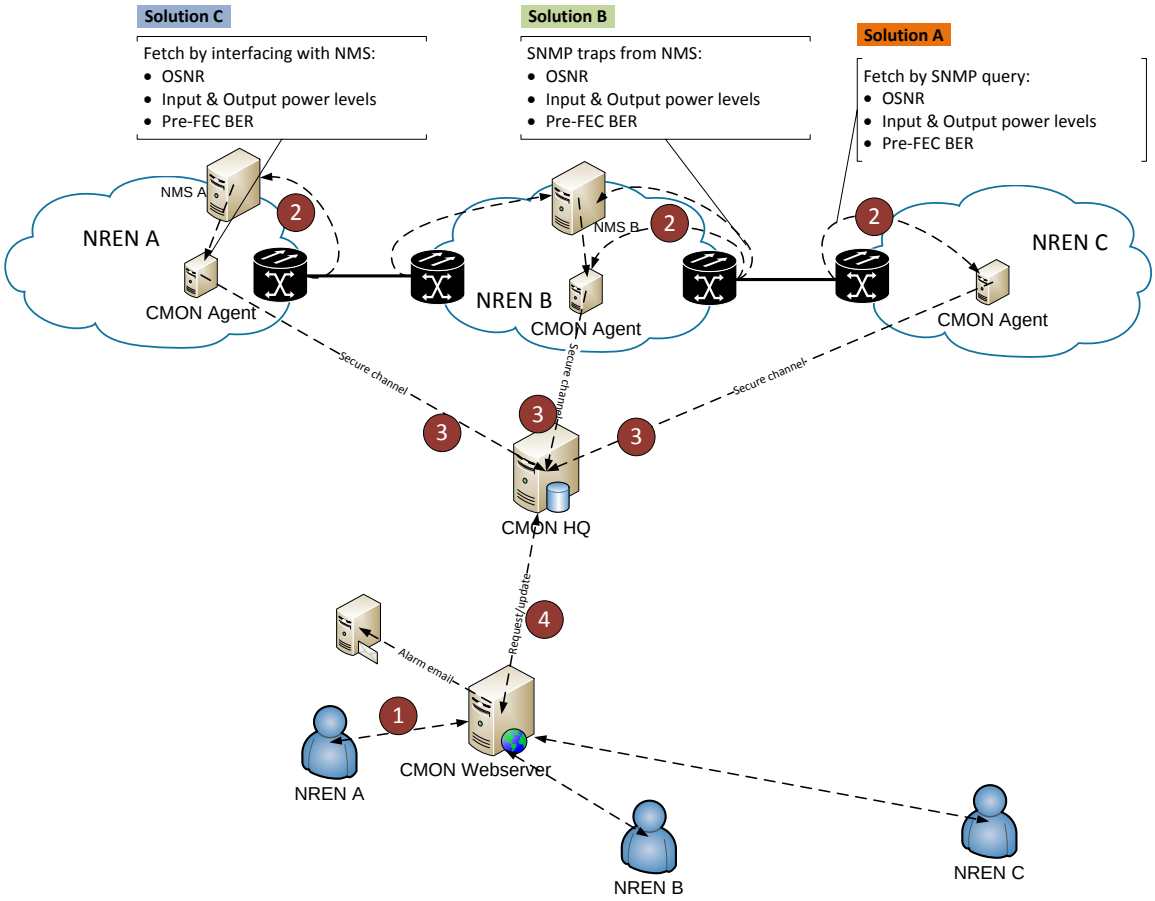

Figure 2.10: PMV solution architecture

At the point of writing this document the AWS deployment is work-in-progress. Evidence in the form of, measurements on circuit monitoring of the AWS use case are expected at the end of the GN4-P1 on the CMon GUI [CMon_MEAS].

2.4 MD-VPN – SQM on MD-VPN

The Service Quality Management (SQM) is a performance verification and monitoring tool (QoS) [SQM]. A first prototype was presented at the GEANT Symposium in Athens. The aim of our activities is to introduce a SQM PM&V solution on the GEANT MD-VPN service.

2.4.1 Overview

Multi-domain VPN (MD-VPN) is a GÉANT service that enables the establishment of private connections between two or more NRENs by using carrier-supporting-carrier (CsC) MPLS VPN technology. MD-VPN offers a range of connectivity options, i.e. point-to-point, point-to-multipoint, L2VPN, L3VPN, VPLS (currently in testing phase), etc. [MD_VPN]. Such diversity in connection deployment along with multi-domain, multi-instance and address overlapping features of MD-VPN service presents a challenging task for performance monitoring implementation.

2.4.2 Use Case Description

MD-VPN offers a seamless connection service between various NREN sites and is delivered with an SLA. As a consequence, details of achieved performance between VPN sites is necessary. Performance tracking is tightly related with the deployed topology (hub and spoke, partial and full mesh) that determines the way in which performance verification and monitoring are going to be deployed. Performance is described using technical parameters such as ***delay, jitter and packet-loss*** rate, which form a set of metrics clearly described in the Service Level Agreement (SLA). In terms of monitoring, the SLA stipulates measurement details, e.g. how often measurement is performed; specification of measurement traffic and how end-points between requested metrics are monitored. Prior to MD-VPN deployment, the SLA specified performance thresholds within a single MD-VPN instance between sites. Thresholds are determined as a trade-off between technical capabilities and user request. In most cases, technical capabilities deliver higher performance than requested. MD-VPN topology determines how SLA monitoring is going to be deployed, e.g. for hub and spoke, SLAs between spokes and hub are monitored, whereas for full mesh topology, the SLA is monitored for each pair of sites.

2.4.3 Feasibility Considerations and Choice of Tools

Service Quality Management (SQM) enables SLA monitoring for MD-VPN service. SQM presents a SLA tracking tool that is optimised for business processes, as well as performing active performance monitoring and verification. Additionally, it is equipped with service inventory that is used for tracking SLA technical parameters and parameters pertaining to individual sites within MD-VPN instances. Bearing in mind that the MD-VPN service is a multi-domain scenario with multiple-service instances, SLA tracking has to be instance-aware with the additional requirement that each instance may potentially have address overlapping. The overall picture of SQM components is depicted in Figure 2.11.

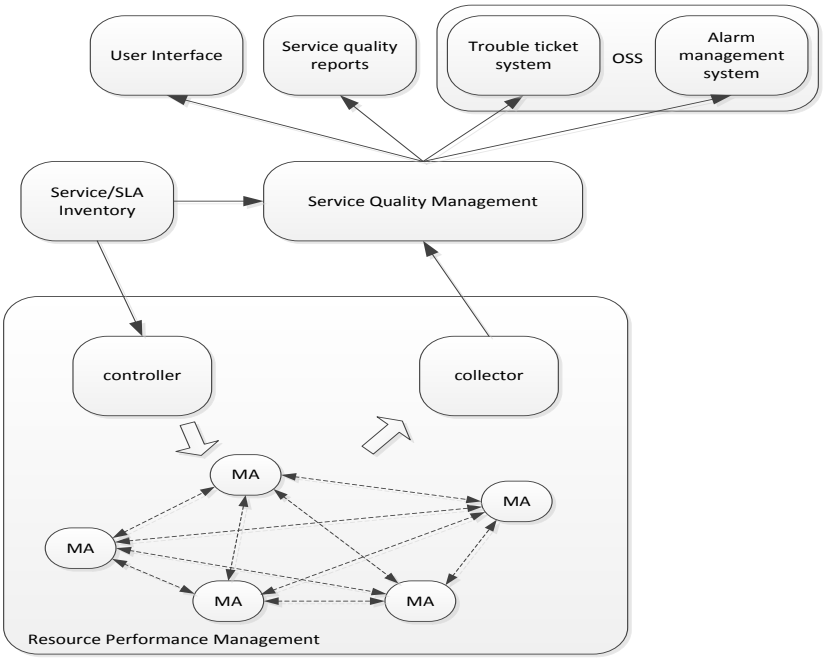

Figure 2.11: SQM components and architecture

SQM architecture consists of components that are entrusted with different functions, such as performance measurement, result collection, inventory management, performance reporting, etc. Notably, modular architecture allows SQM not to be solely limited to performance monitoring, but to also provide additional management activities relevant for service operation.

Service Inventory (SI) stores relevant information associated with separate services, i.e. service instances. Also, SI contains SLA information, i.e. performance metric thresholds, technical contacts, site locations, etc. A SQM Measurement Agent[7] (SQM-MA) configuration template is generated from this information.

Controller presents an integral component of the SQM, which schedules and executes measurements towards other sites, which are located in the designated service instance. For that purpose, each SQM-MA needs to have communication with controller via SQM management VLAN. Once measurements are finished, results are relayed to the controller, which passes it to the collector. Collector parses the results, stores it in the round robin database, which has a service instance ID assigned. Furthermore, collector performs centralised configuration of SQM-MAs by creating configuration templates stemming from service inventory. SQM-MA configuration involves network setup depending on service instances in which SQM-MAs are performing measurements. Bearing in mind high number of sites in a single instance, centralized configuration attributes to the scalability of the SQM solution.

OSS (Operations Support System) component tackles the challenges related to troubleshooting and incident management. SQM modular architecture may be integrated with a third-party trouble ticketing system, allowing more effective troubleshooting and incident resolution. Bearing in mind performance threshold specified in SLAs, it is important that SQM is equipped with the appropriate

[7] Measurement Agent is an IETF IP Performance Metrics (IPPM) working group term. As opposed to measurement agent, perfSONAR uses the term measurement point for entity that performs measurement tasks.

alarm component that would inform NOC engineers of any potential problem in operation and service performance.

Improved scalability of SQM is partly due to the SQM-MA's effective troubleshooting and incident resolution. Bearing in mind the performance threshold specified in SLAs, it is important that SQM is equipped with an appropriate alarm component to inform a NOC engineer. SQM-MA can receive configuration from the controller. Assuming that proper configuration is applied to the PE router, SQM-MA may be used to initiate measurement traffic and send the results to the collector. An example of SQM-MA deployment on the PE router is depicted in Figure 2.12. It is important to highlight that VLANs configured on PE routers are locally significant and that they should be also provided in the SI. Consequently, VLANs may be different for same service instance on different PE routers.

