AutoBAHN and perfSONAR Integration

Circuit Monitoring

Actual Date: 24-03-2011
Activity: JRA2
Task Item: T3
Authors: Jakub Gutkowski (editor), Roman Łapacz, Jacek Łukasik

Version: 0.21.2

Table of Contents

1 Introduction

This document provides a high-level overview of the perfSONAR [perfSONAR] and AutoBAHN [AutoBAHN] integration by presenting the architecture and workflow for monitoring a multi-domain circuit. Due to perfSONAR's and AutoBAHN's distributed architectures it is feasible to perform monitoring across various domains, where the underlying transport technology may vary on different path[1] segments[2].

During GN2 [GN2] prototype circuit monitoring functionality was introduced and tested with AutoBAHN. The prototype was tightly integrated with the AutoBAHN system and has not further evolved. One of the main goals of GN3 [GN3] is the use of available systems/tools as modules for offering complex functionality such as the monitoring of dynamically provisioned circuits. In this framework, perfSONAR is considered appropriate to support AutoBAHN in collecting and managing the monitoring data of the circuits.

There are two kinds of consumers of monitoring data:

- **Internal consumer**

  The Domain Manager (a key component of AutoBAHN) uses the monitoring data to maintain the operational status and traffic characteristics of the path segment within the domain. The complete topology and corresponding monitoring data are available to the Domain Manager without any restrictions (**Intra-Domain View**).

- **External consumer**

  The external consumer is a GUI that can only access the aggregated (abstract) topology and corresponding aggregated monitoring data (**Inter-Domain View**) of segments or the entire path. The internal topology or detailed monitoring data within a domain may not be visible to the client.

Some solutions and terminology presented in this document are partially inspired by the efforts of US partners (led by Internet2) who are working on the integration of perfSONAR with their dynamic circuit provisioning system. More information can be found here:

http://code.google.com/p/perfsonar-ps/wiki/CircuitMonitoring[3].

[1] Path – a sequence of links configured to provide connectivity between end points. The term "path" is being used exchangeably with "circuit".
[2] Segment – a part of a path under administration of a single domain (one domain = one segment).
[3] The discussion about general circuit monitoring model is being proceed by US and GN3 memebers within DICE group [DICE]

2 Integration Requirements

- The specification describes the AutoBAHN and perfSONAR integration (the proposed solution may support other provisioning systems but mainly it should be prepared for AutoBAHN).
- Two types of topology information must be supported: real internal domain topology (Intra-Domain View) and aggregated (abstract) topology (Inter-Domain View) which does not reveal real intra-domain settings.
- The specification should support the Global Identifiers Naming Schema provided by the GLIF [GLIF].
- The AutoBAHN system provides monitoring data aggregation and topology aggregation (Inter-domain View). The aggregation is located and offered by the AutoBAHN component – Domain Manager (DM).

3 Resources Involved in the Integration

3.1 Operator (Client) Site

- **RC** – Reservation Client

  A client application, which requests dynamic circuits (AutoBAHN Client Portal [ABClientPortal], or another application which accesses AutoBAHN functionality through the AutoBAHN API [ABAPI]).

- **VC** – Visualisation Client

  A client application which presents the monitoring information stored in the perfSONAR system for a circuit.

3.2 AutoBAHN (Circuit Provisioning System)

- **IDM** – Inter-Domain Manager [IDM]

  The IDM provides inter-domain-technology-agnostic functionality. It receives and processes a high-level BoD reservation requests from users and other IDMs. It operates on abstract topology views

- **DM** – Domain Manager

  The element of the AutoBAHN architecture that is responsible for intra-domain functionality. It receives the BoD reservation requests from the IDM, searches for a proper path within the domain and handles signalling to/from the data plane. It is responsible for recording the resources' utilisation information.

3.3 perfSONAR (Network Monitoring System)

- **MA** – Measurement Archive

  A service that stores and publishes measurement data.

- **MP** – Measurement Point

  A service that executes the measurements (passive or active) or fetches measurement data directly from a network device (e.g. via SNMP).

- **LS** – Lookup Service

  The Lookup Service (LS) collects information about the operational perfSONAR infrastructure. Each service periodically sends registration request to the LS, so it is listed as operational to client entities (e.g. GUI client applications or a service which needs a functionality offered by another service). To distribute registration functionality (multi-domain nature) in a scalable manner two types of LS have been defined: home LS (hLS) and global LS (gLS). The former is responsible for collecting registration information directly from perfSONAR services (MAs, MPs, etc.). The latter fetches aggregated registration information from the hLS service(s) and shares them with other domains (each domain may have its own hLS for local services and gLS to share the information with other domains). Those two types of LS provide access to services and data located in all domains running the perfSONAR system.

- **MA(ad)** – Measurement Archive for aggregated (abstract) data

  The MA service that stores and publishes aggregated measurement data for visualisation (Inter-Domain View). The MA and the MA(ad) may use the same implementation but the database must be prepared to store both original and aggregated data.

- **MA(at)**[4] – Mesurement Archive for inter-domain aggregated (abstract) topology

  The MA service that stores and publishes aggregated (abstract) topology (Inter-Domain View).

- **MA(t)** – Mesurement Archive for intra-domain topology

  The MA service that stores and publishes real intra-domain topology (Intra-Domain View).

Additional information about perfSONAR services – GN3 MDM implementation [MDM] – can be found in the Admin Guide prepared for the 3.2 release [MDMAdminGuide].

[4] Both services, the MA(at) and the MA(t), offer the functionality which is represented by the Topology Service in the pS-PS [perfSONAR-PS] proposal. MA(at) and MA(t) may be implemented as one application offering both topology types.

3.4 Integration Component

- **SIP** – System Integration Point

  A component that is responsible for the communication between circuit provisioning and network monitoring systems. It manages circuit monitoring, contacts MP services to run measurements and determines where the monitoring data should be stored (selection of MA services).

4 Circuit Description

The multi-domain circuit provisioned by AutoBAHN can be described by the following informational objects:

- **Reservation Descriptor** – an entity identified by a unique[5] Reservation Identifier (used by the AutoBAHN system to track the reservation in all involved domains). It contains a sequence of Segment Identifiers (identifiers which refer to the paths inside the domains).

  The original AutoBAHN Reservation Identifier has the following format:

  `<STARTING_DOMAIN_ID>@<TIMESTAMP>_res_<SEQ_NUM[6]>`

  To support the GLIF format the original AutoBAHN Reservation Identifiers must be transformed into the following form:

  `urn:ogf:network:<STARTING_DOMAIN>:<STARTING_DOMAIN_ID>@<TIMESTAMP>_res_<SEQ_NUM>`

- **Segment Descriptor** – an entity identified by a unique Segment Identifier (used by the AutoBAHN system to track the part of a circuit within a domain). It contains a sequence of network elements selected for the reservation within a domain. This descriptor represents the Intra-Domain View (real internal domain topology)

  The original AutoBAHN Segment Identifier has the following format:

  `<STARTING_DOMAIN_ID>@<TIMESTAMP>_res_<SEQ_NUM>:<DOMAIN_ID>`

  To support the GLIF format the original AutoBAHN Segment Identifiers must be transformed into the following form:

  `urn:ogf:network:<DOMAIN>:<STARTING_DOMAIN_ID>@<TIMESTAMP>_res_<SEQ_NUM>:<DOMAIN_ID>`

[5] The globally unique reservation identifier is calculated by concatenating the identifier of the starting domain (which is unique in the AutoBAHN domains federation) and a timestamp of the moment when the reservation is accepted by the system. Queuing and synchronisation mechanisms implemented in AutoBAHN ensure that only one reservation request can be accepted at any one time.

[6] In the AutoBAHN system it is possible to request several reservations simultaneously. The group of reservations forms a single service. All the reservations within the service must succeed. If any of the reservations is not feasible, all the reservations that were scheduled before the failed reservation are rolled back. SEQ_NUM is the sequence number of a reservation within a service.

- **Aggregated Segment Descriptor** – an entity identified by a unique Segment Identifier. It contains aggrageted part of the circuit so that it does not reveal real internal domain toplogy (Inter-Domain View). The type of aggregation is defined by a domain.

Figure 1 presents the examples of original AutoBAHN reservation and segment identifiers for a reservation traversing the domains: *PIONIER*, *GEANT* and *HEANET*. *PIONIER* is the **starting domain** of the circuit. The IDM in this domain assigns the identifier *PIONIER@1275987204_res_1* to the circuit. The segment identifiers are: *PIONIER@1275987204_res_1:PIONIER, PIONIER@1275987204_res_1:GEANT* and *PIONIER@1275987204_res_1:HEANET.*

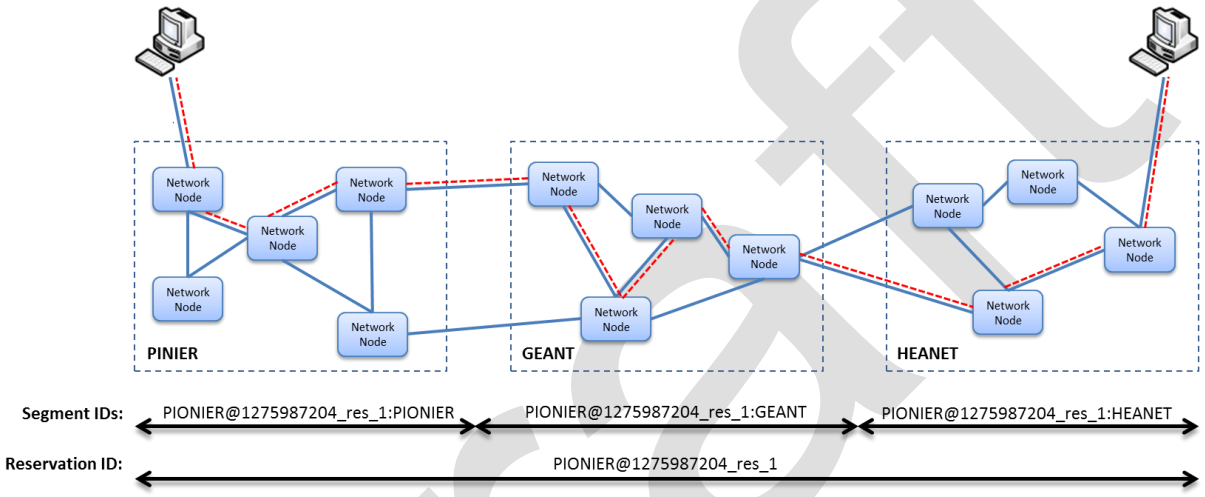

Figure 1 The AutoBAHN Reservation and Segment identifiers for a sample circuit traversing three domains.

Table 1 presents mapping between original AutoBAHN identifiers and identifiers supporting the GLIF format.

Original AutoBAHN identifiers	Identifiers supporting GLIF format
PIONIER@1275987204_res_1	urn:ogf:network:pionier.net:PIONIER@1275987204_res_1
PIONIER@1275987204_res_1:PIONIER	urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER
PIONIER@1275987204_res_1:GEANT	urn:ogf:network:geant.net:PIONIER@1275987204_res_1:GEANT
PIONIER@1275987204_res_1:HEANET	urn:ogf:network:heanet.net:PIONIER@1275987204_res_1:HEANET

Table 1 Mapping between original AutoBAHN identifiers and identifiers supporting GLIF format.

5 Circuit Monitoring

5.1 Functionality Decomposition of the Integration

Figure 2 presents identified functionalities of the integration and relations between them.

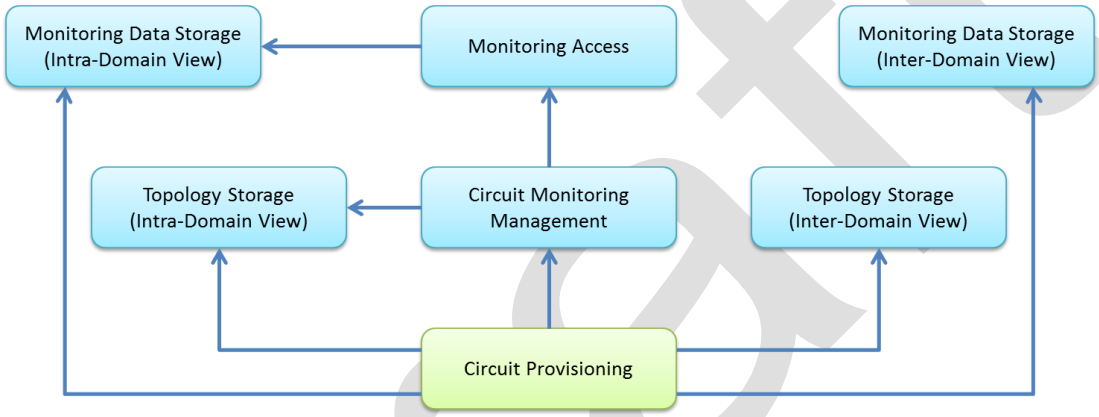

Figure 2 Functional view of the integration.

- Circuit Provisioning – setting up a dynamic circuit.
- Circuit Monitoring Management – recording, organising and exchanging the information required for circuit monitoring.
- Topology Storage (Intra-Domain View) – storage of real intra-domain topology.
- Topology Storage (Inter-Domain View) – storage of aggregated (abstract) topology.
- Monitoring Access – access to network monitoring system/resources (active or passive measurements).
- Monitoring Data Storage (Intra-Domain View) – storage of real (not aggregated) measurement data collected in a single domain.
- Monitoring Data Storage (Inter-Domain View) – storage of aggregated (abstract) measurement data.

Table 2 presents mapping between functionalities of the integration and the services existing in the architecture.

Functionality	Services
Circuit Provisioning	• Inter-Domain Manager - IDM • Domain Manager - DM
Circuit Monitoring Management	• System Integration Point - SIP
Topology Storage (Intra-Domain View)	• Measurement Archive for intra-domain topology - MA(t)
Topology Storage (Inter-Domain View)	• Mesurement Archive for inter-domain aggregated (abstract) topology - MA(at)
Monitoring Access	• Measurement Point - MP
Monitoring Data Storage (Intra-Domain View)	• Mesurement Archive - MA
Monitoring Data Storage (Inter-Domain View)	• Measurement Archive for aggregated (abstract) data - MA(ad)

Table 2 Mapping between functionalities and services

5.2 Setting up Circuit Monitoring

5.2.1 Workflow in a Single Domain

Figure 3 presents a detailed workflow between the components of the AutoBAHN and perfSONAR integration for a single domain.

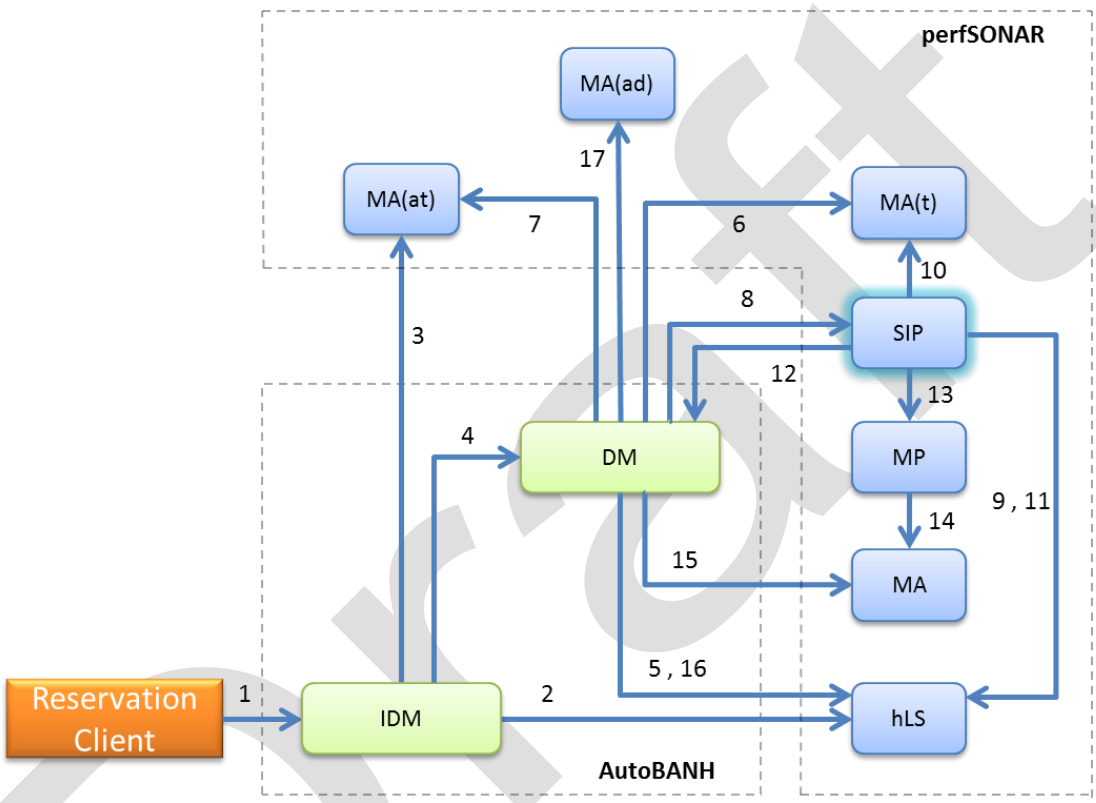

Figure 3 The AutoBAHN and perfSONAR integration workflow (single domain).

Init phase

Step 0: All services (SIP, MPs, MAs) register with the hLS.

Circuit reservation phase

Step 1: The Reservation Client sends a dynamic circuit reservation request to the IDM and obtains the Reservation Identifier.

Step 2: The IDM in the first domain (starting domain) of the circuit requests the MA(at) address from the hLS.

Step 3: The IDM creates and stores the Reservation Descriptor (with the Reservation Identifier) in the MA(at).

Step 4: The IDM requests the DM to reserve resources for the circuit within the domain.

Monitoring management phase

Step 5: The DM obtains the address of the SIP, the MA(t) and the MA(at) from the hLS (this step is triggered when the reservation is activated; getting the addresses does not have to be immediate because the reservation may be scheduled with some delay).

Step 6: The DM creates and stores the Segment Descriptor (with the Segment Identifier) in the MA(t).

Step 7: The DM aggregates the topology contained in the Segment Descriptor and stores the aggregated Segment Descriptor in the MA(at).

Step 8: The DM provides the SIP with the Segment Identifier and requests to conduct monitoring of the intra-domain circuit segment (the SIP records a monitoring session for the reservation).

Step 9: The SIP obtains the MA(t) address from the hLS (this action may also be executed right after step 0).

Step 10: The SIP uses the Segment Identifier to fetch the Segment Descriptor from the MA(t).

Step 11: The SIP contacts the hLS and obtains information about the MAs and the MPs that are available in the domain and relevant to the monitored circuit and its parameters (searching the MAs and MPs is based on the information included in the Segment Descriptor).

Step 12: The SIP informs the DM about the MA service(s) found in the previous step.

Step 13: The SIP requests the relevant MPs to perform measurements and to store the results in selected MAs (one of the options - see section 5.2.3).

Step 14: The MPs perform monitoring tasks and store the results in the relevant MAs (one of the options - see section 5.2.3).

Step 15: The DM periodically contacts the MAs and retrieves the latest measurement data (one of the options - see section 5.2.3). The DM verifies the traffic parameter(s) (metric(s)) of the circuit to check if the reservation requirements are accomplished (if not, an appropriate action can be triggered inside the AutoBAHN system).

Step 16: The DM obtains the address of the MA(ad) from the hLS (this action may also be executed right after step 0).

Step 17: The DM performs monitoring data aggregation so that the aggregated data corresponds to the abstract topology (Inter-Domain View) and stores the data in the MA(ad). From then the MA(ad) registers stored data with the hLS.

5.2.2 Workflow in the multi-domain environment

Figure 4 presents a workflow of setting up a circuit (AutoBAHN) with the enabled monitoring functionality (perfSONAR) in the multi-domain environment.

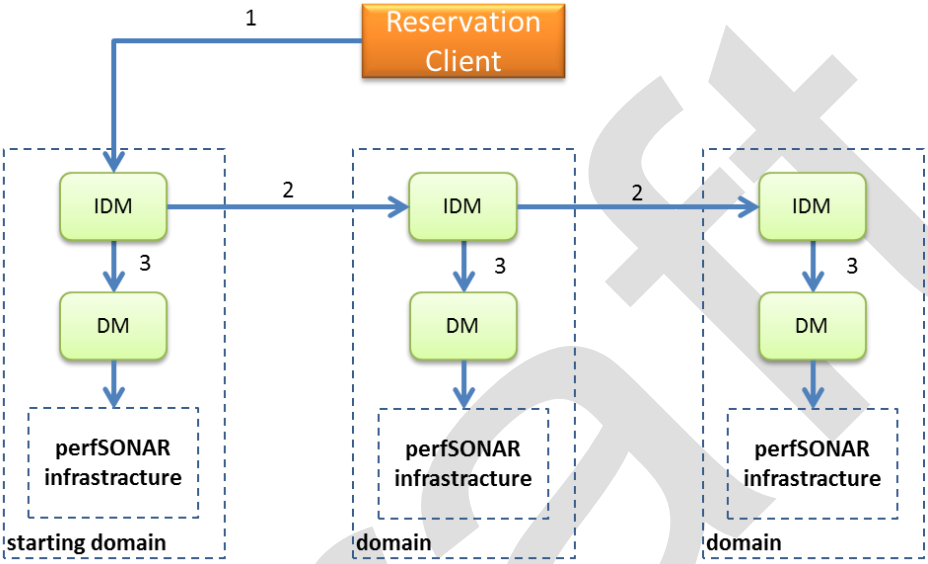

Figure 4 The AutoBAHN and perfSONAR integration workflow (multi-domain view).

Step 1: The Reservation Client sends a dynamic circuit reservation request to the IDM (in the starting domain) and obtains the Reservation Identifier.

Step 2: The IDM in the starting domain initiates multi-domain circuit provisioning.

Step 3: An IDM in each domain executes the path (segment) creaction with monitoring functionality enabled - next steps inside a doman are presented in section 5.2.1.

5.2.3 Requesting the Measurements and Fetching the Data

Use cases:

- The SIP contacts the MP to initiate measurements and instructs it where to store results (an appropriate MA is chosen). Next the DM obtains data from the MA.

- Measurement data is fetched from the MA by the DM (the SIP does not communicate with the MP to initiate the measurements because the MA is populated with data constantly; e.g. when the MRTG [MRTG] tool populates RRD files which are then used by the RRD MA perfSONAR service).

- The SIP contacts the MP to initiate the measurements. The DM obtains data directly from the MP (the MP has the MA functionality and the MA interface).

Measurement requests to the services (MP and MA) are being sent periodically. The polling interval is defined by the DM.

5.2.4 Links Connecting Domains

Two use cases can be considered for monitoring the links which terminate in the neighbouring domains:

- An MP service is allowed to fetch the monitoring data from devices in both domains, so the link status between them can be calculated by the service.

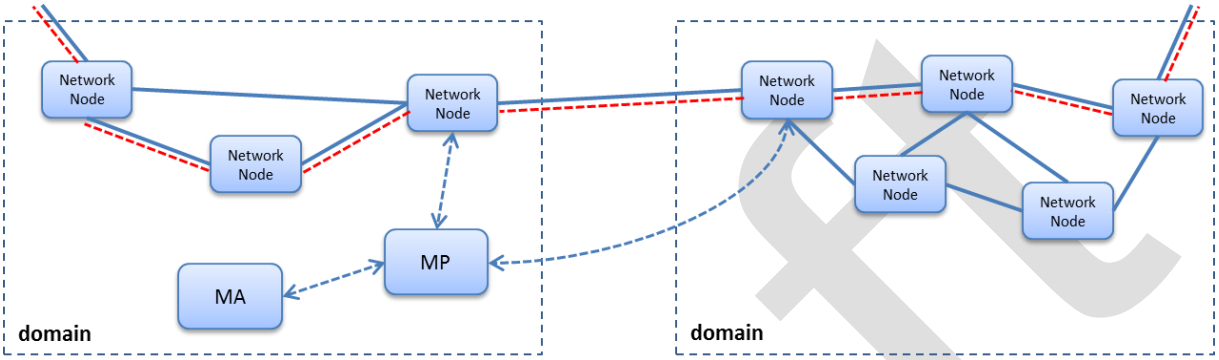

Figure 9 Simple access to the information about the inter-domain link

This use case will not be popular because usually separate domains do not give the MP access to internal resources.

- An MP service is only allowed to fetch the monitoring data from a device in its own domain, so the link status cannot be calculated by the service. Only the port status is known and can be stored in the MA. The link status could be calculated by a component (e.g. VC application) which has access to the MAs in both domains containing the both ports states.

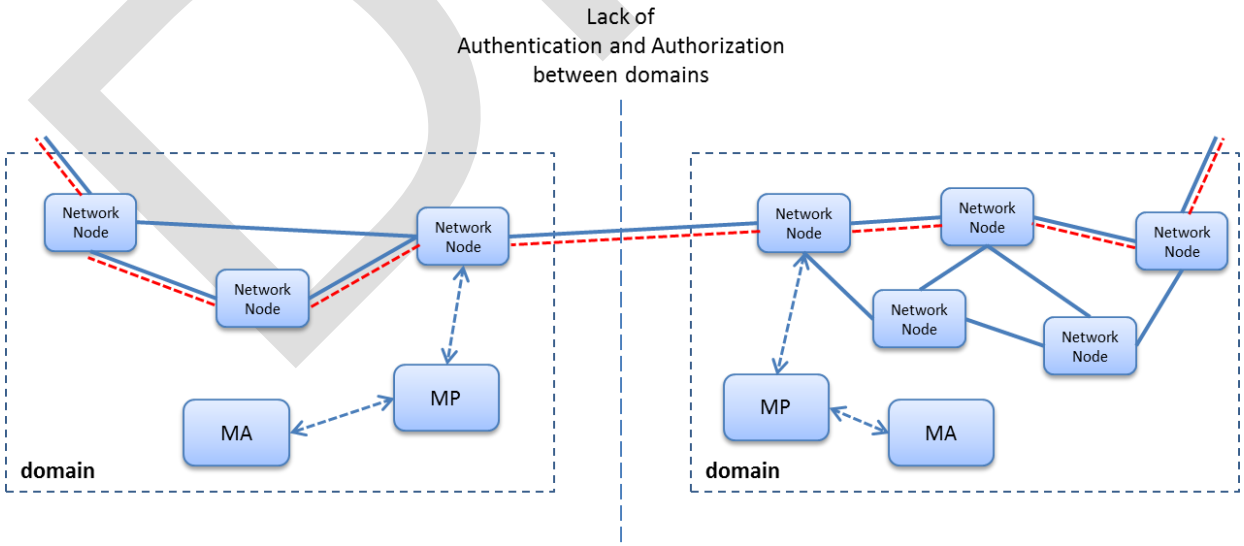

Figure 10 Limited access to inter-domain link information

5.2.5 Domains without Monitoring

If the circuit crosses a domain that does not support perfSONAR network monitoring, the domain is treated as a blank area, and the corresponding segment does not have monitoring information.