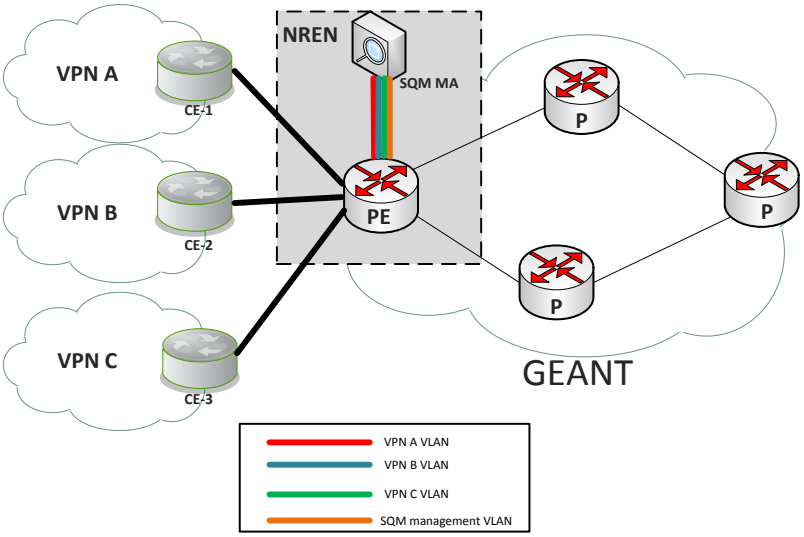

Figure 2.12: SQM on NRENs level

Verification of SQM functionalities, as described in Section 2.4.2, is shown using a SQM prototype as a PM&V solution for the MD-VPN service.

2.4.4 Performance Verification Metrics

OWAMP was used as an underlying measurement protocol for one-way delay, jitter and packet loss metrics and as a part of SLA tracking, appropriate thresholds for each metrics had been determined so that SLA may be verified. A SQM dashboard is depicted in Figure 2.13, whereas indicators take appropriate colour to determine measured value in comparison to SLA-specified metric thresholds between MD-VPN sites. From the dashboard, green indicators confirm that a specific metric value is below the threshold, amber indicates that metric is close to exceeding the threshold value, whereas red indicate that metric value is above the threshold and thus, in violation of the SLA. Since OWAMP is used to measure one-way metrics, inclusion of Network Time Protocol (NTP) was required, which may entail problems when VMs are used as SQM-MAs. However, as a result of SQM-MA ability to accept different measurement protocols, it is fairly easy to extend SQM-MA's capabilities in tracking other metrics, such as bandwidth, HTTP tests, DNS lookups, etc. According to the SLA definition, bandwidth does not present a performance metric that needs to be measured on a regular basis (only

during the service testing phase prior to service provisioning). SQM is not currently equipped with bandwidth measurement, but it can be extended to cover this as well.

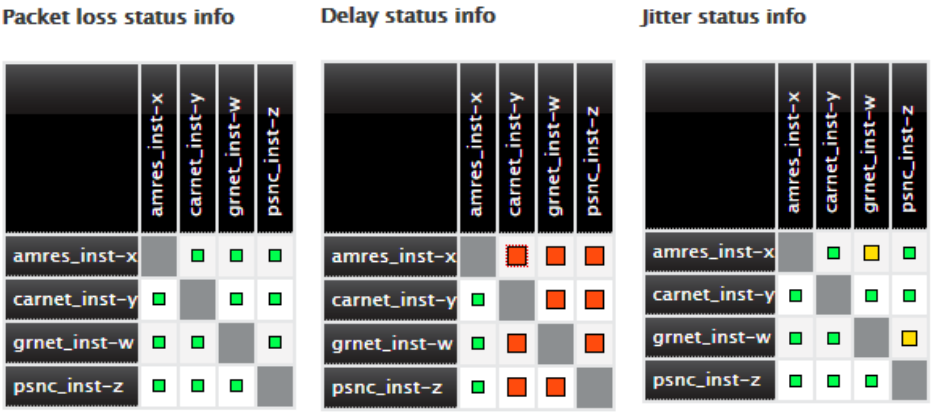

Figure 2.13: SQM dashboard for SLA verification in L3VPN instance

The primary aim of the SQM solution was to provide a scalable and comprehensive monitoring approach to MD-VPN service instances. SQM may easily be expanded to support additional metrics, depending on the SLA, whereas SQM-MA can be configured to adopt other measurement protocols and tools in order to fully address various service needs in terms of performance verification.

2.4.5 Performance Monitoring and Verification Solution

The MD-VPN testbed on which SQM is tested is depicted in Figure 2.14. Two MD-VPN instances are used: L3VPN between sites located on PSNC, GRNET, AMRES and CARNET networks and L2VPN instance established between AMRES and CARNET. On AMRES and CARNET sites, SQM-MA multi-homing ability is particularly highlighted, as a single SQM-MA is used to perform measurements for both L2VPN and L3VPN instances, regardless of their addressing requirements or connection type (point-to-point and point-to-multipoint). Additionally, for this testbed, different form factors of SQM-MAs were used (Beagle Board, CuBox, dedicated servers and VMs) in order to evaluate flexibility aspect of the SQM approach. Tested platforms did not show any difference in accuracy or limitations in terms of scalability [BEAGLE_BOARD], [CuBox-i].

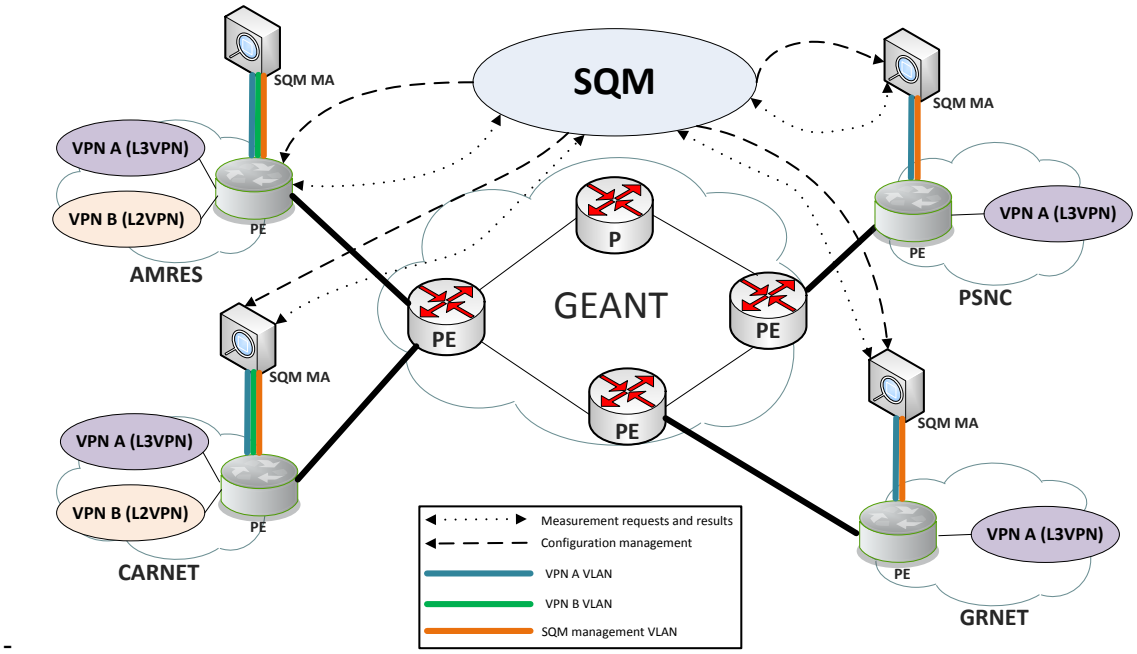

Figure 2.14: PMV solution architecture

Evidence of the SLA measurements was not archived, as SQM at the end of GN3plus was in prototype status, with first MD-VPN performance tests. The source code of the SQM dashboard is on GÉANT svn uploaded [SQM_DASHBOARD].

2.5 Wireless PM&V – Wireless Crowd Source Concept

The aim of this use case is to collect performance data on eduroam-enabled wireless campus networks. This concept was demonstrated live at the TNC2015 conference, to verify and monitor the performance at different locations (lecture rooms) and to analyse the performance behaviour based on chosen metrics / parameters [TNC2015] by correlations.

2.5.1 Overview

The Wireless Crowd source Performance Verification and Monitoring (WCSPMV) at TNC2015 was a first pilot, to implement a user-focus driven performance verification and monitoring on an eduroam enabled and/or guest-WLAN infrastructure [M7.2]. This work is a part of the SA3 Task 3 WiFi-monitoring task initiated as a greenfield approach for collecting experience, with focus on the end-user feedback.

2.5.2 Use Case Description

A summary of the Task's problem statement is: when a user reports a performance problem on WiFi, there is often not the data to help troubleshoot. If the problem affects a subset of users, or occurs only in particular circumstances (however frequent), then the usual approach of attempting to replicate the problem often fails. There is a need to provide performance data for the academic ICT

respective campus network providers to show the users' perspective having the end-user feedback on network performance data. For providing needed data, our collectors, resp. parameters are:

- Bandwidth/latency test in the browser (e.g. at Chrome, Firefox, Safari, Internet Explorer, etc.).
- The aim is to have JavaScript in place gathering data at chosen parameters, which delivers the performance key figures, upload and download speed of images (1 to max 5 MByte).
- Identifier (ID) of the Access Point: The aim is to get this information from the RADIUS logs, when a valid session was established, starting of a successful authN/Z session.

Expectations from the measurements include:

- Bandwidth and latency tests in the browser.
- The IP-address to the user; this value will be a correlation between the user-mac-address from the RADIUS logs, and the IP address from the DHCP logs delivered by the network provider.

Thus the values, (KOPIs) that are important for measurements, are: Client-MAC-address of the user / The access point (AP) mac-address and ID / Client IP-address from the DHCP log and the time stamp.

At TNC2015, the RADIUS accounting logs turned out to be the most essential and the easiest way to provide the necessary mapping between performance data and AP. In other institutions other logs might be needed. The APAN40 Conference confirmed that it is not self-evident that required log files are available, having access to them, and to install the JavaScripts on (sub)web-sources [APAN40].