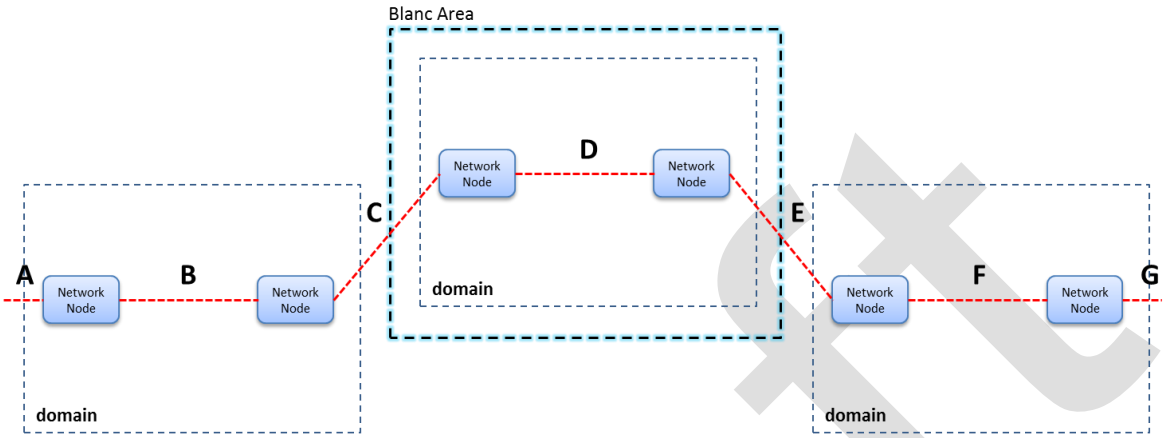

Figure 11 A circuit crossing a domain which does not support monitoring

Abstract link	Stauts
A	Up
B	Up
C	-
D	-
E	-
F	Up
G	Down

Table 3 Link states with blank areas (see Figure 11)

5.3 Access to the monitoring data

5.3.1 Internal consumer

Figure 5 presents the use case where the DM utilises the monitoring data to maintain the operational status and traffic characteristics of the path (part of the circuit) inside the domain. The DM periodically contacts the MA(s) and retrieves the latest measurement data - see section 5.2.1 step 15. The measurement data and topology are not aggregated so they represent the Intra-Domain View.

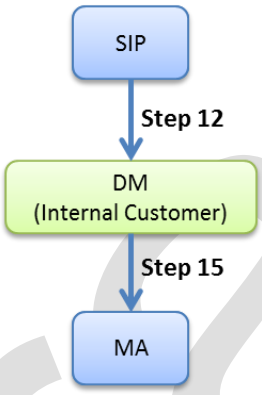

Figure 5 The Internal consumer use case.

5.3.2 External consumer

Figure 6 presents the workflow supporting the **General Standardised Client Access** (see 9.2.3Appendix B). This procedure makes the client access to the measurement data compatible with the PS-pS implementation provided by US partners (led by Internet2). In order to simplify the interactions and configuration complexity it is assumed that each type of service is represented by a single instance (this approach eliminates some interaactions listed in the General Standardised Client Access but the complete use of it is still supported).

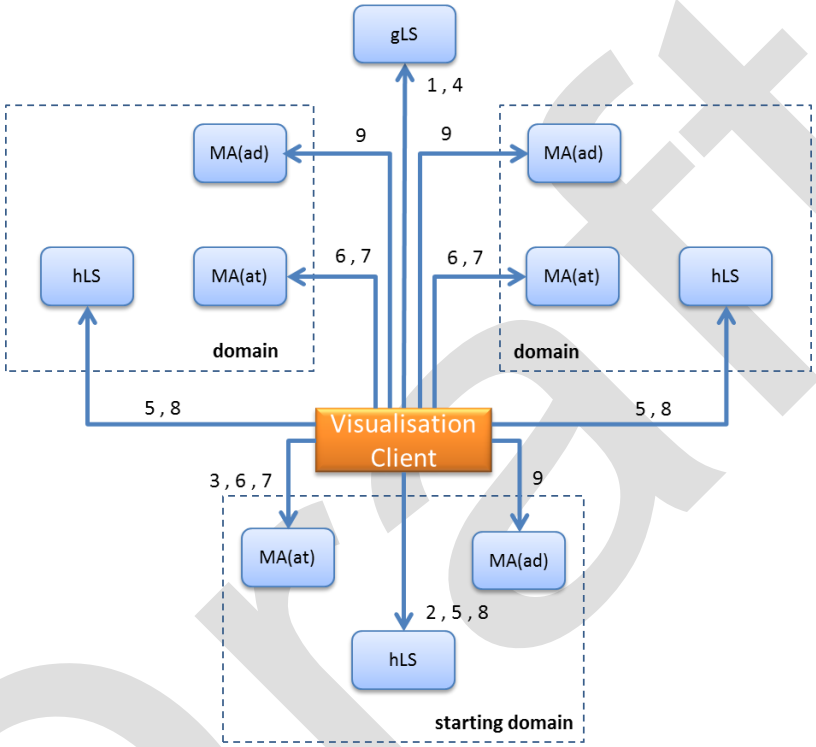

Figure 6 The External consumer use case.

Step 1: The Visualisation Client (VC) obtains the hLS address (starting domain) from the gLS using the domain name existing in the Reservation Identifier (see the format of identifiers in section 4).

Step 2: The VC requests the MA(at) address from the hLS.

Step 3: The VC obtains the Reservation Descriptor(s) (the list of Segment Identifiers) from the MA(at) service.

Step 4: The VC requests the hLS address(es) from the gLS using the domain name(s) existing in the Segment Identifier(s).

Step 5: The VC requests the MA(at) address containing the aggregated Segment Descriptor(s) from the hLS.

Step 6: The VC fetches the aggregated Segment Descriptor(s) from the MA(at) service(s).

Step 7: If the aggregated Segment Descriptor(s) does not have complete topology information – only references to the topology elements, the VC requests the topology information from MA(at) service(s).

Step 8: The hLS informs the VC where the measurement data associated with the topology information are stored (each hLS sends the MA(ad) address).

Step 9: The VC fetches aggregated measurement data from the MA(ad) service.

5.3.3 External consumer - The General Standardised Client Access

Complete sequence of steps defined in the General Standardised Client Access can be performed by the external consumer in order to fetch the measurement data. Go to 9.2.3Appendix B to find details of the procedure.

5.3.4 The use case of the data access compatibility

The following Figure 7 presents an example of the circuit that is created by two different provisioning systems and monitored by two implementations of circuit monitoring solution. This shows that the systems maintained by GN3 and Internet2 can cooperate to set up the connection and the visualisation client may have standardised access to the measurement data for each segment.

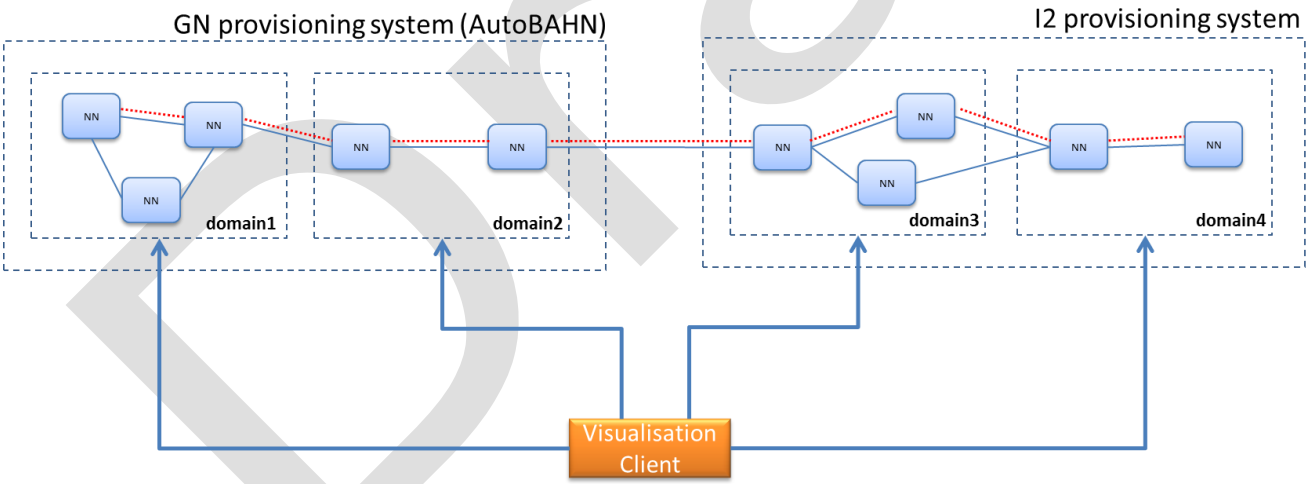

Figure 7: An example of the circuit traversing four domains and created by two provisioning systems.

6 Accessing the Measurement Functionality

The architecture of perfSONAR assumes the existence of different implementations of Measurement Points (MPs). These are software components responsible for retrieving monitoring data from the network. The way MP services obtain monitoring data may vary. At the moment the GN3 MDM release includes the SSH/Telnet MP, BWCTL MP or SNMP MP (still under development).

The SSH/Telnet MP, for example, uses Telnet or SSH protocol to access the command line interface of devices, execute the appropriate command and fetch the result. To perform the measurements using the SSH/Telnet MP for a single network node, the following information can be used:

- Management address (IP)
- Login and password / key
- Equipment model description
- Command (may require additional parameters such as: interface physical address, vlan identifier or logical link identifier). This may be the exact command to be executed in the equipment's management application or a meta command that the SSH/Telnet MP translates into a supported equipment-specific language.

7 Aggregation Functionality

Topology aggregation is performed by AutoBAHN to announce topology data without exposing sensitive domain-internal information[7]. The monitoring data for each circuit must then be aggregated using the same principles so that they can be related/mapped to the aggregated topology.

The following is an example of data aggregation for link status monitoring within the domain.

7.1 Topology

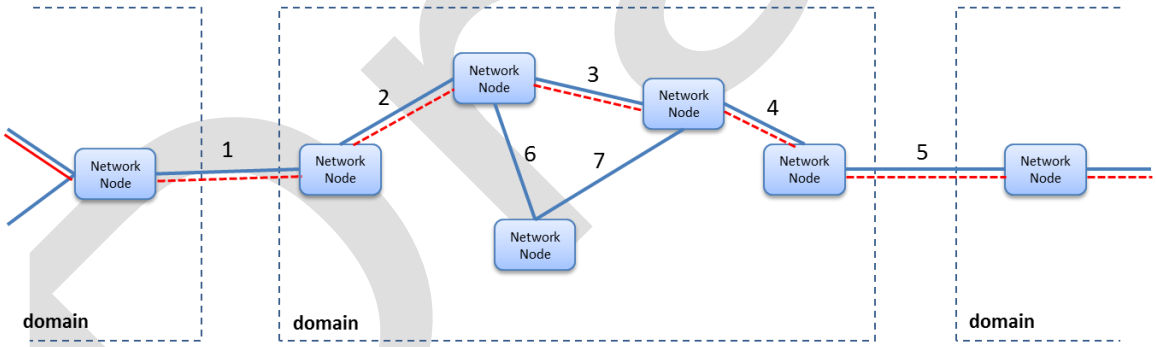

Figure 12 Path established within the domain.

The Domain Manager is requested to establish a path between the domain's edge nodes. The AutoBAHN intra-domain pathfinder informs the DM that there are two possible paths, {2, 3, 4} and {2, 6, 7, 4}. In this case the path {2, 3, 4} is chosen for the reservation. The DM aggregates the path to a single virtual link and keeps the mapping information.

This topology is visible outside the domain as a single abstract link connecting domain ingress and egress points. The resulting aggregated topology and the appropriate mapping are presented below.

[7] In some cases, the AutoBAHN system is interacting with an NMS for circuit provisioning in the underlying domain, therefore some level of aggregation/topology abstraction is already taking place on the NMS-to-AutoBAHN interface.

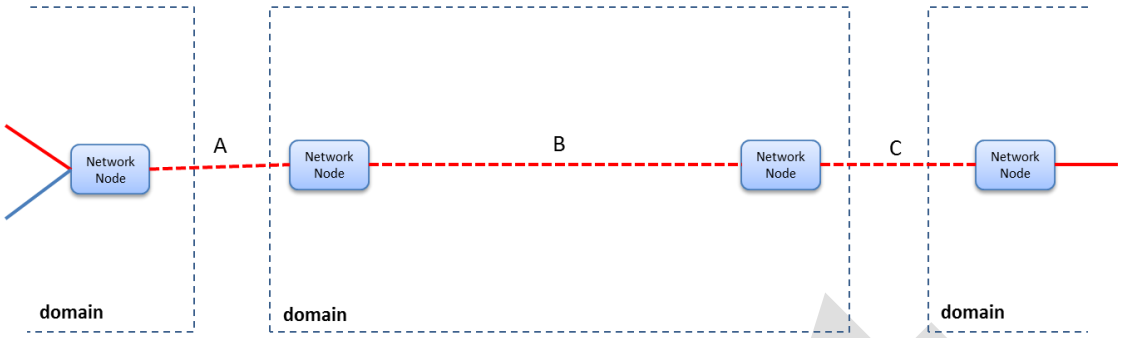

Figure 13 Topology aggregation.

Abstract link	Mapping (for this circuit)
A	{1}
B	{2,3,4}
C	{5}

Table 4 Topology aggregation example

7.2 Measurement Data

The measurement data aggregation must be strictly bound to the topology aggregation. The state of a virtual link is the cumulative state of all the physical links comprising the virtual link within the domain. In the example, the link status is either up (1) or down (0), so the state of a virtual link would be the result of the Boolean AND function on the measured statuses of the physical links comprising the virtual link.

- Link A is always in the same state as link 1.
- Link C is always in the same state as link 5.
- The link B status is the result of AND(2,3,4) and will only be up if all underlying intra-domain links are up.

The MA(ad) stores the aggregated measurement data along with the corresponding abstract links.

Abstract link	Aggregated measurement data
A	{Status: UP; Utilisation: 100000000bps}
B	{Status: UP; Utilisation: 100000000bps}
C	{Status: UP; Utilisation: 100000000bps}

Table 5 Example of aggregated measurement data presenting a single segment of a circuit

7.3 Location

Measurement data aggregation should be performed in the DM, as it already handles the topology aggregation and keeps the mapping information. The DM fetches the measurement data provided for the circuit, performs the aggregation and stores the aggregated data in the MA(ad).

Some modifications need to be introduced to the architecture of the DM. It is proposed to have a flexible system that can perform data aggregation for various complex metrics (for when a simple Boolean function does not suffice). To achieve this goal, a plugin system for handling different types of metrics will be introduced to the Domain Manager. Support for new monitoring metrics will only require registering a new implementation of the metric handling plugin in the DM.