2.5.3 Feasibility Considerations and Choice of Tools

Traditional performance measurements show the infrastructure point of view, answering the question "WHAT" will be measured. What is missing is the answer to the question "HOW" things will be done, namely, the end-user feedback. Thus our motivation is to verify:

"…*It is possible to gather data from multiple sources, including browser-based measurements in addition to traditional monitoring, and extract meaningful information on the performance of a WiFi network from that data…"*

Regarding this motivation the data gathering continued in two parts:

- Use JavaScript on a website that is frequently visited by a network's user to run non-intrusive performance tests of that network.
- Separate or correlate the tests results by AP, using RADIUS data or other means.

This data was used to build a picture of the performance experience by users on the network, broken down by AP (so that problems can be traced to an individual room or AP).

The objectives at TNC15 were:

- To try to validate this procedure as far as attempting to get the tests run by a number of participants at the conference.
- To correlate the results with the APs.

If those objectives were reached, then there would be data to analyse and an understanding gained regarding what results would be reasonable and possible next steps.

2.5.3.1 *Building Blocks*

Instead of building a full first system based on the WiFiMon architecture below, it was decided to dive right in and implement sections. These were piloted in real-life scenarios, TNC2015 use case and more, so that results could be quickly gathered and used to guide the rest of the Task 3 team's work.

Because this is an iterative approach of quickly prototyping and adjusting, it is expected that requirements will change. Therefore, during the project implementation, some of the proposed technologies may also be changed. However, the overall picture is focused, as follows:

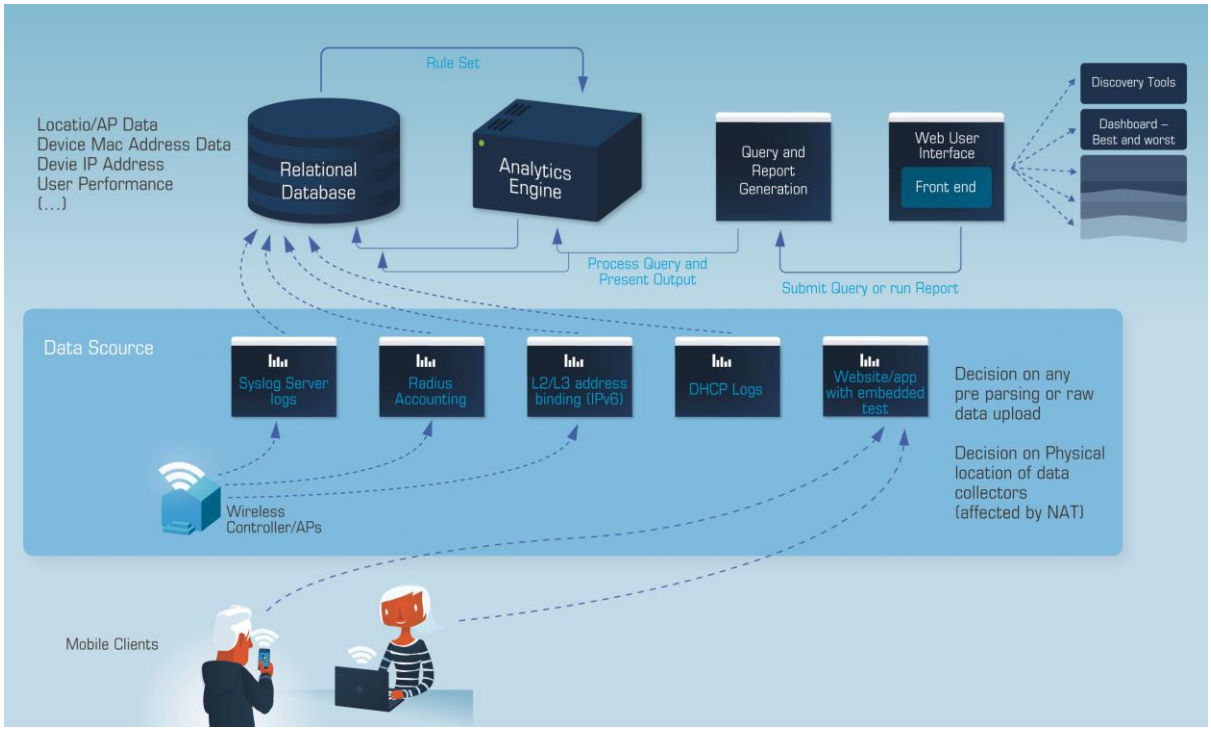

Figure 2.15: Building blocks wireless performance and verification

There are two items of data required: the results of a performance test, and which access point the user was connected to when that test took place. The performance test itself is easily generated and recorded when the user visits websites (part of Data Sources) with the JavaScript embedded.

Mapping this to an access point, however, is more complex and varies from site to site. Therefore, the "Data Sources" part of Figure 2.15 allows for any of a number of different sources: RADIUS accounting logs, DHCP logs, IPv6 address binding, to be used to create that mapping. It is important to be mindful of user privacy here, because while the data sought is quite innocuous, these sources contain quite a lot more information about user behaviour and identity that must be carefully handled.

The required data is gathered on a relational database, so that the various inputs can be correlated and then queries run. From that point, there are tools to generate the appropriate information,

including: an analytics engine for specific queries, a report generator for long-term trends and visualisation, and of course, a front-end user interface (GUI).

Details of the WCSPMV technical architecture and the concept are available in the M7.1 document and from the guide/dashboard [M7.1] [WCSPMV].

2.5.4 Performance Verification Metrics

As described in Section 2.5.3, the WiFiMon setup consists of two logical parts: the JavaScript front-end performance tests, and the backend log analysis. Nettest hosted on a server in Athens was used for the performance tests at TNC2015 [NETTEST]. The Nettest.js operated by downloading an image, which, in this case, was configured to be 1Mbyte in size and measured **download, upload speed and latency** of the request. In a conference lightning talk, we showed (in near real time) the download speed and latency as possible proxies for network performance.

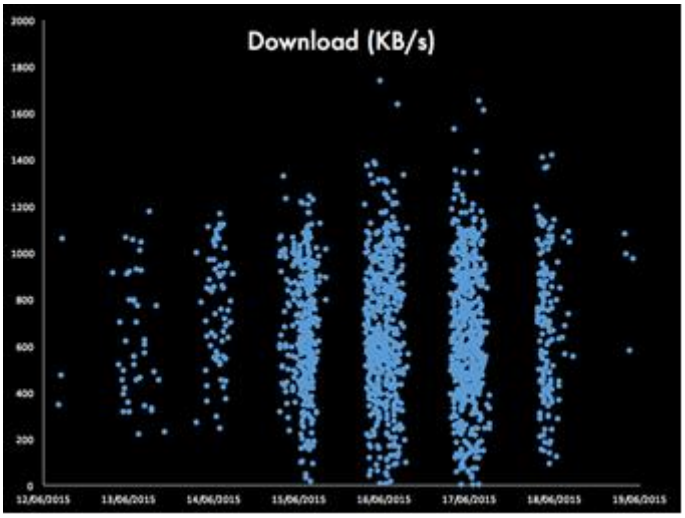

Figure 2.16: Download speed

Download speed showed a wide spread of results (as shown in Figure 2.16), with no clear patterns, probably in part due to the fact that only a single 1 megabyte image was used for the tests. It is likely that this isn't enough to get a clear, reliable view of the actual download performance experienced by a variety of users. The image size can be increased, and other tools that perform multiple download can be used. A decision was taken at TNC2015, not to do this because of the uncertainty of the impact the tests would have on the overall network, but it seems clear now that it is safe to use more accurate tests.

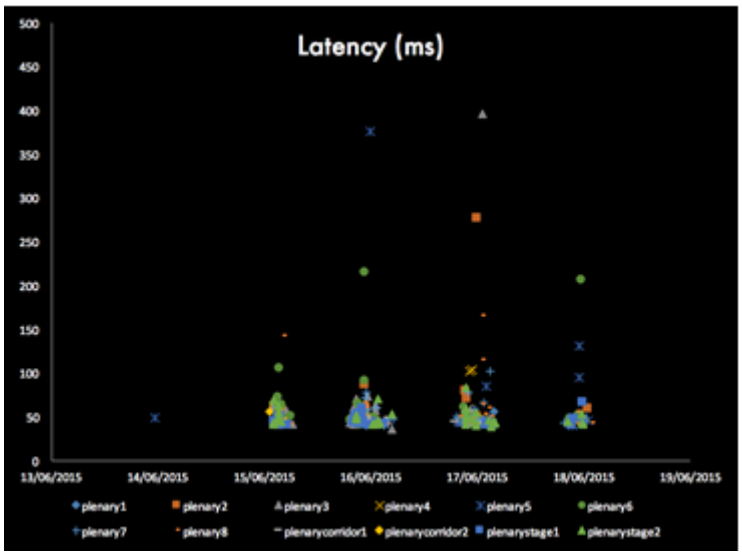

Figure 2.17: Latency tests

However, latency of the requests was also measured, which did show a number of clear patterns (see Figure 2.17). In particular, looking at the chart of latency in milliseconds of requests in the plenary rooms, there are three, clear characteristics:

- There are no results below about 40ms, which are believed to represent the latency between the conference location (Porto) and the test server (Athens).
- Most results cluster in the 10-20 ms range above that, which is indicative of healthy network activity.
- There are fewer results above that range, and they become very sparse, but are still present, above 100ms.

As a first result, this is a good proxy for network performance, and the outliers showed isolated moments of problematic connectivity on the network.

For this test, the test server was some distance away from the site, which means that the problems could, in principle, have occurred anywhere in between the server and the site. However, if investigating a problem that was specific to a certain room at the site, this would show up when results were compared room by room.

2.5.5 Performance Monitoring and Verification Solution

The mechanism as depicted in Figure 2.15 is used to generate the results at TNC2015, while rudimentary and custom to that location, was nonetheless close to the principle of the planned architecture (and indeed, in the spirit of rapid iteration, guided the design of that architecture).

The JavaScript monitoring was performed on every page of the TNC2015-website, with a cookie (expiring after one hour) used to prevent the test being rapidly re-run by the same browser. (This appears in the architecture diagram).

The RADIUS accounting logs and test results were then processed using a pair of Python scripts that were written for the purpose. First, the RADIUS accounting logs were processed as described above, to associate each IP address with an access point and timestamp of when that IP address was associated with that AP. This was stored in an SQLite database. Then the database of test results was converted into a CSV. Each row in the CSV was one test result. Each test result had a timestamp and an IP address, and queried the SQLite database to find the post recent access point with which that IP address had been associated prior to that test being run. (This is the data that is loaded into the relational database in the architecture diagram).

This left a new CSV of test results that recorded download speed, upload speed and latency, along with access point and other information such as platform and browser names. (In the architecture diagram, this would be the equivalent of the relational database.)

This CSV was loaded into Excel and used to generate the charts that were used in the Lightning Talk. (This was the equivalent of the analytics engine and report generator in the architecture diagram.)

The 5-minute talk at TNC2015 can be seen at the “Thunderclaps”, beginning at the timestamp 39m30s [LIGHTNING]. Evidence supporting the use case measurements TNC2015 can be viewed on the WCSPMV-dashboard (Table of Measurements) [TNC_MEASURE].

3 Qualitative PM&V Solutions Assessment

This section assesses the described use case and PM&V solutions (as introduced in Section 2) on six, technology-agnostic independent and relevant questions for exploring collaboration solutions (detailed in Section 4) recognised for a future PMV architecture framework. Questions have been prepared for PM&V assessment on the following six topics: integration, reusability, extensibility, control and data management, deployment scalability and application performance and standardisation.

3.1 Integration

Question(s): Is the monitoring/measurement solution able to exchange monitoring or/and control data with the monitoring tools/systems that are already deployed and running in the network infrastructure? Such integration should not:

- Require additional substantial software development (only minor updates with configuration adjustment are suffice).
- Interfere/restructure exiting and operational monitoring approach in an infrastructure (it does not require an update of operational procedures and workflows).

Evaluation:

Use Cases / Question	Integration
perfSONAR on eduSAFE	perfSONAR measurement tools can integrate seamlessly with a Nagios install to provide alarm capabilities (Nagios plugin scripts are provided by perfSONAR). As an example, this would allow the eduSAFE monitoring solution to be integrated into any existing Nagios deployment.
CMon on GTS	The CMon solution has defined a set of RESTful APIs to implement for better integration with other systems, i.e. provisioning systems in northbound API, and bespoke or industry-favoured NMSs, such as Nagios, Icinga, Cacti, etc. in southbound API.
CMon on AWS	Apart from RESTful APIs for integration as mentioned above, CMon is capable of integrating with AWS system to receive automatic notifications of provisioning messages, should it be implemented by the AWS service.
SQM on MD-VPN	The concept of SQM allows seamless integration with existing performance monitoring tools, especially since its operation (measurement scheduling) is based on Nagios. As a result of being able to deliver multi-instance service performance monitoring, SQM may be easily adapted to different measurement scenarios. Consequently, its integration properties are high, given that SQM-MAs can be

	configured for a wide range of performance metrics specified according to the SLA. Centralised configuration of SQM-MAs delivers additional integration merit as a result of simple deployment of multiple SQM-MAs spread across the network.
Wireless Crowd Source PM&V	The Wireless Crowd Source Performance Measurement and Verification method (WCSPM&V), looking for the end-user feedback is complementary to the objective PM&V measurements by HW probes. All modern databases, including PostgreSQL and InfluxDB that have been used offer a REST API. It should therefore be easy to exchange monitoring data with other systems without any substantial software development. For example, it should not be too difficult to integrate the results from WiFiMon into a traditional network monitoring system that can generate alarms if WiFiMon detects any problems.

Table 3.1: Integration

3.2 Reusability

Question(s): Is it possible to reuse selected component(s) of the monitoring solution in the existing monitoring tools/systems to provide a missing functionality? Reusability should not require additional substantial software development (only minor updates with configuration adjustment are suffice).

Evaluation:

Use Cases / Question	Reusability
perfSONAR on eduSAFE	The basic blocks of the perfSONAR monitoring solution that have been deployed for eduSAFE can be reused by other perfSONAR visualisation tools: • MaDDash: used to visualise a measurement mesh through a dashboard, with colours to quickly identify problems. • psUI can be used to display the measurements stored in the Measurement Archive.
CMon on GTS	CMon uses SNMP for polling and receiving traps from devices in order to collect monitoring data in GTS. This module is a lightweight module that is reusable by other software.
CMon on AWS	The CMon solution uses standard open source SNMP Python library for querying and receiving traps from the equipment. The SNMP core is certainly reusable by other software.
SQM on MD-VPN	Current SQM deployment uses only OWAMP tool as an underlying measurement protocol for monitoring delay, jitter and packet loss rate. Thus, SQM-MAs facilitate only active measurements, whereas passive measurements may be deployed using SNMP established in multi-instance environment. Combined active and passive monitoring allows delivery of complete set of performance metrics, which are specified according to the SLA.
Wireless Crowd Source PM&V	Functionalities and components as JavaScript (on the websites) Access Point Identifier (AP-ID), DHCP-/Radius logs (Wireless Controller Log files) and a valid time-stamp (access to the WLAN) is used for making correlations, independent which project or initiative will be continued (see Section 2.5.3.1).

Table 3.2: Reusability

3.3 Extensibility

Question(s): Is it possible to extend the monitoring/measurement solution by adding new functionalities without changing the implementation architecture and interfaces. Does it support existing well-known interfaces, protocols and software implementation approaches (e.g., SNMP, SOAP, JSON, SSL, REST, OSGi, etc.)?

Evaluation:

Use Cases / Question	Extensibility
perfSONAR on eduSAFE	The perfSONAR Measurement Archive (MA) has a REST interface that provides access to all stored measurements. The measurements are provided in JSON format for easy consumption by other tools. In the near future (second half of 2016), perfSONAR will have extension where users should be able to define new measurements tools and new metrics (BWCTL, that is used in perfSONAR, will have extension points where one can easily define new measurement tools, for example, to replace iperf with udpmon, or even new metrics and their corresponding tools).
CMon on GTS	CMon modules are loosely coupled in that any update to one module does not affect other modules as long as the interface between them is unchanged. This makes the system extensible. The CMon AGT (Agent) is the only lightweight module that the NREN is responsible for updating. CMon Agent now support standard SNMP, SSL, SOAP, and JSON.
CMon on AWS	The above holds true for CMon solution for AWS use case too. AGT will be updated with new features and is the only module that NRENs will need to upgrade.
SQM on MD-VPN	Introduction of namespaces in SQM tool allows separate measurement specification per individual service instance. Aside from adding SNMP functionalities into the SQM, additional measurement components may be introduced such as ICMP ping and Ethernet Operations, Administration and Maintenance (OAM) components. By avoiding significant modification of SQM tool implementation of additional performance metrics is possible. SQM extension with Ethernet OAM would allow L2 monitoring of L2 VPN circuits without SQM-MA IP addressing requirements as well as better overall insight in L2 circuit performance. Current SQM version does not support SOAP, JSON, SSL, REST, OSGi.
Wireless Crowd Source PM&V	Our architecture is modular (see Section 2.5.3.1 Building Blocks) and WiFiMon is built on a Service Oriented Architecture (SOA). It's based on Java micro services that concludes any new functionality can be added as a new micro service.

Table 3.3: Extensibility

3.4 Control and Data Management

Question(s): Can the monitoring/measurement solution be deployed in the multi-domain network infrastructure and measure the metrics of data transfer through multiple domains? Can monitoring/measured data be distributed among network domains and be managed by independent

domain administrative entities? Can distributed monitoring/measurement components be managed by independent domain administrative entities?

Evaluation:

Use Cases / Question	Control and Data Management
perfSONAR on eduSAFE	perfSONAR was built for multi-domain monitoring from the start: • The MA provides access to store measurements from other domains. • The OPPD and BWCTL components enable a user from another domain to trigger measurements remotely. • The psUI has federated access allowing users from other domains to visualise measurements in different domains. The mesh-config server and agents facilitate the configuration of a measurement mesh spanning multiple domains. BWCTL limits and firewall rules can be set to allow measurements requests coming from trusted parties only.
CMon on GTS	The CMon solution is a distributed, multi-domain monitoring system. It is designed to work in multiple domains, stitching together monitoring information for a multi-domain circuit. Each NREN can install, configure, maintain, update, manage, add and remove its own CMon Agent(s) independently.
CMon on AWS	The same as above, as CMon is the same as above in design and functionality for AWS service as well.
SQM on MD-VPN	SQM presents multi-domain measurement tool, which can be used to perform measurements of performance metrics across multiple domains and service instances. However, it is not designed to be managed by multiple domains. The results of the measurement tests are available for individual SLA participants in order to perform SLA validation. Initiation of performance tests may be implemented with few modifications to the SQM tool, particularly SQM controller component.
Wireless Crowd Source PM&V	In Wireless Crowd Source PM&V, the multi-domain nature is understood as multi-locations on a campus or ISPs WLAN. It can be deployed over multiple locations with collecting measurements according the requirements of the use case (in this case, PM&V). From the concept, generically spoken the access to the measured/visualised data can be selected (domain based) – target audience is usually the academic ICT, resp. NOCs or research organisations of the (N)RENs.

Table 3.4: Control and data management

3.5 Deployment Scalability and Application Performance

Question(s): Is the monitoring/measurement solution prepared to work properly without additional management and performance burden in case of larger network infrastructure to monitor or complex configuration?

Answers:

Use Cases / Question	Deployment Scalability and Application Performance
perfSONAR on eduSAFE	perfSONAR Measurement Points (MP) are easily deployed and do not need many resources. They can be configured on a small appliance or a VM with proper hardware. A central perfSONAR MA can scale to hold measurements coming from tens of MP. The perfSONAR mesh-config agent eases the maintenance of a measurement mesh by holding all configuration in a central place. In the near future (second half of 2016), the central piece of the eduSAFE monitoring architecture will be able to make measurement reliably on different VPN even if those are using virtual interfaces and overlapping address space.
CMon on GTS	CMon uses a No-SQL database so that it can be used to measure and record metrics as per a domain's requirements, and also is scalable. The central CMon HQ is able to handle multiple traffic from CMon Agents (AGT) installed in different domains.
CMon on AWS	Same as above, as the CMon design and architecture is unchanged for AWS use case.
SQM on MD-VPN	Given that SQM has only passed the proof of concept test, detailed scalability tests have not been conducted yet. However, given the architecture of SQM-MAs, underlying protocols and tools as well as MD-VPN use case, SQM solution is prepared for high scalability requirements.
Wireless Crowd Source PM&V	In Wireless Crowd Source PM&V modules can be added that will gather data from multiple Web sources. From a generic point of view the scalability is given, and other micro services can be used for collecting data from other sources (e.g. non free radius protocols). Today, the processor micro service collects data from free radius protocol. From the performance point of view, a large-scale deployment on a big campus can be an issue, as there is a risk of too many students doing performance tests at the same time. If round trip measurements are considered this should not be a problem. Today It is not tested on large deployments, but it is planned.