8 Information Model

The information model used for Reservation and Segment Descriptors is based on the OGF NML schema.

8.1 A Reservation Descriptor Example

The following XML snippet includes the Reservation Identifier and three Segment Identifiers referring to tree segments (see an example of a circuit in the section 9.2).

```
<!-- reservation descriptor -->

<nml:topology xmlns:nml="http://schemas.ogf.org/nml/ base/201103/beta/">

  <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1">
    <nml:relation type="serialcompound">

      <nml:link idRef="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER">
        <nml:relation type="next">
          <nml:link idRef="urn:ogf:network:geant.net:PIONIER@1275987204 res 1:GEANT" />
        </nml:relation>
      </nml:link>

      <nml:link idRef="urn:ogf:network:geant.net:PIONIER@1275987204_res_1:GEANT">
        <nml:relation type="next">
          <nml:link idRef="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:HEANET" />
        </nml:relation>
      </nml:link>

      <nml:link idRef="urn:ogf:network:heanet.net:PIONIER@1275987204_res_1:HEANET" />

    </nml:relation>
  </nml:link>
</nml:topology>
```

The Reservation Descriptor represents the Inter-Domain View as it refers to the topology aggregated by the AutoBAHN system.

8.2 A Segment Descriptor Example

The following XML snippet includes network elements comprising the path in the segment (see an example of a circuit in the section 9.2).

```
<!-- segment descriptor -->

<nml:topology xmlns:nml="http://schemas.ogf.org/nml/base/201103/beta/"
              xmlns:nml-eth=" http://schemas.ogf.org/nml/ethernet/201103/beta/">


  <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER">

    <nml:relation type="bidirectional">
      <nml:link idRef="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER AtoB"/>
      <nml:link idRef="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER BtoA"/>
    </nml:relation>

  </nml:link>


  <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER AtoB">

      <nml:capacity>1000000000</nml:capacity>
      <nml:relation type="connects">

        <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/3.100">
          <nml:relation type="next">
            <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/11.100" />
          </nml:relation>
        </nml:port>

        <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/11.100">
          <nml:relation type="next">
            <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/2.100" />
          </nml:relation>
        </nml:port>

        <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/2.100">
          <nml:relation type="next">
            <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/3.100" />
          </nml:relation>
        </nml:port>

        <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/3.100">
          <nml:relation type="next">
            <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/5.100" />
          </nml:relation>
        </nml:port>

        <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/5.100">
          <nml:relation type="next">
            <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/11.100" />
          </nml:relation>
        </nml:port>

        <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/11.100"/>

      </nml:relation>

  </nml:link>


  <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER_BtoA">

      <nml:capacity>1000000000</nml:capacity>
      <nml:relation type="connects">

        <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/11.100">
```

```
          <nml:relation type="next">
            <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/5.100" />
          </nml:relation>
        </nml:port>

        <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/5.100">
          <nml:relation type="next">
            <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/3.100" />
          </nml:relation>
        </nml:port>

        <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/3.100">
          <nml:relation type="next">
            <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/2.100" />
          </nml:relation>
        </nml:port>

        <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/2.100">
          <nml:relation type="next">
            <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/11.100" />
          </nml:relation>
        </nml:port>

        <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/11.100">
          <nml:relation type="next">
            <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/3.100" />
          </nml:relation>
        </nml:port>

        <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/3.100" />

      </nml:relation>

</nml:link>



<!-- Port definitions: -->

<nml:port id="urn:ogf:network:pionier.net:POZ:Gig1/0/3">
  <nml:name>Gig1/0/3</nml:name>
  <nml:relation type="contained-in">
    <nml:node idRef="urn:ogf:network:pionier.net:POZ"/>
  </nml:relation>
  <nml:relation type="contains">
    <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/3.100"/>
  </nml:relation>
</nml:port>

<nml:port id="urn:ogf:network:pionier.net:POZ:Gig1/0/11">
  <nml:name>Gig1/0/11</nml:name>
  <nml:relation type="contained-in">
    <nml:node idRef="urn:ogf:network:pionier.net:POZ"/>
  </nml:relation>
  <nml:relation type="contains">
    <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/11.100"/>
  </nml:relation>
</nml:port>

<nml:port id="urn:ogf:network:pionier.net:WAR:Gig1/0/2">
  <nml:name>Gig1/0/2</nml:name>
  <nml:relation type="contained-in">
    <nml:node idRef="urn:ogf:network:pionier.net:WAR"/>
  </nml:relation>
  <nml:relation type="contains">
    <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/2.100"/>
  </nml:relation>
</nml:port>

<nml:port id="urn:ogf:network:pionier.net:WAR:Gig1/0/3">
  <nml:name>Gig1/0/3</nml:name>
  <nml:relation type="contained-in">
    <nml:node idRef="urn:ogf:network:pionier.net:WAR"/>
  </nml:relation>
```

```
  <nml:relation type="contains">
    <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/3.100"/>
  </nml:relation>
</nml:port>

<nml:port id="urn:ogf:network:pionier.net:CRA:Gig1/0/5">
  <nml:name>Gig1/0/5</nml:name>
  <nml:relation type="contained-in">
    <nml:node idRef="urn:ogf:network:pionier.net:CRA"/>
  </nml:relation>
  <nml:relation type="contains">
    <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/5.100"/>
  </nml:relation>
</nml:port>

<nml:port id="urn:ogf:network:pionier.net:CRA:Gig1/0/11">
  <nml:name>Gig1/0/11</nml:name>
  <nml:relation type="contained-in">
    <nml:node idRef="urn:ogf:network:pionier.net:CRA"/>
  </nml:relation>
  <nml:relation type="contains">
    <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/11.100"/>
  </nml:relation>
</nml:port>



<!-- vlan definitions:-->

<nml:port id="urn:ogf:network:pionier.net:POZ:Gig1/0/3.100">
  <nml:name>Gig1/0/3.100</nml:name>
  <nml-eth:vlan>100</nml-eth:vlan>
  <nml:relation type="contained-in">
    <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/3"/>
  </nml:relation>
</nml:port>

<nml:port id="urn:ogf:network:pionier.net:POZ:Gig1/0/11.100">
  <nml:name>Gig1/0/11.100</nml:name>
  <nml-eth:vlan>100</nml-eth:vlan>
  <nml:relation type="contained-in">
    <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/11"/>
  </nml:relation>
</nml:port>

<nml:port id="urn:ogf:network:pionier.net:WAR:Gig1/0/2.100">
  <nml:name>Gig1/0/2.100</nml:name>
  <nml-eth:vlan>100</nml-eth:vlan>
  <nml:relation type="contained-in">
    <nml:port idRef="urn:ogf:network:pionier.net:WAW:Gig1/0/2"/>
  </nml:relation>
</nml:port>

<nml:port id="urn:ogf:network:pionier.net:WAR:Gig1/0/3.100">
  <nml:name>Gig1/0/3.100</nml:name>
  <nml-eth:vlan>100</nml-eth:vlan>
  <nml:relation type="contained-in">
    <nml:port idRef="urn:ogf:network:pionier.net:WAW:Gig1/0/3"/>
  </nml:relation>
</nml:port>

<nml:port id="urn:ogf:network:pionier.net:CRA:Gig1/0/5.100">
  <nml:name>Gig1/0/5.100</nml:name>
  <nml-eth:vlan>100</nml-eth:vlan>
  <nml:relation type="contained-in">
    <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/5"/>
  </nml:relation>
</nml:port>

<nml:port id="urn:ogf:network:pionier.net:CRA:Gig1/0/11.100">
  <nml:name>Gig1/0/11.100</nml:name>
  <nml:vlan-eth>100</nml:vlan-eth>
  <nml:relation type="contained-in">
    <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/11"/>
```

```
    </nml:relation>
  </nml:port>



  <!-- Node definitions: -->

  <nml:node id="urn:ogf:network:pionier.net:POZ">
    <nml:name>POZ</nml:name>
    <nml:location>
      <nml:country>Poland</nml:country>
      <nml:city>Poznan</nml:city>
    </nml:location>
    <nml:relation type="contains">
      <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/3"/>
      <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/11"/>
    </nml:relation>
  </nml:node>

  <nml:node id="urn:ogf:network:pionier.net:WAR">
    <nml:name>WAR</nml:name>
    <nml:location>
      <nml:country>Poland</nml:country>
      <nml:city>Warsaw</nml:city>
    </nml:location>
    <nml:relation type="contains">
      <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/2"/>
      <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/3"/>
    </nml:relation>
  </nml:node>

  <nml:node id="urn:ogf:network:pionier.net:CRA">
    <nml:name>CRA</nml:name>
    <nml:location>
      <nml:country>Poland</nml:country>
      <nml:city>Cracow</nml:city>
    </nml:location>
    <nml:relation type="contains">
      <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/5"/>
      <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/11"/>
    </nml:relation>
  </nml:node>


</nml:topology>
```

The Segment Descriptor represents the Intra-Domain View as it refers to the real topolology (not aggregated) inside a domain.

8.3 An Aggregated Segment Descriptor Example

```
<!-- aggregated (abstract) segment descriptor -->

<nml:topology xmlns:nml="http://schemas.ogf.org/nml/2013/10/base/">

  <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER">
    <nml:relation type="bidirectional">
      <nml:link idRef="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER_AtoB"/>
      <nml:link idRef="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER_BtoA"/>
    </nml:relation>
  </nml:link>

  <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER_AtoB">
    <nml:relation type="serialcompound">
      <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER:10.10.64.1">
        <nml:name>10.10.64.1</nml:name>
        <nml:relation type="next">
          <nml:link idRef="urn:ogf:network:geant.net:PIONIER@1275987204_res_1:PIONIER:10.10.64.2" />
        </nml:relation>
      </nml:link>
      <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER:10.10.64.2">
        <nml:name>10.10.64.2</nml:name>
        <nml:relation type="next">
          <nml:link idRef="urn:ogf:network:geant.net:PIONIER@1275987204_res_1:PIONIER:10.10.64.3" />
        </nml:relation>
      </nml:link>
      <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER:10.10.64.3">
        <nml:name>10.10.64.3</nml:name>
      </nml:link>
    </nml:relation>
  </nml:link>

  <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER BtoA">
    <nml:relation type="serialcompound">
      <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER:10.10.64.3">
        <nml:name>10.10.64.3</nml:name>
        <nml:relation type="next">
          <nml:link idRef="urn:ogf:network:geant.net:PIONIER@1275987204 res 1:PIONIER:10.10.64.2" />
        </nml:relation>
      </nml:link>
      <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER:10.10.64.2">
        <nml:name>10.10.64.2</nml:name>
        <nml:relation type="next">
          <nml:link idRef="urn:ogf:network:geant.net:PIONIER@1275987204 res 1:PIONIER:10.10.64.1" />
        </nml:relation>
      </nml:link>
      <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER:10.10.64.1">
        <nml:name>10.10.64.1</nml:name>
      </nml:link>
    </nml:relation>
  </nml:link>

</nml:topology>
```

The aggregated Segment Descriptor represents the Inter-Domain View as it refers to the aggregated (abstract) topolology of a domain. The AutoBAHN aggregation transforms sequence of links into only three links.

8.4 Schema

See schema definitions in section 9.1 (the same schema files are used for descriptors and messages).

9 Communication

9.1 Schema

Schema definitions can be found in separate files attached to this specification document (schema definitions that are not extensions for the AutoBAHN and perfSOANR integration are also available on the SVN server – see table below).

9.1.1 perfSONAR Schema

To define the schema for perfSONAR XML elements, the RELAX NG [RelaxNG] standard is used.

Namespace	URI	Location (address)
nmwg	http://ggf.org/ns/nmwg/base/2.0/	https://<svn address>*/nmbase.rnc
perfsonar	http://ggf.org/ns/nmwg/tools/org/perfsonar/1.0/	https://<svn address>*/sonar.rnc
psservice	http://ggf.org/ns/nmwg/tools/org/perfsonar/service/1.0/	https://<svn address>*/service.rnc
xquery	http://ggf.org/ns/nmwg/tools/org/perfsonar/service/lookup/xquery/1.0/	https://<svn address>*/service.rnc
ifevt	http://ggf.org/ns/nmwg/event/status/base/2.0/	https://<svn address>*/event.rnc
select	http://ggf.org/ns/nmwg/ops/select/2.0/	https://<svn address>*/filter.rnc
nmtm	http://ggf.org/ns/nmwg/time/20070914/	https://<svn address>*/nmtime.rnc
nmwgr	http://ggf.org/ns/nmwg/result/2.0/	https://<svn address>*/result.rnc

Table 6 Already existing schemas

* <svn address> is svn.internet2.edu/svn/nmwg/trunk/schema/rnc

Table 7 New perfSONAR schema files supporting integration with AutoBAHN

TODO

9.1.2 AutoBAHN Schema

To define the schema for AutoBAHN XML elements, the WSDL [10] standard is used.

wsdl:definitions name	Location (file name)
Idm2Dm	Idm2dm-ps.wsdl
UserAccessPoint	Useraccesspoint.wsdl

Table 8 AutoBAHN schema files with modifications supporting integration with perfSONAR

9.1.2.1 *Modifications to the AutoBAHN schema to support integration with perfSONAR*

```
<!-- New type - boolean enable + sequence of strings –
     for monitoring  metrics
-->
<xs:complexType name="MonitoringParams">
  <xs:sequence>
    <xs:element name="enable" type="xs:boolean"/>
    <xs:element maxOccurs="unbounded" name="metrics" nillable="true" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

9.2 Communication Example (Messages)

Figure 14 presents an example of a circuit between Poznan and Dublin end points. The path passes three domains: PIONIER, GÉANT and HEANET. The examples of messages presented in this section contain communications that relate mainly to the PIONIER domain and are focused on the **path status metric**. Only PIONIER-internal topology is described.

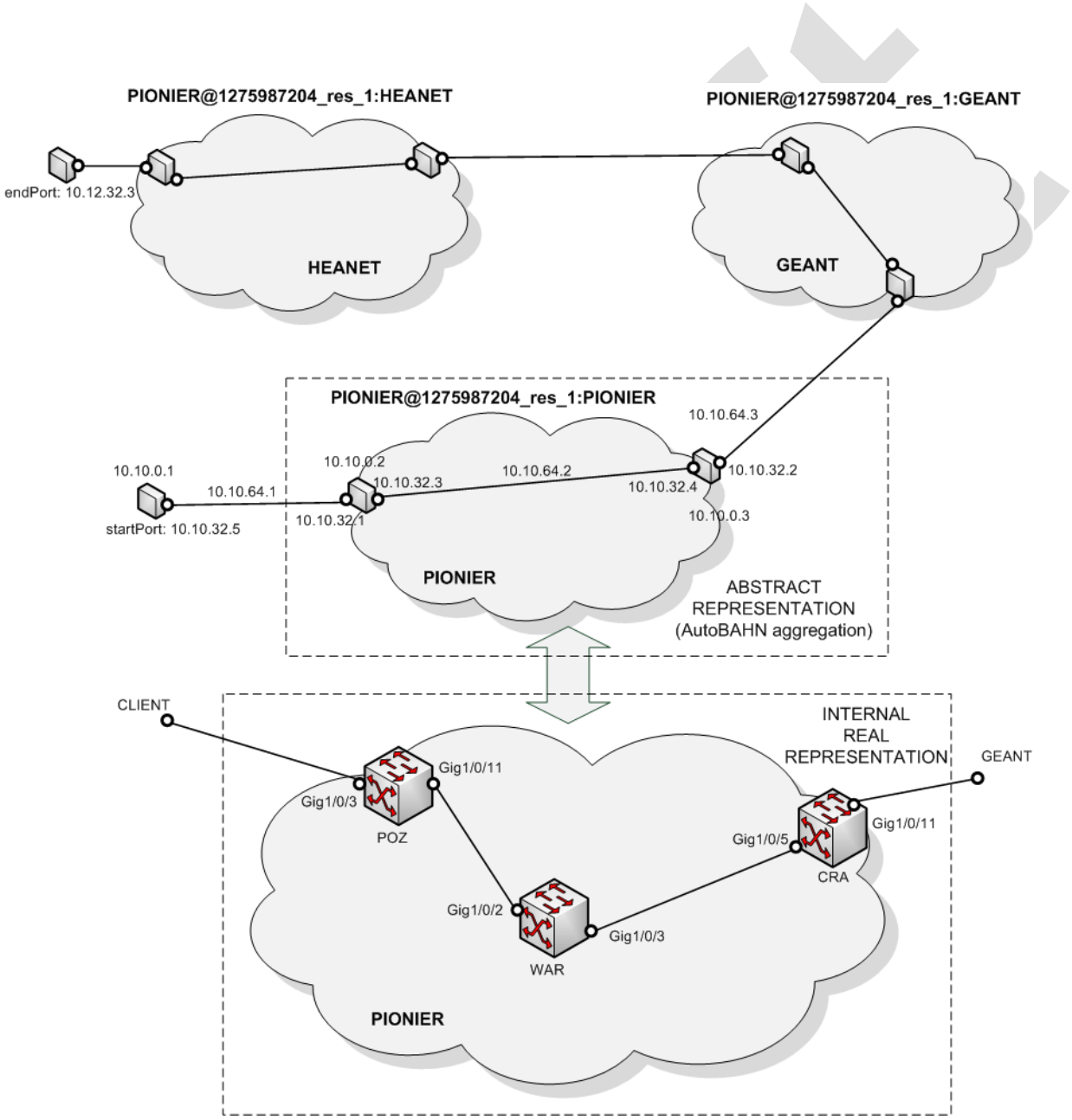

Figure 14 An example of a circuit traversing three domains.

Key:

- Three domains: PIONIER, GÉANT, HEANET (three segments with the unique identifiers: PIONIER@1275987204_res_1:PIONIER, PIONIER@1275987204_res_1:GEANT, PIONIER@1275987204_res_1:HEANET)
- End ports: 10.10.32.5, 10.12.32.3 (AutoBAHN addressing)
- Domain PIONIER – abstract representation (AutoBAHN aggregation)
  - Node 10.10.0.2 (AutoBAHN addressing)
  - Ports: 10.10.32.1, 10.10.32.3 (AutoBAHN addressing)
  - Node 10.10.0.3 (AutoBAHN addressing)
    - Ports: 10.10.32.4, 10.10.32.2 (AutoBAHN addressing)
- Domain PIONIER – internal real topology
  - Node: POZ
    - Ports: Gig1/0/3, Gig1/0/11
  - Node: WAR
    - Ports: Gig1/0/2, Gig1/0/3
  - Node: CRA
    - Ports: Gig1/0/5, Gig1/0/11

9.2.1 Setting up Circuit Monitoring

The message examples in the following sections refer to the workflow steps presented in section 5.2.

9.2.1.1 *Step 0*

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Request message from SQL MA to LS -->
<!-- Step 0 -->

<nmwg:message type="LSRegisterRequest"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:perfsonar="http://ggf.org/ns/nmwg/tools/org/perfsonar/1.0/"
              xmlns:psservice="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/1.0/">

  <nmwg:metadata id="serviceLookupInfo">
    <perfsonar:subject id="commonParameters">
      <psservice:service id="serviceParameters">
        <psservice:serviceName>SQL MA</psservice:serviceName>
        <psservice:accessPoint>
          http://host:8080/services/ma
        </psservice:accessPoint>
        <psservice:serviceType>MA</psservice:serviceType>
      </psservice:service>
    </perfsonar:subject>
    <!-- The following parameter(s) indicates which metric(s) -->
    <!-- can be stored in SQL MA -->
    <psservice:parameters>
      <nmwg:parameter name="eventType">
        http://ggf.org/ns/nmwg/event/status/2.0
      </nmwg:parameter>
      <nmwg:parameter name="eventType">
        http://ggf.org/ns/nmwg/ops/storage/measurement/2.0
      </nmwg:parameter>
    </psservice:parameters>
```

```
  </nmwg:metadata>

  <!-- There is no data element(s) with metadata information because -->
  <!-- at the beginning (when the service starts) MA does not store any data -->
  <nmwg:data id="data1" metadataIdRef="serviceLookupInfo"/>


  <!-- If the MA contains any data it sends their metadata -->
  <!-- elements to the hLS. -->
  <!--
  <nmwg:data id="data1" metadataIdRef="serviceLookupInfo">
    ...
  </nmwg:data>
  -->

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Response message from LS to SQL MA -->
<!-- Step 0 -->

<nmwg:message type="LSRegisterResponse"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/">

  <nmwg:metadata id="resultCodeMetadata">
    <nmwg:eventType>success.ls.register</nmwg:eventType>
    <nmwg:key>
      <nmwg:parameters>
        <nmwg:parameter
          name="lsKey"
          value="http://host:8080/services/ma"/>
      </nmwg:parameters>
    </nmwg:key>
  </nmwg:metadata>

  <nmwg:data
    id="resultCodeData"
    metadataIdRef="resultCodeMetadata">
    <nmwgr:datum xmlns:nmwgr="http://ggf.org/ns/nmwg/result/2.0/">
      Data has been registered with key [http://host:8080/services/ma]
    </nmwgr:datum>
  </nmwg:data>

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Request message from MP to LS -->
<!-- Step 0 -->

<nmwg:message type="LSRegisterRequest"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:perfsonar="http://ggf.org/ns/nmwg/tools/org/perfsonar/1.0/"
              xmlns:psservice="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/1.0/">

  <nmwg:metadata id="meta1">
    <perfsonar:subject id="1">
      <psservice:service id="mp 1">
        <psservice:serviceName>
          Link Status MP
        </psservice:serviceName>
        <psservice:accessPoint>
          http://host:8080/services/mp
        </psservice:accessPoint>
        <psservice:serviceType>MP</psservice:serviceType>
        <psservice:serviceDescription>
          MP contacts devices to find statuses of links
        </psservice:serviceDescription>
        <psservice:organization>PSNC</psservice:organization>
```

```
      </psservice:service>
    </perfsonar:subject>
    <psservice:parameters>
      <!-- MP finds link status information -->
      <nmwg:parameter name="eventType">
        http://ggf.org/ns/nmwg/event/status/2.0
      </nmwg:parameter>
      <nmwg:parameter name="eventType">
        http://ggf.org/ns/nmwg/ops/creation/measurement/2.0
      </nmwg:parameter>
    </psservice:parameters>
  </nmwg:metadata>

  <nmwg:data id="data1" metadataIdRef="meta1"/>

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Response message from LS to MP -->
<!-- Step 0 -->

<nmwg:message type="LSRegisterResponse"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/">

  <nmwg:metadata id="resultCodeMetadata">
    <nmwg:eventType>success.ls.register</nmwg:eventType>
    <nmwg:key>
      <nmwg:parameters>
        <nmwg:parameter
          name="lsKey"
          value="http://host:8080/services/mp"/>
      </nmwg:parameters>
    </nmwg:key>
  </nmwg:metadata>

  <nmwg:data
    id="resultCodeData"
    metadataIdRef="resultCodeMetadata">
    <nmwgr:datum xmlns:nmwgr="http://ggf.org/ns/nmwg/result/2.0/">
      Data has been registered with key [http://host:8080/services/mp]
    </nmwgr:datum>
  </nmwg:data>

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Request message from SIP to LS -->
<!-- Step 0 -->

<nmwg:message type="LSRegisterRequest"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:perfsonar="http://ggf.org/ns/nmwg/tools/org/perfsonar/1.0/"
              xmlns:psservice="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/1.0/">

  <nmwg:metadata id="meta1">
    <perfsonar:subject id="1">
      <psservice:service id="sip 1">
        <psservice:serviceName>
          System Integration Point
        </psservice:serviceName>
        <psservice:accessPoint>
          http://host:8080/services/sip
        </psservice:accessPoint>
        <!-- System Integration Point (SIP) is a new type of perfSONAR service -->
        <psservice:serviceType>SIP</psservice:serviceType>
        <psservice:organization>PIONIER</psservice:organization>
      </psservice:service>
    </perfsonar:subject>
```

```
    <psservice:parameters>
      <nmwg:parameter name="eventType">
        http://ggf.org/ns/nmwg/ops/integration/autobahn/2.0
      </nmwg:parameter>
    </psservice:parameters>
  </nmwg:metadata>

  <!-- There is no data element(s) with metadata information because -->
  <!-- at the beginning (when the service starts) SIP does not store any data -->
  <nmwg:data id="data1" metadataIdRef="meta1"/>

  <!--
  <nmwg:data id="data1" metadataIdRef="meta1">
    -->
    <!-- Segment Identifier-->
    <!--
    <nmwg:metadata id="meta1">
      <nmwg:subject id="sub1">
        <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER"/>
      </nmwg:subject>
      <nmwg:eventType>http://ggf.org/ns/nmwg/event/status/2.0</nmwg:eventType>
      <nmwg:parameters>
        <nmwg:parameter name="xxx">yyy</nmwg:parameter>
      </nmwg:parameters>
    </nmwg:metadata>
  </nmwg:data>
  -->

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Request message from LS to SIP -->
<!-- Step 0 -->

<nmwg:message type="LSRegisterResponse"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/">

  <nmwg:metadata id="resultCodeMetadata">
    <nmwg:eventType>success.ls.register</nmwg:eventType>
    <nmwg:key>
      <nmwg:parameters>
        <nmwg:parameter
          name="lsKey"
          value="http://host:8080/services/sip"/>
      </nmwg:parameters>
    </nmwg:key>
  </nmwg:metadata>

  <nmwg:data
    id="resultCodeData"
    metadataIdRef="resultCodeMetadata">
    <nmwgr:datum xmlns:nmwgr="http://ggf.org/ns/nmwg/result/2.0/">
      Data has been registered with key [http://host:8080/services/sip]
    </nmwgr:datum>
  </nmwg:data>

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Request message from MA(at) to LS -->
<!-- Step 0 -->

<nmwg:message type="LSRegisterRequest"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:perfsonar="http://ggf.org/ns/nmwg/tools/org/perfsonar/1.0/"
              xmlns:psservice="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/1.0/">

  <nmwg:metadata id="meta1">
```

```
    <perfsonar:subject id="1">
      <psservice:service id="maat_1">
        <psservice:serviceName>
          Measurement Archive for inter-domain(abstract) topology
        </psservice:serviceName>
        <psservice:accessPoint>
          http://host:8080/services/ma_at
        </psservice:accessPoint>
        <psservice:serviceType>MA</psservice:serviceType>
        <psservice:organization>PIONIER</psservice:organization>
      </psservice:service>
    </perfsonar:subject>
    <psservice:parameters>
      <nmwg:parameter name="eventType">
        http://ggf.org/ns/nmwg/ops/storage/topology/aggregated/autobahn/2.0
      </nmwg:parameter>
    </psservice:parameters>
  </nmwg:metadata>

  <!-- There is no data element(s) with metadata information because -->
  <!-- at the beginning (when the service starts) MA(at) does not store any data -->
  <nmwg:data id="data1" metadataIdRef="meta1"/>

  <!--
  <nmwg:data id="data1" metadataIdRef="meta1"/>
    <nmwg:metadata id="m1">
      <nmwg:subject id="s1">
        <nml:domain xmlns:nml=http://schemas.ogf.org/nml/base/201103/beta/
                    id="urn:ogf:pionier.net" />
      </nmwg:subject>
      <nmwg:eventType>http://ggf.org/ns/nmwg/topology/20110308</nmwg:eventType>
    </nmwg:metadata>
  </nmwg:data>
  -->

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Request message from LS to MA(at) -->
<!-- Step 0 -->

<nmwg:message type="LSRegisterResponse"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/">

  <nmwg:metadata id="resultCodeMetadata">
    <nmwg:eventType>success.ls.register</nmwg:eventType>
    <nmwg:key>
      <nmwg:parameters>
        <nmwg:parameter
          name="lsKey"
          value="http://host:8080/services/ma at"/>
      </nmwg:parameters>
    </nmwg:key>
  </nmwg:metadata>

  <nmwg:data
    id="resultCodeData"
    metadataIdRef="resultCodeMetadata">
    <nmwgr:datum xmlns:nmwgr="http://ggf.org/ns/nmwg/result/2.0/">
      Data has been registered with key [http://host:8080/services/ma at]
    </nmwgr:datum>
  </nmwg:data>

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Request message from MA(t) to LS -->
<!-- Step 0 -->

<nmwg:message type="LSRegisterRequest"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:perfsonar="http://ggf.org/ns/nmwg/tools/org/perfsonar/1.0/"
              xmlns:psservice="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/1.0/">

  <nmwg:metadata id="meta1">
    <perfsonar:subject id="1">
      <psservice:service id="mat_1">
        <psservice:serviceName>
          Measurement Archive for intra-domain topology
        </psservice:serviceName>
        <psservice:accessPoint>
          http://host:8080/services/ma t
        </psservice:accessPoint>
        <psservice:serviceType>MA</psservice:serviceType>
        <psservice:organization>PIONIER</psservice:organization>
      </psservice:service>
    </perfsonar:subject>
    <psservice:parameters>
      <nmwg:parameter name="eventType">
        http://ggf.org/ns/nmwg/ops/storage/topology/2.0
      </nmwg:parameter>
    </psservice:parameters>
  </nmwg:metadata>

  <!-- There is no data element(s) with metadata information because -->
  <!-- at the beginning (when the service starts) MA(t) does not store any data -->
  <nmwg:data id="data1" metadataIdRef="meta1"/>

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Request message from LS to MA(t) -->
<!-- Step 0 -->

<nmwg:message type="LSRegisterResponse"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/">

  <nmwg:metadata id="resultCodeMetadata">
    <nmwg:eventType>success.ls.register</nmwg:eventType>
    <nmwg:key>
      <nmwg:parameters>
        <nmwg:parameter
          name="lsKey"
          value="http://host:8080/services/ma t"/>
      </nmwg:parameters>
    </nmwg:key>
  </nmwg:metadata>

  <nmwg:data
    id="resultCodeData"
    metadataIdRef="resultCodeMetadata">
    <nmwgr:datum xmlns:nmwgr="http://ggf.org/ns/nmwg/result/2.0/">
      Data has been registered with key [http://host:8080/services/ma t]
    </nmwgr:datum>
  </nmwg:data>

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Request message from MA(ad) to LS -->
<!-- Step 0 -->

<nmwg:message type="LSRegisterRequest"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:perfsonar="http://ggf.org/ns/nmwg/tools/org/perfsonar/1.0/"
              xmlns:psservice="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/1.0/">

  <nmwg:metadata id="meta1">
    <perfsonar:subject id="1">
      <psservice:service id="maad_1">
        <psservice:serviceName>PIONIER Aggregated Data MA</psservice:serviceName>
        <psservice:accessPoint>http://host:8080/services/ma_ad</psservice:accessPoint>
        <psservice:serviceType>MA</psservice:serviceType>
        <psservice:serviceDescription>E2E Circuit Aggregated Data</psservice:serviceDescription>
        <psservice:organization>PIONIER</psservice:organization>
      </psservice:service>
    </perfsonar:subject>
    <!-- The following parameter(s) determine what kind of metric(s) -->
    <!-- can be stored in MA(ad) -->
    <psservice:parameters>
      <nmwg:parameter name="eventType">
        http://ggf.org/ns/nmwg/event/status/2.0
      </nmwg:parameter>
      <!--
      <nmwg:parameter name="eventType">
        http://ggf.org/ns/nmwg/characteristic/utilization/2.0
      </nmwg:parameter>
      -->
      <nmwg:parameter name="eventType">
        http://ggf.org/ns/nmwg/ops/storage/measurement/2.0
     </nmwg:parameter>
      <nmwg:parameter name="eventType">
        http://ggf.org/ns/nmwg/ops/storage/measurement/aggregated/autobahn/2.0
      </nmwg:parameter>
     </psservice:parameters>
  </nmwg:metadata>

  <nmwg:data id="data1" metadataIdRef="meta1">
    <!-- Place for metadata element(s) stored in MA(ad) -->
  </nmwg:data>
</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Request message from LS to MA(ad) -->
<!-- Step 0 -->

<nmwg:message
  id="msg1 resp"
  messageIdRef="msg1"
  type="LSRegisterResponse" xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/">

  <nmwg:metadata id="resultCodeMetadata">
    <nmwg:eventType>success.ls.register</nmwg:eventType>
    <nmwg:key>
      <nmwg:parameters>
        <nmwg:parameter
          name="lsKey"
          value="http://host:8080/services/maad"/>
      </nmwg:parameters>
    </nmwg:key>
  </nmwg:metadata>

  <nmwg:data
    id="resultCodeData"
    metadataIdRef="resultCodeMetadata">
```

```
    <nmwgr:datum xmlns:nmwgr="http://ggf.org/ns/nmwg/result/2.0/">
      Data has been registered with key [http://host:8080/services/maad]
    </nmwgr:datum>
  </nmwg:data>

</nmwg:message>
```