Table 3.5: Deployment scalability and application performance

3.6 Standardisation

Question(s): Does a monitoring/measurement tool have new categories* to combine with/integrate traditional categories of active, passive measurement techniques without changing the fundamental architecture of the tool? *category = Methods and Metric.

Answers:

Use Cases / Question	Standardisation
perfSONAR on eduSAFE	perfSONAR uses standardised metrics such as OWAMP [RFC-4656]. The perfSONAR MA is versatile and allows for storage of new metrics.
CMon on GTS	The CMon solution interfaces with provisioning systems to gather circuit information, which may be as per standards such as NSI or OpenNSA, and also

	collects metrics by standard SNMP methods from the equipment MIB database when it comes to collecting monitoring data from individual domains.
CMon on AWS	Same as above (CMon solution for GTS).
SQM on MD-VPN	Currently, SQM supports only active measurements via OWAMP protocol, but can be easily extended with SNMP to perform passive monitoring per instance. Therefore, extension with passive monitoring is possible without changing the fundamental architecture of the SQM.
Wireless Crowd Source PM&V	The implementation is based on a SOA, from the measurement on Java micro services and works (today) with WLAN standards 802.1xx (eduroam). Also other well-known tools are used, like InfluxDB (NonSQLbased) and PostgreSQL.

Table 3.6: Standardisation

3.7 Assessment Conclusion

One of the goals of Section 3 is to prove that the PM&V solutions presented in Section 2 are not limited to only five use cases, which are the network services being investigated in GN4-1. These services – eduSAFE, GÉANT Testbed Service, Alien Wavelength, MD-VPN and wireless crowd sourced systems – have certain monitoring and verification requirements that can be met if a set of metrics is measured and a set of functionalities are present, along with some characteristics.

The SA3 Task 3 team has proposed a PM&V solution for each aforementioned service, as well as characterised those solutions to show their benefits and strengths. Such information will instruct and support network engineers in the process of selection of PM&V solutions for other network services. Explained characteristics will help to decide whether perfSONAR, CMon, SQM or the Wireless Crowd Source tools are suitable for monitoring and verification requirements a network engineer deals with. In addition to the descriptions of five, well-defined use cases and their PM&V solutions in Section 3, these tools will be useful for new use cases of other network services.

In order to describe the PM&V solutions independent of the use cases, six characteristics have been used: **integration, reusability, extensibility, control and data management, deployment scalability and application performance, and standardisation**.

The first one, **integration**, indicates if an analysed PM&V solution is able to exchange monitoring and control data with other applications, which are already deployed and running in the production network infrastructure. Normally, a network operator does not want to replace all its tools but only replace some components or add new ones in order to improve the network management, and thus decrease operational costs. All the described PM&V solutions can be easily integrated and a part of heterogeneous platform by using standardised interfaces (see Table 3.6). The descriptions of perfSONAR, CMon and SQM clearly point out use of the well-known monitoring application Nagios as a way to make a composition of complementary functionalities.

The second characteristic of Section 3 reflects the current approach to implement new products. **Reusability** allows picking up a piece of the implementation and reusing it somewhere else. There is no need to “reinvent the wheel” or repeat work if readily available, high-quality solutions can be exploited. It is a matter of time and cost, especially in EU-funded projects, which should efficiently

spend the allocated funding. The PM&V solutions of this document apply to external components (e.g., SQM uses OWAMP; CMon uses SNMP libraries) or offer their parts to be reused (e.g. perfSONAR's eduSAFE elements can be reused by MaDDash and psUI).

Extensibility is crucial when making long-term plans. The five PM&V solutions are able to transform to become more and more advanced and powerful. Their architectures are ready to cope with new demands and emerging requirements. For example, The Wireless Crowdsource PM&V is built on the Service-Oriented Architecture (SOA), and new functionalities can be added as new micro-services. In case of CMon, modules are loosely coupled so a new addition does not affect other modules and can be easily integrated. Also, in SQM, extensibility can be achieved by using namespaces for adding new measurements.

Another very important characteristic of investigated PM&V solutions is **data management**. The GÉANT network is composed of many NRENs establishing federated network services, so the multi-domain nature of all project solutions, including PM&V, is a must. This is also a basic concept implemented in solutions presented in this deliverable. Unfortunately, multi-domain deployment has a cost, which is scalability. Highly distributed systems bring powerful functionalities but they often struggle with performance and management issues, for example, performance measurements on virtualised environments, computing elements, links, middle boxes, etc. Development teams of the described PM&V solutions considered such issues and addressed them by designing suitable architecture that creates efficient communication between lightweight components.

Nevertheless, investigation and work on **scalability and performance improvements** are ongoing. Last, but not least, **standardisation** regarding all the PM&V solutions of this deliverable deserves to be highlighted. In fact, it is connected with the previous characteristics, such as integration, reusability and extensibility. Use of standards enforces those other characteristics and helps to support the acceptance of the PM&V solutions by the wider networking community. All PM&V solutions apply standardised approaches (e.g. protocols like OWAMP, SNMP or WLAN standards).

4 Collaboration Solutions Using Existing Tools

The current toolset in PMV is an assortment of different applications providing features ranging from monitoring information to performance statistics, and from utilisation information to SLA parameters. It is this blend that makes up the complete architecture that provides solutions for many use cases, as shown in Section 2 of this document.

Of the many tools, some with complementing functionality may be brought together to provide a greater, more versatile functionality contained within its own ecosystem. It is possible that the requirements of some higher-layer services span 2 or more different toolsets, as offered in the PMV framework. In such cases, it is worth bringing such tools together to provide a singular application that fulfills service needs. A common denominator in facilitating collaboration/core functionality in two or more tools is their architecture, ability to interface and a shared functional area, which can provide a simplistic, singular application for a PM&V service.

The greatest convergence of functionality on used, existing tools (pS, CMon and SQM) results from the PMV solution assessment (see Section 3.7) and will be described in the next two subsections as:

- CMon and SQM
- pS and SQM.

4.1 CMon and SQM

This section explores the convergence of the aforementioned tools by evaluating the similarities in their architecture, and concludes with a recommendation of benefits of embarking upon such an exercise. Both tools are part of SA3 Task 3 (WiredMon) team, and the aim is to demonstrate that their convergence provides a wider set of functionalities/benefits for the end-user community, researchers and operations.

For further details about CMon see Section 2.2, and about SQM, see Section 2.4.

Service Quality Management (SQM) – is based on the operations process part of the eTOM model, to support processes for day-to-day customer and network operations and management [eTOM]. It was developed as a prototype to support Layer 2 and Layer 3 VPNs, as provisioned by the MD-VPN service. SQM provides verification and conformance to the SLA by collecting measurement parameters as per the agreement between different domains. An SQM-MA (Measurement Agent) is installed in each PE router in an NREN, which collects measurements and relays them to the central SQM controller (see Section 2.4.3).

SQM tailed information about its architecture diagram and performance metrics and management. It was developed as a prototype to support Layer 2 and Layer 3 VPNs as provisioned by MD-VPN serve the metrics thus collected. This results in better service delivery, which translates to success, and

greater uptake of service. Without a monitoring solution such as this, maintaining an SLA is not easy. SQM's multi-homing feature is useful in that it becomes scalable and can be extended to a wide range of measurement tools and protocols, which gives a high degree of flexibility.

Convergence – Convergence formation about its architecture CMon and SQM, and also their functionality in catering to multi-domain services, including the common subset of collecting performance measurements, makes them the ideal candidates for convergence and collaboration. It is worth mentioning that the tools should still be made available individually, since some use cases are not ideal for one of them. A separate fork, called C-SQM, short for CMon-SQM, should be created that is independent of them. The converged tool architecture should be such that any modified components, whether they belong to CMon or SQM, can be simply replaced or added to C-SQM without any significant effort to change the interface in existing modules.

Below is the deployment architecture diagram of the proposed C-SQM:

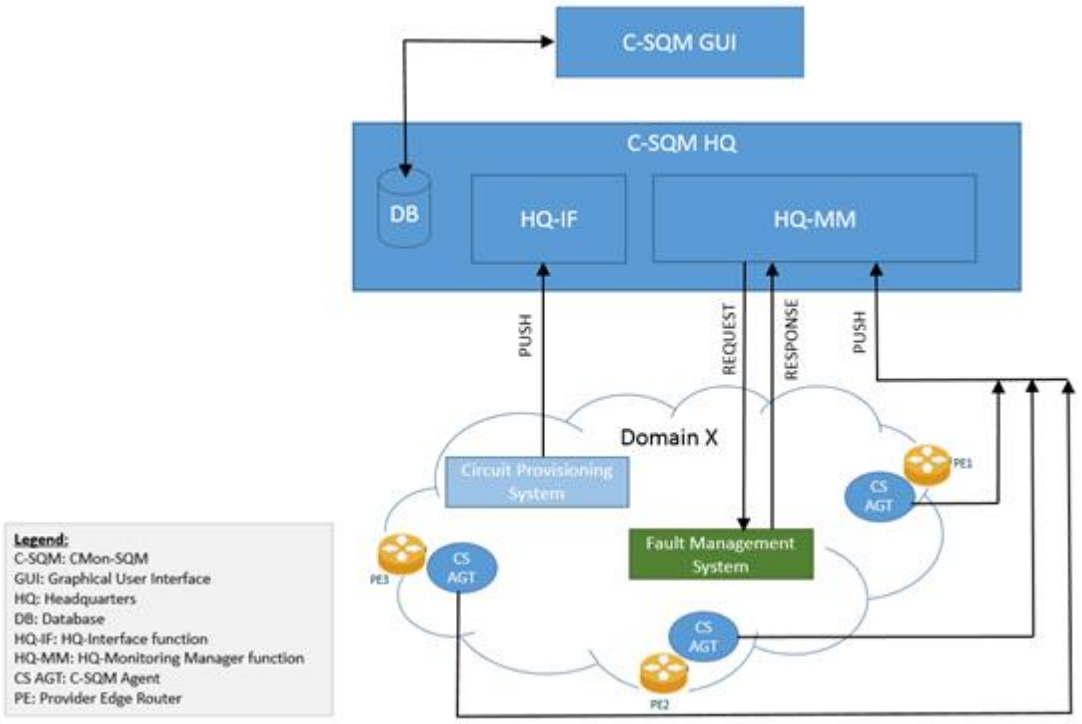

Figure 4.1: C-SQM (CMon, SQM) architecture

A typical domain such as this is an ideal candidate to deploy and use C-SQM, the converged CMon and SQM tool. Here is how the architecture is designed to work:

1. The domain typically has a Circuit Provisioning System (CPS) and Fault Management System (FMS). It also has a number of PE routers, above diagram shows three such routers.
2. C-SQM HQ is the brain of the C-SQM system. It is made up of the central database, an Interface Function module (HQ-IF), and the Monitoring Manager (HQ-MM).
3. A CPS, upon creation of an end-to-end circuit employs a PUSH mechanism to notify the HQ-IF module, an Interface Function of such an operation. Other domains too, not shown in Figure 4.1, with a similar deployment, also send notifications to HQ-IF regarding operations. This Push notification format should be uniform across CPSs. Upon receiving such notification, the HQ-IF parses the data to gather information about which domains are participating in the

circuit, the name of the circuit, allocated bandwidth, etc. This information can also be entered manually, which is currently the case for long-term static multi-domain circuits. CMon GUI provides this feature, which can easily be implemented for C-SQM as well.

This PUSH model also holds true for VPNs, in which case, the information can either come from three distinct sources: automatically via a provisioning system, manual entry via the C-SQM GUI as done for static circuits above, or imported from the domain's Service Inventory (SI), if available. This provides flexibility to the domain regarding the preferable source, thus demonstrating the versatility of HQ-IF module in the converged architecture.

At the time of writing this document, known VPN systems did not provide automatic VPN setup and teardown notification. While it is desirable that this is implemented, it is beyond the scope of this task. That said, if this changes in future, then the above C-SQM diagram will need to be slightly updated to show that HQ-IF can receive and process such automatic notifications from VPN provisioning systems.

4. The HQ-MM comes into play when the service, whether an end-to-end circuit or a VPN, is to be monitored. HQ-MM is notified by HQ-IF which domains are to be contacted for, thus to extract monitoring information. Then the HQ-MM sends a 'Request' to the domain's FMS to collect data, which is interface status, at the minimum. The FMS sends a 'Response' back to the HQ-MM, which is stored in the C-SQM HQ's database. This applies to all domains with C-SQM deployment.
5. At the same time, the CS-AGTs in a domain constantly send performance measurement data such as delay, jitter, etc. to the HQ-MM. Optionally, this function can also be periodic, and independent of any circuits or VPNs explicitly setup, as long as the data is available. HQ-MM stores this information in the central database so that it is available to be extracted when displaying data on the C-SQM GUI.
6. The C-SQM GUI connects to the central database in C-SQM HQ module, and extracts information to display regarding services the C-SQM system monitors, so that it is available via a single interface for users, and all information displayed in a consistent manner. An improvement would be to introduce REST API for communication between the GUI and HQ modules, rather than querying the database.

It is highly recommended that an effort is made in realization of this specific collaboration effort, as both tools collectively bring a feature set that is beneficial to many use cases as given in Section 2. Both end-to-end circuits and VPNs are multi-domain services, which are popular in the community for secure exchange of data within private networks or links. It is imperative that these multi-domain services are not only monitored, but also that service management information is shown in a manner that is consistent to the nature of these services. 'As many tools, as many services' only leads to a dissatisfaction among the user community, leading to the danger of domains adopting solutions within their own domain, meaning no end-to-end view or performance statistics, or worse, no monitoring at all.

The convergence can be extended to other tools, in order to contribute to PMV framework in a meaningful way. This process is also a positive step ahead in the direction of standardisation, so that the converged tools and even the PVM architecture/framework itself is interoperable with similar tools developed in R&E communities around the world.

4.2 perfSONAR and SQM

In the context of monitoring tools and performance verification architectures, the close relationship between perfSONAR (pS) and SQM becomes apparent. This provides incentives to explore possibilities for common deployment in scenarios for which these tools would cover a complete set of performance verification requirements. Identification of common elements and complementary functionalities is crucial to determine the deployment of performance monitoring and provision pS and SQM features in an efficient and scalable manner. Consequently, the benefits of integration become obvious, as it will produce a scalable performance verification solution that will provide features not present when individual tools are implemented, without having to deploy multiple and overlapping monitoring tools.

pS is a proven and widely deployed multi-domain monitoring solution with developed monitoring features allowing high scalability and flexibility along with advanced features for custom measurement scheduling and visualisation tools. Closer inspection of pS results in realisation that it is not suitable for deployment in various VPN scenarios, as it lacks Linux namespaces separation awareness and address-overlapping capabilities. Absence of these features may be addressed with the integration of SQM in pS.

Two notable potential approaches relate to pS and SQM integration. The first describes loose integration whereas the other one refers to tight integration. Both approaches assume that best properties are used from both tools providing measurement points (MP) capable of addressing what individual tools could not. For instance, advanced measurement scheduling and visualisation, test configuration and UI are highly developed in pS, whereas SQM MP has multi-homing, instance awareness and address overlapping capabilities. Placing all of these features in one MP achieves higher scalability, flexibility and easier deployment.

4.2.1 Loose integration approach

Loose integration basically assumes that pS and SQM MP coexist in a hardware box with minimal integration effort (Figure 4.2). Small efforts refer to the fact that both tools would basically keep their schedulers (SQM SI remotely triggers measurements, whereas local pS scheduler would trigger measurements according to the configuration) as well as underlying measurement tools, i.e. SQM and pS continue to operate with pure OWAMP and BWCTL, respectively. In terms of multi-homing, pS operates in the main Linux instance unaware of the existence of namespaces, whereas the OWAMP daemon is executed in separate namespaces for the purpose of SQM tests carried out in each instance.

The integration encompasses modification to storage, in which measurement results are archived so that service instance ID (value associated with measurement results originating from specific namespace) is relayed to the pS measurement archive (MA) via Nagios plugin, which is already being used for triggering SQM measurements. Consequently, certain modifications are required in the SQM SI and SQM controller components that store results in the Esmond database in order to distinguish measurement results from different instances [ESMOND]. Furthermore, pS UI modification is required if there is a need for visualisation of SQM measurement results in the pS UI. Visualisation modification may also include the capability to display data from different service instances, introduction of a dashboard(s) and the comparison of current results with predetermined SLA thresholds.

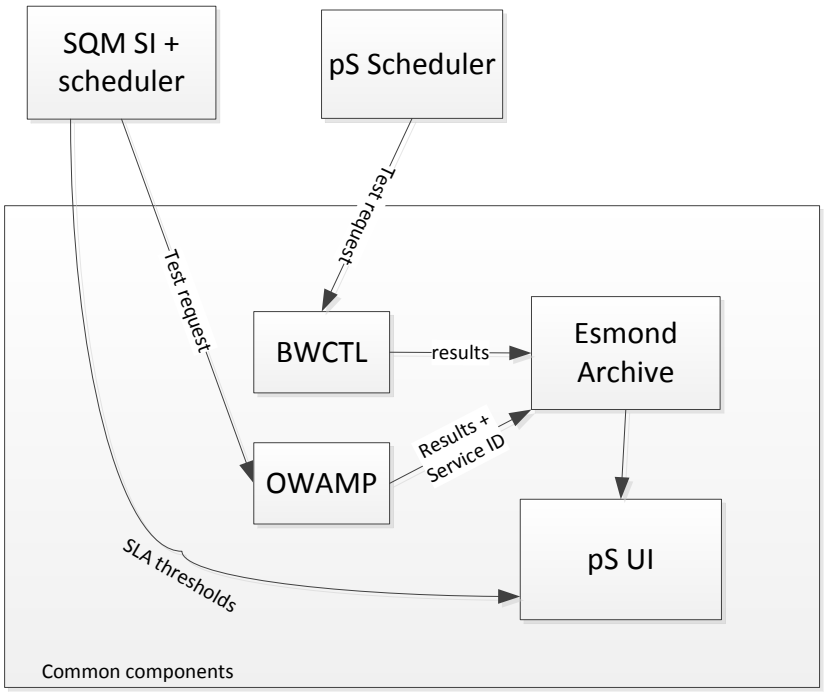

Figure 4.2: Loose integration of pS and SQM

The main advantage of this loose scenario is the minimal integration effort to address performance monitoring and verification requirements with a single hardware form factor. On the other hand, drawback of this approach is twofold. The first drawback may be reflected in the potential conflict of pS and SQM schedulers (e.g. two (2) measurements could happen at the same time, making measurement results invalid or less accurate) since the implementation assumes concurrent and independent operation of both tools without any correlation or control of scheduled tests. Although this may not be the problem with several namespaces, higher number of namespaces may result in operating problems. However, extension of this effect is yet to be determined with suitable scalability tests. The second drawback is from ad administrative perspective, referring to the simultaneous management of two tools, namely pS and SQM, where there would be two different monitoring tools intended for different services.

4.2.2 Tight Integration

In comparison to loose integration, a tight integration approach requires modifications of several elements in pS and SQM in order to have more compact integration and cooperation between the tools. Notably, the introduction of these changes improves pS multi-homing capabilities and allows it to be used simultaneously in service instances with overlapping address spaces. Furthermore, more compact integration is achieved where current pS features are extended with SQM functionalities, whereas from the management point of view there is only a single tool, namely pS, which is managed and used for different service requirements providing end-to-end performance monitoring and verification. A high-level overview of a tight integration approach is depicted in Figure 4.3.

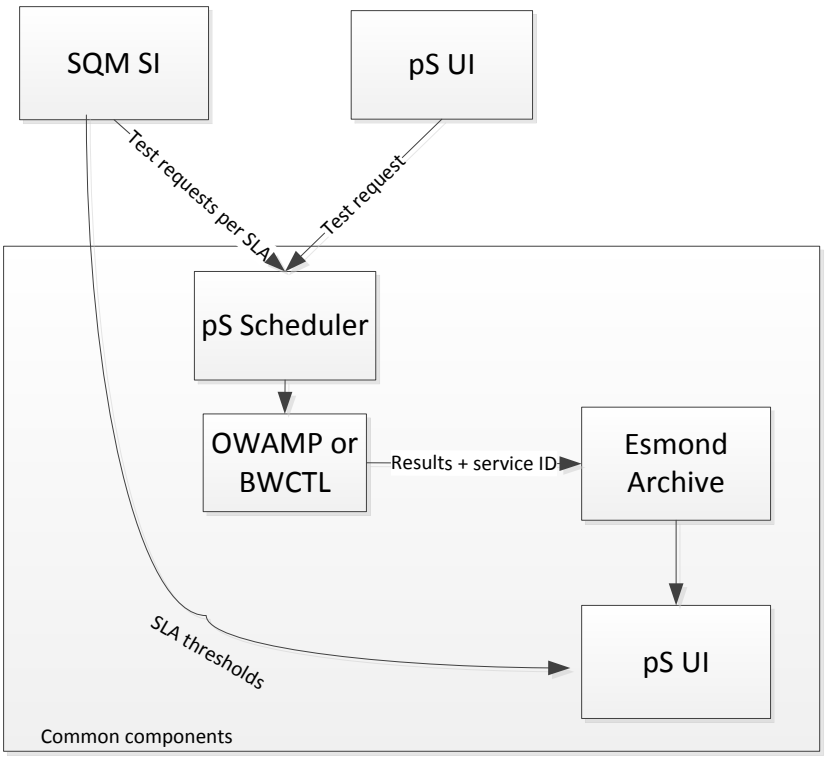

Figure 4.3: Tight integration of pS and SQM

Integration of common components and pS service instance awareness via namespaces presents a complex task that encompasses modification of several pS components in order to address namespace visibility. The following components require some level of modification effort in order to achieve a desirable level of integration:

1. BWCTL and pS regular-testing daemon (test scheduler): A new tool integrating functionalities of both current BWCTL (client and daemon) and regular-testing daemon is needed: a pS-scheduler. This tool would orchestrate all measurements and be aware of Linux namespaces, preventing invalid measurements by avoiding overlap. This tool would also need to know the relationship between VPN instances and Linux namespaces (the service ID) and would store this information in the Esmond database along the measurements.

- Mesh configuration management: Mesh configuration allows centralised management of regular tests on multiple pS nodes. In order to address tight integration, mesh configuration would need to be generated by the SQM SI following the perfSONAR mesh syntax/description. This syntax would need to be augmented to convey service ID and Linux namespace information. This information would then be used by the new tool described above (pS-scheduler) to account for specific namespaces from which measurements should be performed and to store the service ID into the Esmond database.

- psUI for visualisation: To distinguish measurements of different VPNs stored in the same database, changes would be required to the psUI tool in the same way as in the loose integration approach.

Previously described components require modifications in both the SQM SI and the perfSONAR tools. Most of the perfSONAR changes are being considered for the coming versions of perfSONAR (the new pS-scheduler is now a prototype). Previously mentioned approaches are mainly different in the modification effort in order to achieve a more compact and scalable performance monitoring solution, which is able to address most complex use cases.

5 Conclusion

The expertise on Performance Monitoring and Verification (PM&V) is based on the bottom-up approach demonstrated by the design and implementation of PM&V-proposals (see Section 2) to identify an adequate metric (Figure 1.1) and to follow-up the steps of the generic PM&V-process is essential for measuring a service landscape, the SLAs and OLAs.

The statement *Providing a network service without a proper performance measurement and their verification cannot be imagined these days* – is true. More than that, there is recognition of the need for "proper tool sets" in PM&V and validation of network assurance today, but also for future research and operations [NIF-P15_227].

Assessing the architecture of used tools (see section 3) shows how flexible, modular and standard based (e.g REST, SOAP, JSON etc.) the new tool-set must be. Only then it can be implemented as a part of any already existing heterogeneous platforms, on NMS at the (N)RENs.

The PM&V-solutions apply to external components (e.g. SQM uses OWAMP, CMon uses SNMP (traps) libraries) or offer their parts to be reused. Investigations in PM&V solutions, is a manner of data management that means the multi-domain nature is a must to support as many NRENs establishing federated network services as possible. Of course, efforts on scalability and performance improvements are work-in-progress, and individually, more or less a challenge. Furthermore, using standards that trigger integration, reusability and extensibility (see Section 3) enforces the other characteristics (e.g. scalability) so that PM&V solutions will be more acceptable to a wider network community.

The result from the assessment in Section 3 enables the integration of functionalities found in existing tools – such as CMon and SQM; pS and SQM (see Section 4). As there is no unique toolset/application available covering all end-user requirements and wishes (see NIF P15-227), the missing measurements of QoA and their visualisation, the missing multi-homing capability of tools on the Toolkit Layer (Figure 5.1) or new PM&V toolkits (Section 4) that is compatible with existing tools, services and use cases (Section 2) have been identified. A comprehensive architecture and software framework is offered that defines deterministic measurement parameters of services in a formal way that can then be automatically tracked in adequate and appropriate verification processes.

A generic PM&V architecture/framework has been determined, as depicted in Figure 5.1.

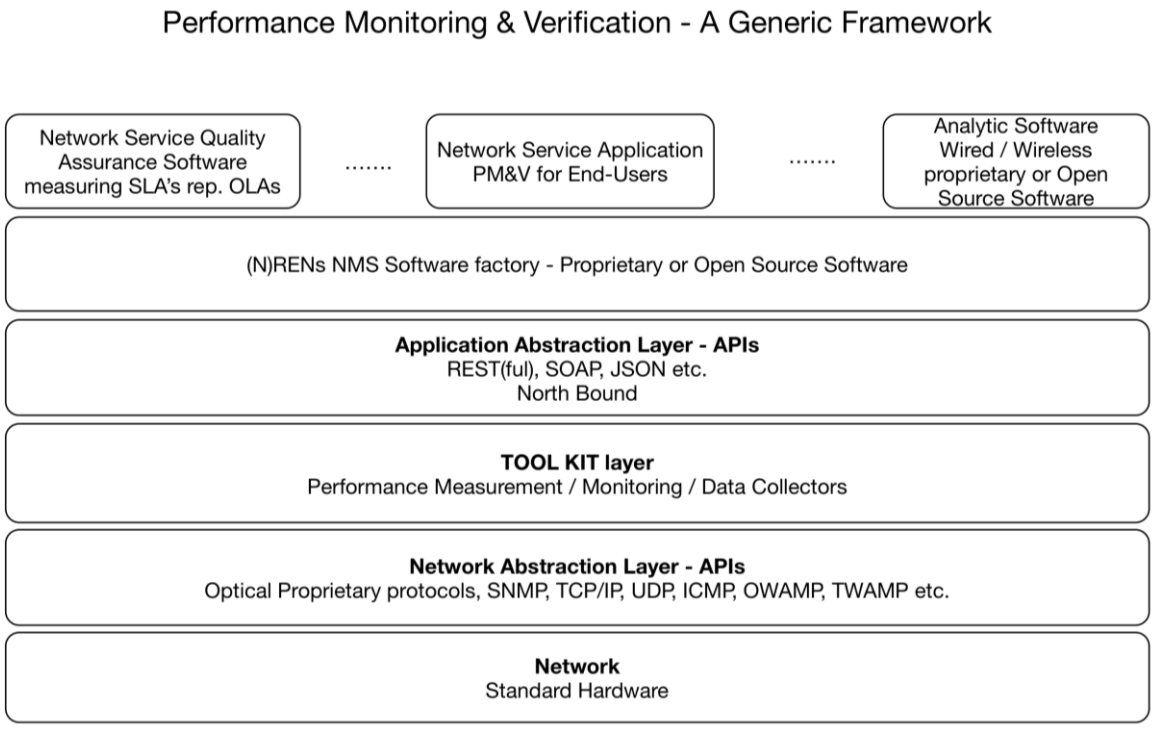

Figure 5.1: PMV generic architecture framework

The Tool-Kit Layer includes a software framework that must be based on automated processing to allow complex, multi-domain, multi-layered and multi-faced processing. This will include: our expertise, and it will be modular in order to facilitate the integration of diverse performance verification components that enables fast addition of new capabilities and adaption to emerging technologies (e.g. Cloud computing, SDN) and services (e.g. eduSAFE, BoD, MD-VPN, etc.). Three major steps are proposed, with focus on the next project phase:

- To figure out the real requirements of the (N)RENs' users (e.g. researchers, academic ICTs, and private research labs) is a must. Real requirements are the base of all investigations building new functionalities on existing tools or clean-slate approaches, for example, to reinvent SQM into the GÉANT service ecosystem.

- To develop based on these requirements, a formal description of (network) services and evolution of a PM&V software architecture/framework, according Figure 5.1.

- To build an automated framework/software factory for performance monitoring, measurements and verification, which is definitely missing by using the existing tools.

Regarding the results from Section 3 and evolution of new (collaboration) functionalities (see Section 4) Network providers (e.g. GEANT resp. the NRENs) will deliver needed tool-kits and APIs to the academia. This allows a lot of freedom for the (N)RENs community using proprietary, open source tools/NMS measuring, monitoring and verifying performance on their physical (virtualised) networks.

Appendix A CMon Future Scope

The work between CMon and GTS can be further enhanced in the future by working on the following issues, which also align with the GTS development plans.

- Monitoring VCs as provisioned by GTS can be made more robust, by providing the service at 'Reserve' state. This helps support users by the GTS infrastructure being aware that it needs to carry traffic, and whether it is capable of doing so, to its fullest capability. CMon can go one step further to report on monitoring status and performance data.
- CMon can also be enhanced to make the communication with GTS system two-way, for example, to notify the GTS system of VC unavailability or performance issues, and even interface with an alerting system and ticketing system. This will ensure monitoring through the complete state machine of the RCA-VC, which adds to the robustness of both GTS as a provisioning system and CMon as monitoring system.
- It is also possible for CMon system to collect monitoring metrics from a domain-specific router-based OS, for reliability and faster access to information. For instance, some of the metrics can be retrieved from REST API provided by JunOS, the OS in Juniper MXs, which are the current hardware used in GTS service available in GÉANT.