9.2.1.2 *Step 1*

```
<!-- Request message from RC to IDM -->
<!-- Step 1 -->

<ns2:submitService xmlns:ns2="http://useraccesspoint.autobahn.geant.net/">
  <request>
    <userName>user1</userName>
    <userHomeDomain>PIONIER</userHomeDomain>
    <userEmail>sample@test.com</userEmail>
    <justification>Sample service for testing purposes</justification>

    <reservations>
      <!--
        Abstract Autobahn identifier of ingress port. The identifier is not
        the real IP address of the interface, it is an abstract identifier
        which is assigned to each network entity by AutoBAHN system during the
        abstraction process.
      -->
      <startPort>10.10.32.5</startPort>
      <!-- Abstract Autobahn identifier of egress port -->
      <endPort>10.12.32.3</endPort>
      <startTime>2010-07-01T20:58:11.109+02:00</startTime>
      <endTime>2010-07-11T10:58:11.109+02:00</endTime>
      <priority>NORMAL</priority>
      <description>Reservation 1: Poznan - Dublin</description>
      <capacity>1000000000</capacity>
      <userVlanId>0</userVlanId>
      <maxDelay>50</maxDelay>
      <resiliency>NONE</resiliency>
      <bidirectional>false</bidirectional>
      <processNow>false</processNow>

         <!--
           The following new elements enable
           monitoring functionality for the reservation
         -->
      <enable-monitoring>true</enable-monitoring>
      <metrics>status</metrics>
      <!-- Here could come more metrics like utilization -->
      <!-- <metrics>utilization</metrics> -->
    </reservations>
    <!--
      Each reservation request may include more than
      one E2E Circuit:
      <reservations>
      ... First E2E Circuit
      </reservations>
      <reservations>
      ... Second E2E Circuit
      </reservations>
    -->
  </request>
</ns2:submitService>
```

```
<!-- Response message from IDM to RC -->
<!-- Step 1 -->

<ns2:submitServiceResponse xmlns:ns2="http://useraccesspoint.autobahn.geant.net/">
  <!--
    Each serviceID determines a set of E2E Circuits
    which were requested to be reserved
  -->
  <serviceID>PIONIER@1275987204</serviceID>
</ns2:submitServiceResponse>
```

9.2.1.3 *Step 2*

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Request message from IDM to LS -->
<!-- Step 2 -->

<nmwg:message type="LSQueryRequest"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:xquery="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/lookup/xquery/1.0/">

  <nmwg:metadata id="meta1">
    <xquery:subject id="sub1">

      <!-- xquery searching the MA(at) -->

      declare namespace nmwg='http://ggf.org/ns/nmwg/base/2.0/';

      for $m in /nmwg:store[@type='LSStore']/nmwg:metadata
        let $eventType := $m//nmwg:parameter
          where contains($eventType,'http://ggf.org/ns/nmwg/ops/storage/topology/aggregated/2.0')
            return $m

    </xquery:subject>
    <nmwg:eventType>
      http://ogf.org/ns/nmwg/tools/org/perfsonar/service/lookup/query/xquery/2.0
    </nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data metadataIdRef="meta1" id="d1"/>

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Response message from LS to IDM -->
<!-- Step 2 -->

<nmwg:message type="LSQueryResponse"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:psservice="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/1.0/"
              xmlns:perfsonar="http://ggf.org/ns/nmwg/tools/org/perfsonar/1.0/">

  <nmwg:metadata id="LSQueryResponseMetadata">
    <nmwg:eventType>success.ls.query</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="LSQueryResponseData"
             metadataIdRef="LSQueryResponseMetadata">
    <psservice:datum>
      <nmwg:metadata id="meta1">
        <perfsonar:subject id="1">
          <psservice:service id="maat 1">
            <psservice:serviceName>
              Measurement Archive for inter-domain(abstract) topology
```

```
            </psservice:serviceName>
            <psservice:accessPoint>
              http://host:8080/services/ma at
            </psservice:accessPoint>
            <psservice:serviceType>MA</psservice:serviceType>
            <psservice:organization>PIONIER</psservice:organization>
          </psservice:service>
        </perfsonar:subject>
        <psservice:parameters>
          <nmwg:parameter name="eventType">
            http://ggf.org/ns/nmwg/ops/storage/topology/aggregated/2.0
          </nmwg:parameter>
        </psservice:parameters>
      </nmwg:metadata>
    </psservice:datum>
  </nmwg:data>

</nmwg:message>
```

9.2.1.4 *Step 3*

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Request message from IDM to MA(at) -->
<!-- Step 3 -->

<nmwg:message type="TSAddRequest"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:nml="http://schemas.ogf.org/nml/base/201103/beta/">

  <nmwg:metadata id="meta0">
    <nmwg:eventType>http://ggf.org/ns/nmwg/topology/201103</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="data0" metadataIdRef="meta0">

    <!-- reservation/circuit descriptor -->
    <nml:topology>

      <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1">
        <nml:relation type="serialcompound">
          <nml:link idRef="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER">
            <nml:relation type="next">
              <nml:link idRef="urn:ogf:network:geant.net:PIONIER@1275987204_res_1:GEANT" />
            </nml:relation>
          </nml:link>
          <nml:link idRef="urn:ogf:network:geant.net:PIONIER@1275987204 res 1:GEANT">
            <nml:relation type="next">
              <nml:link idRef="urn:ogf:network:heanet.net:PIONIER@1275987204_res_1:HEANET" />
            </nml:relation>
          </nml:link>
          <nml:link idRef="urn:ogf:network:heanet.net:PIONIER@1275987204 res 1:HEANET" />
        </nml:relation>
      </nml:link>

    </nml:topology>
  </nmwg:data>

 </nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Response message from MA(at) to IDM -->
<!-- Step 3 -->

<nmwg:message type="TSAddResponse"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/">

  <nmwg:metadata id="meta0">
    <nmwg:eventType>success.ma.added</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="data0" metadataIdRef="meta0">
    <nmwgr:datum xmlns:nmwgr="http://ggf.org/ns/nmwg/result/2.0/">
      1 element(s) successfully added
    </nmwgr:datum>
  </nmwg:data>

</nmwg:message>
```

9.2.1.5 *Step 4*

```
<!-- Request message from IDM to DM -->
<!-- Step 4 -->

<ns2:addReservation xmlns:ns2="http://idm2dm.autobahn.geant.net/">
  <resID>PIONIER@1275987204 res 1</resID>
  <links>
    <bodID>10.10.64.1</bodID>
    <kind>3</kind>
    <startPort>
      <bodID>10.10.32.1</bodID>
      <address>10.10.32.1</address>
      <technology>Ethernet</technology>
      <bundled>false</bundled>
      <node>
        <bodID>10.10.0.1</bodID>
        <provisioningDomain>
          <bodID>PIONIER</bodID>
          <adminDomain>
            <bodID>PIONIER</bodID>
            <name>PIONIER</name>
            <clientDomain>false</clientDomain>
          </adminDomain>
        </provisioningDomain>
      </node>
    </startPort>
    <endPort>
      <bodID>10.10.32.5</bodID>
      <address>10.10.32.5</address>
      <technology>Ethernet</technology>
      <bundled>false</bundled>
      <node>
        <bodID>10.10.0.2</bodID>
        <provisioningDomain>
          <bodID>client-domain1.PIONIER</bodID>
          <adminDomain>
            <bodID>client-domain1.PIONIER</bodID>
            <name>client-domain1.PIONIER</name>
            <clientDomain>true</clientDomain>
          </adminDomain>
        </provisioningDomain>
      </node>
    </endPort>
    <bidirectional>false</bidirectional>
    <delay>0</delay>
    <manualCost>0.0</manualCost>
    <monetaryCost>0.0</monetaryCost>
```

```
    <granularity>0</granularity>
    <minResCapacity>0</minResCapacity>
    <maxResCapacity>0</maxResCapacity>
    <capacity>1000000000</capacity>
    <type>
      <type>2</type>
    </type>
    <operationalState>
      <state>1</state>
    </operationalState>
    <administrativeState>
      <state>1</state>
    </administrativeState>
  </links>
  <!--
    More links come here:
    <links>
    ...
    </links>
  -->
  <params>
    <capacity>1000000000</capacity>
    <maxDelay>50</maxDelay>
    <resiliency>NONE</resiliency>
    <bidirectional>false</bidirectional>
    <startTime>2010-07-01T21:10:05.484+02:00</startTime>
    <endTime>2010-07-11T11:10:05.484+02:00</endTime>
    <pathConstraints>
      <rangeNames>VLANS</rangeNames>
      <rangeConstraints>
        <ranges>
          <min>100</min>
          <max>100</max>
        </ranges>
      </rangeConstraints>
    </pathConstraints>
    <userVlanId>0</userVlanId>

    <!--
      The following new elements enable
      monitoring functionality for the reservation
    -->
    <enable-monitoring>true</enable-monitoring>
    <metrics>status</metrics>
    <!-- Here could come more metrics like utilization -->
    <!-- <metrics>utilization</metrics> -->
  </params>
</ns2:addReservation>
```

```
<!-- Response message from DM to IDM -->
<!-- Step 4 -->

<ns2:addReservationResponse xmlns:ns2="http://idm2dm.autobahn.geant.net/" />
```