Appendix B Open Source Software

The following table of Open Source Software (OSS) lists the most important OSS used for the Deliverable's work, the link to the relevant website (if any), the link to the licence agreement (if any), and the point when OSS is used in the context of deployments in this document and remarks.

Name of OSS	Link Website/Licence Agreement (if any)	OSS used since	Remarks / Use of the software
perfSONAR	http://www.perfsonar.net/about/software-license/	GN3	A perfSONAR licence is distributed under an Apache 2.0 licence (both perfSONAR and OWAMP)
Debian	https://www.debian.org/social_contract	GN3 and earlier	There is no licence for the OS by itself as it is composed of thousands of different software that all have different licences. For Debian, used to install perfSONAR, refer to the following link: https://www.debian.org/social_contract
CentOS	https://www.centos.org/about/	GN3 and earlier	There is no licence for the OS by itself as it is composed of thousands of different software that all have different licences For CentOS used to install perfSONAR, refer to the following link: https://www.centos.org/legal/trademarks/#license-and-attribution
Web2Py Web2Py Licence	http://www.web2py.com/ http://www.web2py.com/init/default/license	GN3+	The Circuit Monitoring System CMon uses Web2Py for the webserver development. Web2py is Licensed under the LGPL license version 3 (http://www.gnu.org/licenses/lgpl.html).

Nettest	https://code.google.com/archive/p/nettest/ and Licence link: https://opensource.org/licenses/mit-license.php	GN4-1	Browser-based network measurement library, capable of determining throughput, latency, and other network parameters, using JavaScript and/or Flash
Boomerang	http://www.lognormal.com/boomerang/doc/ and Licence link: http://www.lognormal.com/boomerang/LICENSE.txt	GN4-1	Piece of javascript that you add to your web pages, where it measures the performance of your website from your end user's point of view (currently not used but plans to be included
Google API loader	https://developers.google.com/loader/ and Licence link: http://creativecommons.org/licenses/by/3.0/	GN4-1	Used for IP-based geolocation
PostgreSQL	http://www.postgresql.org/ and Licence link: https://opensource.org/licenses/postgresql	GN4-1	Database used to store nettest measurements Database used to store FreeRadius server logs
InfluxDB	http://influxdata.com/ and Licence link: https://opensource.org/licenses/mit-license.php	GN4-1	Database used to store timeseries with nettest measurements
Grafana	http://grafana.org/ and Licence link: http://www.apache.org/licenses/LICENSE-2.0	GN4-1	Used for visualising time series with nettest measurements and correlation results

Table B.1: List of Open Source Software (OSS)

References

[APAN40]	https://www.apan.net/meetings/KualaLumpur2015/NRW2015.php
[AutoBAHN]	https://forge.geant.net/forge/display/autobahn/Home
[AWS]	AWS deployment: https://www.terena.org/activities/netarch/ws1/slides/lund-alienwave-201112-JRA1.pdf
[BEAGLE_BOARD]	https://beagleboard.org/
[CMon]	https://forge.geant.net/forge/display/cmon/Home
[CMon_INSTALL]	CMon Installation Guide: https://forge.geant.net/forge/download/attachments/6979757/CMon+-+Installation+Guide.pdf
[CMon_MEAS]	http://cmongui.qalab.geant.net/ (Ordinary users can register to log in)
[CONFLUENCE]	MEASUREMENT:https://confluence.geant.net/confluence/display/eduSAFE/Hardware+resources+and+services
[eduPERT]	https://drive.google.com/file/d/0B4QzlQzRZg9yd3lQWElKZzJWVUE/view?usp=sharing
[CUBOX]	https://www.solid-run.com/freescale-imx6-family/cubox-i/cubox-i-documentation-block-diagram/
[D6.2]	http://geant3plus.archive.geant.net/Resources/Deliverables/Documents/D6-2_D2-3-2_TaaS_v2%200.pdf
[eduPERT]	http://services.geant.net/edupert/Pages/Home.aspx
[eduSAFE]	https://confluence.geant.net/confluence/display/eduSAFE/eduSAFE
[ESMOND]	http://software.es.net/esmond/index.html and DB http://software.es.net/esmond/deployment_cookbook.html
[eTOM]	https://de.wikipedia.org/wiki/Enhanced_Telecom_Operations_Map
[FCAPS]	https://en.wikipedia.org/wiki/FCAPS
[GÉANTIP]	http://www.geant.org/Services/Connectivity_and_network/Pages/GEANT_IP.aspx
[GÉANTPlus]	http://www.geant.org/Services/Connectivity_and_network/PublishingImages/Pages/GEANT_Point-to-Point/GEANT%20Plus%20Service%20Description%20Jul%202015.pdf
[GTS]	http://www.geant.net/Resources/Deliverables/Documents/D6-2_D2-3-2_TaaS_v2%200.pdf
[LIGHTNING]	https://geant.23video.com/tnc15-8a-thunderclap-talks
[M7.1]	Milestone Document M7.1 *Wireless Crowd Source PM&V* available to GN4-1 Project Participants on the Project site https://intranet.geant.org/gn4/1/Activities/SA3/Milestones%20Documents/Forms/AllItems.aspx

	https://intranet.geant.org/gn4/1/Activities/SA3/T3/Documents/SA3T3_WirelessMon/ML7_1%20Milestone%20Document/M7-1_Wireless-Crowd-Source-Performance-Measurement-and-Verification_clean.docx?Web=1
[M7.2]	Wireless Crowd Source Performance Monitoring and Verification: https://drive.google.com/file/d/0B4QzlQzRZg9yNmFXNFYzbEZGMEk/view?usp=sharing
[MaDDash]	http://software.es.net/maddash/
[MD_VPN]	http://geant3plus.archive.geant.net/Resources/Deliverables/Documents/D7.1_DS%203%203%201-MDVPN-service-architecture.pdf
[MongoDB]	https://www.mongodb.org/
[NETTEST]	https://code.google.com/archive/p/nettest/
[NIF-P15_227]	(available to GN4-1 Project Participants on the Project site) https://intranet.geant.net/ProjectOffice/Document%20Library/P15_227/NIF_Global%20Performance%20Monitoring%20and%20Verification%20Framework.doc?Web=1
[OpenNSA]	https://github.com/NORDUnet/opennsa
[perfSONAR]	http://www.perfsonar.net/
[RFC-4656]	https://tools.ietf.org/html/rfc4656
[SQM]	https://drive.google.com/open?id=0B4QzlQzRZg9yZ0k1YmotRk1Fd3M
[SQM_DASHBOARD]	https://svn.geant.net/fisheye/browse/service-quality-management-system/branches/SQM-prototype/service-quality-management (authN is needed)
[SWITCHENGINES]	https://www.switch.ch/engines/
[TNC2015]	TNC2015: https://tnc15.terena.org/
[TNC_MEASURE]	TNC2015 measurements: https://vm3-gn3-sa2t5.vm.grnet.gr/nettest-0.9a/tnc15_correlation.php
[WCSPMV]	Dashboard/Guide:https://62.217.125.88:8443/login (authN is needed) - and simulation test resource http://62.217.125.88/wifimon/measurement.php

Glossary

AP	Access Point
API	Application Program Interface
AW	Alien Wave
AWS	Alien Wavelength Service
BER	Bit Error Rate
BWCTL	Bandwidth Test Controller (A policy daemon that allows the user to invoke a series of other tools (iperf, Ping) via a set of commands)
C-SQM	CMon-SQM
CPS	Circuit Provisioning System
CPU	Central Processing Unit
CS AGT	CS Agent
CsC	Carrier-supporting-carrier
FCAPS	Fault-management, Configuration, Accounting, Performance, and Security
FEC	Forward Error Correction
FMS	Fault Management System
GTS	GÉANT Testbed Service
GUI	Graphic User Interface
HQ	Headquarters
HQ-IF	Headquarters Interface Function
HQ-MM	Headquarters Monitoring Manager
HW	Hardware
ICMP	Internet Control Message Protocol
iperf	Network performance measurement tool written in C
IP	Internet Protocol
IPPM	IETF IP Performance Metrics
JMS	Java Messaging System
JSON	JavaScript Object Notation
KOPI	Key Operational Performance Indicators
L2	Level 2
MA	Measurement Archive
MP	Measurement Point
MD-VPN	Multi-Domain Virtual Private Network
NAT	Network Address Translation
NDT	Network Diagnostic Tool
NIC	Network Interface Controller
NIF	New Ideas Form
NMS	Network Management System
NOC	Network Operation Centre

NREN	National Research and Education Network
NTP	Network Time Protocol
OAM	Operations, Administration and Maintenance
OLA	Operational Level Agreements
OpenNSA	Open Source implementation of Network Service Interface (NSI) protocol (manages virtual circuit provisioning)
Ops	Operations
OSNR	Optical Signal-to-Noise
OSS	Operations Support System, Open Source Software
OTN	Optical Transport Network
OWAMP	One-Way Active Measurement Protocol
PE	Provider Edge
PM&V	Performance Monitoring and Verification
pS	perfSONAR
psUI	perfSONAR web User Interface
QoS	Quality of Service
RA VPN	Remote Access Virtual Private Network
RCA-VC	Resource Control Agent ring dual Circuit
REST	Representational State Transfer
RRD	RoundRobinDatabase
SA2, T3	Service Activity 2 Testbeds, Task 3 Testbeds Services Management
SA3, T3	Service Activity 3 Network Delivery and Support, Task 3 Multi-Domain Monitoring
SC	Service Consumers
SI	Service Inventory
SLA	Service Level Agreement
SNMP	Simple Network Management Protocol
SO	Service Owners
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
SQM	Service Quality Management
SQM-MA	Service Quality Management Measurement Agent
SW	Software
VC	Virtual Circuit
VM	Virtual Machine
VPN	Virtual Private Network
WCSPM&V	Wireless Crowd Source Performance Measurement and Verification