9.2.1.6 *Step 5*

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Request message from DM to LS -->
<!-- Step 5 -->

<nmwg:message type="LSQueryRequest"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:xquery="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/lookup/xquery/1.0/">

  <nmwg:metadata id="meta1">
    <xquery:subject id="sub1">

      <!-- xquery searching the MA(at) or MA(t) or the SIP -->

      <!-- xquery searching the MA(at) -->
      declare namespace nmwg='http://ggf.org/ns/nmwg/base/2.0/';
      for $m in /nmwg:store[@type='LSStore']/nmwg:metadata
        let $eventType := $m//nmwg:parameter
          where contains($eventType,'http://ggf.org/ns/nmwg/ops/storage/topology/aggregated/2.0')
            return $m

      <!-- xquery searching the MA(t) -->
      <!--
      declare namespace nmwg='http://ggf.org/ns/nmwg/base/2.0/';
      for $m in /nmwg:store[@type='LSStore']/nmwg:metadata
        let $eventType := $m//nmwg:parameter
          where contains($eventType,'http://ggf.org/ns/nmwg/ops/storage/topology/2.0')
            return $m
      -->

      <!-- xquery searching the SIP -->
      <!--
      declare namespace nmwg='http://ggf.org/ns/nmwg/base/2.0/';
      for $m in /nmwg:store[@type='LSStore']/nmwg:metadata
        let $eventType := $m//nmwg:parameter
          where contains($eventType,'http://ggf.org/ns/nmwg/ops/integration/autobahn/2.0')
            return $m
      -->

    </xquery:subject>
    <nmwg:eventType>
      http://ogf.org/ns/nmwg/tools/org/perfsonar/service/lookup/query/xquery/2.0
    </nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="data1" metadataIdRef="meta1"/>

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Response message from LS to DM -->
<!-- Step 5 -->

<nmwg:message type="LSQueryResponse"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:psservice="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/1.0/"
              xmlns:perfsonar="http://ggf.org/ns/nmwg/tools/org/perfsonar/1.0/">

  <nmwg:metadata id="LSQueryResponseMetadata">
    <nmwg:eventType>success.ls.query</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="LSQueryResponseData"
             metadataIdRef="LSQueryResponseMetadata">
    <psservice:datum>
```

```
      <nmwg:metadata id="meta1">
        <perfsonar:subject id="1">
          <psservice:service id="maat 1">
            <psservice:serviceName>
              Measurement Archive for inter-domain(abstract) topology
            </psservice:serviceName>
            <psservice:accessPoint>
              http://host:8080/services/ma_at
            </psservice:accessPoint>
            <psservice:serviceType>MA</psservice:serviceType>
            <psservice:organization>PIONIER</psservice:organization>
          </psservice:service>
        </perfsonar:subject>
        <psservice:parameters>
          <nmwg:parameter name="eventType">
            http://ggf.org/ns/nmwg/ops/storage/topology/aggregated/2.0
          </nmwg:parameter>
        </psservice:parameters>
      </nmwg:metadata>
    </psservice:datum>
    <!--
    <psservice:datum>
      <nmwg:metadataid="meta1">
        <perfsonar:subject id="1">
          <psservice:service id="mat 1">
            <psservice:serviceName>
              Measurement Archive for intra-domain topology
            </psservice:serviceName>
            <psservice:accessPoint>
              http://host:8080/services/ma t
            </psservice:accessPoint>
            <psservice:serviceType>MA</psservice:serviceType>
            <psservice:organization>PIONIER</psservice:organization>
          </psservice:service>
        </perfsonar:subject>
        <psservice:parameters>
          <nmwg:parameter name="eventType">
            http://ggf.org/ns/nmwg/ops/storage/topology/2.0
          </nmwg:parameter>
        </psservice:parameters>
      </nmwg:metadata>
    </psservice:datum>
    or
    <psservice:datum>
      <nmwg:metadata id="meta1">
        <perfsonar:subject id="1">
          <psservice:service id="sip_1">
            <psservice:serviceName>
              System Integration Point
            </psservice:serviceName>
            <psservice:accessPoint>
              http://host:8080/services/sip
            </psservice:accessPoint>
            <psservice:serviceType>SIP</psservice:serviceType>
            <psservice:organization>PIONIER</psservice:organization>
          </psservice:service>
        </perfsonar:subject>
        <psservice:parameters>
          <nmwg:parameter name="eventType">
            http://ggf.org/ns/nmwg/ops/integration/autobahn/2.0
          </nmwg:parameter>
        </psservice:parameters>
      </nmwg:metadata>
    </psservice:datum>
  -->
  </nmwg:data>

</nmwg:message>
```

9.2.1.7 *Step 6*

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Request message from DM to MA(t) -->
<!-- Step 6 -->

<nmwg:message type="TSAddRequest"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/">

  <nmwg:metadata id="meta1">
    <nmwg:eventType>http://ggf.org/ns/nmwg/topology/201103</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="data1" metadataIdRef="meta1">

    <!-- Segment Descriptor -->
    <nml:topology xmlns:nml="http://schemas.ogf.org/nml/base/201103/beta/"
                  xmlns:nml-eth="http://schemas.ogf.org/nml/ethernet/201103/beta/">


      <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER">
        <nml:relation type="bidirectional">
          <nml:link idRef="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER AtoB"/>
          <nml:link idRef="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER BtoA"/>
        </nml:relation>
      </nml:link>

      <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER_AtoB">

        <nml:capacity>1000000000</nml:capacity>

        <nml:relation type="connects">

          <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/3.100">
            <nml:relation type="next">
              <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/11.100" />
            </nml:relation>
          </nml:port>

          <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/11.100">
            <nml:relation type="next">
              <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/2.100" />
            </nml:relation>
          </nml:port>

          <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/2.100">
            <nml:relation type="next">
              <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/3.100" />
            </nml:relation>
          </nml:port>

          <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/3.100">
            <nml:relation type="next">
              <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/5.100" />
            </nml:relation>
          </nml:port>

          <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/5.100">
            <nml:relation type="next">
              <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/11.100" />
            </nml:relation>
          </nml:port>

          <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/11.100"/>

        </nml:relation>

      </nml:link>

      <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER_BtoA">
```

```
      <nml:capacity>1000000000</nml:capacity>
      <nml:relation type="connects">

        <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/11.100">
          <nml:relation type="next">
            <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/5.100" />
          </nml:relation>
        </nml:port>

        <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/5.100">
          <nml:relation type="next">
            <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/3.100" />
          </nml:relation>
        </nml:port>

        <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/3.100">
          <nml:relation type="next">
            <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/2.100" />
          </nml:relation>
        </nml:port>

        <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/2.100">
          <nml:relation type="next">
            <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/11.100" />
          </nml:relation>
        </nml:port>

        <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/11.100">
          <nml:relation type="next">
            <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/3.100" />
          </nml:relation>
        </nml:port>

        <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/3.100" />

      </nml:relation>

    </nml:link>

    <!-- Port definitions: -->

    <nml:port id="urn:ogf:network:pionier.net:POZ:Gig1/0/3">
      <nml:name>Gig1/0/3</nml:name>
      <nml:relation type="contained-in">
        <nml:node idRef="urn:ogf:network:pionier.net:POZ"/>
      </nml:relation>
      <nml:relation type="contains">
        <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/3.100"/>
      </nml:relation>
    </nml:port>

    <nml:port id="urn:ogf:network:pionier.net:POZ:Gig1/0/11">
      <nml:name>Gig1/0/11</nml:name>
      <nml:relation type="contained-in">
        <nml:node idRef="urn:ogf:network:pionier.net:POZ"/>
      </nml:relation>
      <nml:relation type="contains">
        <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/11.100"/>
      </nml:relation>
    </nml:port>

    <nml:port id="urn:ogf:network:pionier.net:WAR:Gig1/0/2">
      <nml:name>Gig1/0/2</nml:name>
      <nml:relation type="contained-in">
        <nml:node idRef="urn:ogf:network:pionier.net:WAR"/>
      </nml:relation>
      <nml:relation type="contains">
        <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/2.100"/>
      </nml:relation>
    </nml:port>

    <nml:port id="urn:ogf:network:pionier.net:WAR:Gig1/0/3">
      <nml:name>Gig1/0/3</nml:name>
      <nml:relation type="contained-in">
```

```
        <nml:node idRef="urn:ogf:network:pionier.net:WAR"/>
      </nml:relation>
      <nml:relation type="contains">
        <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/3.100"/>
      </nml:relation>
    </nml:port>

    <nml:port id="urn:ogf:network:pionier.net:CRA:Gig1/0/5">
      <nml:name>Gig1/0/5</nml:name>
      <nml:relation type="contained-in">
        <nml:node idRef="urn:ogf:network:pionier.net:CRA"/>
      </nml:relation>
      <nml:relation type="contains">
        <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/5.100"/>
      </nml:relation>
    </nml:port>

    <nml:port id="urn:ogf:network:pionier.net:CRA:Gig1/0/11">
      <nml:name>Gig1/0/11</nml:name>
      <nml:relation type="contained-in">
        <nml:node idRef="urn:ogf:network:pionier.net:CRA"/>
      </nml:relation>
      <nml:relation type="contains">
        <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/11.100"/>
      </nml:relation>
    </nml:port>

    <!-- vlan definitions:-->

    <nml:port id="urn:ogf:network:pionier.net:POZ:Gig1/0/3.100">
      <nml:name>Gig1/0/3.100</nml:name>
      <nml-eth:vlan>100</nml-eth:vlan>
      <nml:relation type="contained-in">
        <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/3"/>
      </nml:relation>
    </nml:port>

    <nml:port id="urn:ogf:network:pionier.net:POZ:Gig1/0/11.100">
      <nml:name>Gig1/0/11.100</nml:name>
      <nml-eth:vlan>100</nml-eth:vlan>
      <nml:relation type="contained-in">
        <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/11"/>
      </nml:relation>
    </nml:port>

    <nml:port id="urn:ogf:network:pionier.net:WAR:Gig1/0/2.100">
      <nml:name>Gig1/0/2.100</nml:name>
      <nml-eth:vlan>100</nml-eth:vlan>
      <nml:relation type="contained-in">
        <nml:port idRef="urn:ogf:network:pionier.net:WAW:Gig1/0/2"/>
      </nml:relation>
    </nml:port>

    <nml:port id="urn:ogf:network:pionier.net:WAR:Gig1/0/3.100">
      <nml:name>Gig1/0/3.100</nml:name>
      <nml-eth:vlan>100</nml-eth:vlan>
      <nml:relation type="contained-in">
        <nml:port idRef="urn:ogf:network:pionier.net:WAW:Gig1/0/3"/>
      </nml:relation>
    </nml:port>

    <nml:port id="urn:ogf:network:pionier.net:CRA:Gig1/0/5.100">
      <nml:name>Gig1/0/5.100</nml:name>
      <nml-eth:vlan>100</nml-eth:vlan>
      <nml:relation type="contained-in">
        <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/5"/>
      </nml:relation>
    </nml:port>

    <nml:port id="urn:ogf:network:pionier.net:CRA:Gig1/0/11.100">
      <nml:name>Gig1/0/11.100</nml:name>
      <nml:vlan-eth>100</nml:vlan-eth>
      <nml:relation type="contained-in">
        <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/11"/>
```

```
        </nml:relation>
      </nml:port>

      <!-- Node definitions: -->

      <nml:node id="urn:ogf:network:pionier.net:POZ">
        <nml:name>POZ</nml:name>
        <nml:location>
          <nml:country>Poland</nml:country>
          <nml:city>Poznan</nml:city>
        </nml:location>
        <nml:relation type="contains">
          <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/3"/>
          <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/11"/>
        </nml:relation>
      </nml:node>

      <nml:node id="urn:ogf:network:pionier.net:WAR">
        <nml:name>WAR</nml:name>
        <nml:location>
          <nml:country>Poland</nml:country>
          <nml:city>Warsaw</nml:city>
        </nml:location>
        <nml:relation type="contains">
          <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/2"/>
          <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/3"/>
        </nml:relation>
      </nml:node>

      <nml:node id="urn:ogf:network:pionier.net:CRA">
        <nml:name>CRA</nml:name>
        <nml:location>
          <nml:country>Poland</nml:country>
          <nml:city>Cracow</nml:city>
        </nml:location>
        <nml:relation type="contains">
          <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/5"/>
          <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/11"/>
      </nml:relation>
      </nml:node>

    </nml:topology>

  </nmwg:data>

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Response message from MA(t) to DM -->
<!-- Step 6 -->

<nmwg:message type="TSAddResponse"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/">

  <nmwg:metadata id="meta1">
    <nmwg:eventType>success.ma.added</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="data1" metadataIdRef="meta1">
    <nmwgr:datum xmlns:nmwgr="http://ggf.org/ns/nmwg/result/2.0/">
      N element(s) successfully added
    </nmwgr:datum>
  </nmwg:data>

</nmwg:message>
```

9.2.1.8 *Step 7*

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Request message from DM to MA(at) -->
<!-- Step 7 -->

<nmwg:message type="TSAddRequest"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/">

  <nmwg:metadata id="meta1">
    <nmwg:eventType>http://ggf.org/ns/nmwg/topology/201103</nmwg:eventType>
    <nmwg:eventType>http://ggf.org/ns/nmwg/topology/aggregated/autobahn/2.0</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="data1" metadataIdRef="meta1">

    <!-- aggregated segment descriptor -->
    <nml:topology xmlns:nml="http://schemas.ogf.org/nml/base/201103/beta/">

      <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER">
        <nml:relation type="bidirectional">
          <nml:link idRef="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER_AtoB"/>
          <nml:link idRef="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER BtoA"/>
        </nml:relation>
      </nml:link>

      <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER_AtoB">
        <nml:relation type="serialcompound">
          <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER:10.10.64.1">
            <nml:name>10.10.64.1</nml:name>
            <nml:relation type="next">
             <nml:link idRef="urn:ogf:network:geant.net:PIONIER@1275987204_res_1:PIONIER:10.10.64.2"/>
            </nml:relation>
          </nml:link>
          <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER:10.10.64.2">
            <nml:name>10.10.64.2</nml:name>
            <nml:relation type="next">
             <nml:link idRef="urn:ogf:network:geant.net:PIONIER@1275987204 res 1:PIONIER:10.10.64.3"/>
            </nml:relation>
          </nml:link>
          <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER:10.10.64.3">
            <nml:name>10.10.64.3</nml:name>
          </nml:link>
        </nml:relation>
      </nml:link>

      <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER BtoA">
        <nml:relation type="serialcompound">
          <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER:10.10.64.3">
            <nml:name>10.10.64.3</nml:name>
            <nml:relation type="next">
             <nml:link idRef="urn:ogf:network:geant.net:PIONIER@1275987204 res 1:PIONIER:10.10.64.2"/>
            </nml:relation>
          </nml:link>
          <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER:10.10.64.2">
            <nml:name>10.10.64.2</nml:name>
            <nml:relation type="next">
             <nml:link idRef="urn:ogf:network:geant.net:PIONIER@1275987204_res_1:PIONIER:10.10.64.1"/>
            </nml:relation>
          </nml:link>
          <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER:10.10.64.1">
            <nml:name>10.10.64.1</nml:name>
          </nml:link>
        </nml:relation>
      </nml:link>

    </nml:topology>

  </nmwg:data>

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Response message from MA(at) to DM -->
<!-- Step 7 -->

<nmwg:message type="TSAddResponse"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/">

  <nmwg:metadata id="meta1">
    <nmwg:eventType>success.ma.added</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="data1" metadataIdRef="meta1">
    <nmwgr:datum xmlns:nmwgr="http://ggf.org/ns/nmwg/result/2.0/">
      N element(s) successfully added
    </nmwgr:datum>
  </nmwg:data>

</nmwg:message>
```

9.2.1.9 *Step 8*

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Request message from DM to SIP -->
<!-- Step 8 -->

<nmwg:message type="MeasurementArchiveStoreRequest"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:nml="http://schemas.ogf.org/nml/base/201103/beta/">

  <nmwg:metadata id="meta1">
    <nmwg:subject>
      <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER"/>
    </nmwg:subject>
    <!-- parameters element helps to contain additional info about the segment and its monitoring -->
    <!--
    <nmwg:parameters id="par1">
      <nmwg:parameter name="xxx">yyy</nmwg:parameter>
    </nmwg:parameters>
    -->
    <nmwg:eventType>http://ggf.org/ns/nmwg/event/status/2.0</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="data1" metadataIdRef="meta1"/>

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Response message from SIP to DM -->
<!-- Step 8 -->

<nmwg:message type="MeasurementArchiveStoreResponse"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/">

  <nmwg:metadata id="resultCodeMetadata">
    <nmwg:eventType>success.ma.write</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="resultDescriptionData_for_resultCodeMetadata"
             metadataIdRef="resultCodeMetadata">
    <nmwgr:datum xmlns:nmwgr="http://ggf.org/ns/nmwg/result/2.0/">
      Data have been stored
    </nmwgr:datum>
  </nmwg:data>
```

```
</nmwg:message>
```

9.2.1.10 *Step 9*

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Request message from SIP to LS -->
<!-- Step 9 -->

<nmwg:message type="LSQueryRequest"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:xquery="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/lookup/xquery/1.0/">

  <nmwg:metadata id="meta1">
    <xquery:subject id="sub1">

      <!-- xquery searching the MA(t) -->
      declare namespace nmwg='http://ggf.org/ns/nmwg/base/2.0/';
      for $m in /nmwg:store[@type='LSStore']/nmwg:metadata
        let $eventType := $m//nmwg:parameter
          where contains($eventType,'http://ggf.org/ns/nmwg/ops/storage/topology/2.0')
            return $m

    </xquery:subject>
    <nmwg:eventType>
      http://ogf.org/ns/nmwg/tools/org/perfsonar/service/lookup/query/xquery/2.0
    </nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data metadataIdRef="meta1" id="d1"/>

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Response message from LS to SIP -->
<!-- Step 9 -->

<nmwg:message type="LSQueryResponse"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:psservice="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/1.0/"
              xmlns:perfsonar="http://ggf.org/ns/nmwg/tools/org/perfsonar/1.0/">

  <nmwg:metadata id="LSQueryResponseMetadata">
    <nmwg:eventType>success.ls.query</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="LSQueryResponseData"
             metadataIdRef="LSQueryResponseMetadata">
    <psservice:datum>
      <nmwg:metadata id="meta1">
        <perfsonar:subject id="1">
          <psservice:service id="mat 1">
            <psservice:serviceName>
              Measurement Archive for intra-domain topology
            </psservice:serviceName>
            <psservice:accessPoint>
              http://host:8080/services/ma_t
            </psservice:accessPoint>
            <psservice:serviceType>MA</psservice:serviceType>
            <psservice:organization>PIONIER</psservice:organization>
          </psservice:service>
        </perfsonar:subject>
        <psservice:parameters>
          <nmwg:parameter name="eventType">
            http://ggf.org/ns/nmwg/ops/storage/topology/2.0
          </nmwg:parameter>
        </psservice:parameters>
```

```
      </nmwg:metadata>
    </psservice:datum>
  </nmwg:data>

</nmwg:message>
```

9.2.1.11 *Step 10*

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Request message from SIP to MA(t) -->
<!-- Step 10 -->

<nmwg:message type="TSQueryRequest"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:xquery="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/lookup/xquery/1.0/">

  <nmwg:metadata id="meta1">
    <xquery:subject id="sub0">
      <!-- xquery statement that fetches the segment descriptor with a given segment identifier -->

      declare namespace nmwg='http://ggf.org/ns/nmwg/base/2.0/';
      declare namespace nml='http://schemas.ogf.org/nml/base/201103/beta/';

      for $lk in /nml:topology/nml:link
        where contains($lk/@id, "urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER")
          or contains($lk/@idRef, "urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER")
        return $lk

    <!-- This query does not return port and node elements pointed by idRef's !!!                  -->
    <!-- SIP may send additional simple request(s) to the MA(t) with the xqery statement(s)        -->
    <!-- including port and node identifiers taken from the response received earlier.             -->
    <!-- Other solutions:                                                                          -->
    <!-- 1) do not use id references in the segment descriptor and have nested element definitions -->
    <!-- 2) create more advanced xquery statement that can understand and follow id references     -->

    </xquery:subject>
    <!-- or we can use xpath -->
    <!--
    <xpath:subject xmlns:xpath="http://ggf.org/ns/nmwg/tools/org/perfsonar/xpath/1.0/">
      /*:topology/*:link[contains(@id,"urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER")]
    </xpath:subject>
    -->
    <nmwg:eventType>http://ggf.org/ns/nmwg/topology/201103</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data metadataIdRef="meta1" id="data1" />

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Response message from MA(t) to SIP -->
<!-- Step 10 -->

<nmwg:message type="TSQueryResponse"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:nml="http://schemas.ogf.org/nml/base/201103/beta/"
              xmlns:nml-eth="http://schemas.ogf.org/nml/ethernet/201103/beta/"
              xmlns:xquery="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/lookup/xquery/1.0/">

  <nmwg:metadata id="meta1">
    <xquery:subject id="sub0">
      declare namespace nmwg='http://ggf.org/ns/nmwg/base/2.0/';
      declare namespace nml='http://schemas.ogf.org/nml/base/201103/beta/';
      for $lk in /nml:topology/nml:link
        where contains($lk/@id, "urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER")
          or contains($lk/@idRef, "urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER")
        return $lk
```

```
  </xquery:subject>
  <nmwg:eventType>http://ggf.org/ns/nmwg/topology/201103</nmwg:eventType>
</nmwg:metadata>

<nmwg:data metadataIdRef="meta1" id="data1">

 <!-- The message contains all elements of segment descriptor-->
 <!-- organised using id references. -->

  <nmwg:datum>
    <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER">
      <nml:relation type="bidirectional">
        <nml:link idRef="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER_AtoB"/>
        <nml:link idRef="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER_BtoA"/>
      </nml:relation>
    </nml:link>
  </nmwg:datum>

  <nmwg:datum>
    <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER_AtoB">
      <nml:capacity>1000000000</nml:capacity>
      <nml:relation type="connects">
        <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/3.100">
          <nml:relation type="next">
            <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/11.100" />
          </nml:relation>
        </nml:port>
        <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/11.100">
          <nml:relation type="next">
            <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/2.100" />
          </nml:relation>
        </nml:port>
        <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/2.100">
          <nml:relation type="next">
            <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/3.100" />
          </nml:relation>
        </nml:port>
        <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/3.100">
          <nml:relation type="next">
            <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/5.100" />
          </nml:relation>
        </nml:port>
        <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/5.100">
          <nml:relation type="next">
            <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/11.100" />
          </nml:relation>
        </nml:port>
        <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/11.100"/>
      </nml:relation>
    </nml:link>
  </nmwg:datum>

  <nmwg:datum>
    <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER BtoA">
      <nml:capacity>1000000000</nml:capacity>
      <nml:relation type="connects">
        <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/11.100">
          <nml:relation type="next">
            <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/5.100" />
          </nml:relation>
        </nml:port>
        <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/5.100">
          <nml:relation type="next">
            <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/3.100" />
          </nml:relation>
        </nml:port>
        <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/3.100">
          <nml:relation type="next">
            <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/2.100" />
          </nml:relation>
        </nml:port>
        <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/2.100">
          <nml:relation type="next">
            <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/11.100" />
```

```
          </nml:relation>
        </nml:port>
        <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/11.100">
          <nml:relation type="next">
            <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/3.100" />
          </nml:relation>
        </nml:port>
        <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/3.100" />
      </nml:relation>
    </nml:link>
  </nmwg:datum>

  <!-- Port definitions: -->

  <nmwg:datum>
    <nml:port id="urn:ogf:network:pionier.net:POZ:Gig1/0/3">
      <nml:name>Gig1/0/3</nml:name>
      <nml:relation type="contained-in">
        <nml:node idRef="urn:ogf:network:pionier.net:POZ"/>
      </nml:relation>
      <nml:relation type="contains">
        <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/3.100"/>
      </nml:relation>
    </nml:port>
  </nmwg:datum>
  <nmwg:datum>
    <nml:port id="urn:ogf:network:pionier.net:POZ:Gig1/0/11">
      <nml:name>Gig1/0/11</nml:name>
      <nml:relation type="contained-in">
        <nml:node idRef="urn:ogf:network:pionier.net:POZ"/>
      </nml:relation>
      <nml:relation type="contains">
        <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/11.100"/>
      </nml:relation>
    </nml:port>
  </nmwg:datum>
  <nmwg:datum>
    <nml:port id="urn:ogf:network:pionier.net:WAR:Gig1/0/2">
      <nml:name>Gig1/0/2</nml:name>
      <nml:relation type="contained-in">
        <nml:node idRef="urn:ogf:network:pionier.net:WAR"/>
      </nml:relation>
      <nml:relation type="contains">
        <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/2.100"/>
      </nml:relation>
    </nml:port>
  </nmwg:datum>
  <nmwg:datum>
    <nml:port id="urn:ogf:network:pionier.net:WAR:Gig1/0/3">
      <nml:name>Gig1/0/3</nml:name>
      <nml:relation type="contained-in">
        <nml:node idRef="urn:ogf:network:pionier.net:WAR"/>
      </nml:relation>
      <nml:relation type="contains">
        <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/3.100"/>
      </nml:relation>
    </nml:port>
  </nmwg:datum>
  <nmwg:datum>
    <nml:port id="urn:ogf:network:pionier.net:CRA:Gig1/0/5">
      <nml:name>Gig1/0/5</nml:name>
      <nml:relation type="contained-in">
        <nml:node idRef="urn:ogf:network:pionier.net:CRA"/>
      </nml:relation>
      <nml:relation type="contains">
        <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/5.100"/>
      </nml:relation>
    </nml:port>
  </nmwg:datum>
  <nmwg:datum>
    <nml:port id="urn:ogf:network:pionier.net:CRA:Gig1/0/11">
      <nml:name>Gig1/0/11</nml:name>
      <nml:relation type="contained-in">
       <nml:node idRef="urn:ogf:network:pionier.net:CRA"/>
```

```
      </nml:relation>
      <nml:relation type="contains">
        <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/11.100"/>
      </nml:relation>
    </nml:port>
  </nmwg:datum>

  <!-- vlan definitions:-->

  <nmwg:datum>
    <nml:port id="urn:ogf:network:pionier.net:POZ:Gig1/0/3.100">
      <nml:name>Gig1/0/3.100</nml:name>
      <nml-eth:vlan>100</nml-eth:vlan>
      <nml:relation type="contained-in">
        <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/3"/>
      </nml:relation>
    </nml:port>
  </nmwg:datum>
  <nmwg:datum>
    <nml:port id="urn:ogf:network:pionier.net:POZ:Gig1/0/11.100">
      <nml:name>Gig1/0/11.100</nml:name>
      <nml-eth:vlan>100</nml-eth:vlan>
      <nml:relation type="contained-in">
        <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/11"/>
      </nml:relation>
    </nml:port>
  </nmwg:datum>
  <nmwg:datum>
    <nml:port id="urn:ogf:network:pionier.net:WAR:Gig1/0/2.100">
      <nml:name>Gig1/0/2.100</nml:name>
      <nml-eth:vlan>100</nml-eth:vlan>
      <nml:relation type="contained-in">
        <nml:port idRef="urn:ogf:network:pionier.net:WAW:Gig1/0/2"/>
      </nml:relation>
    </nml:port>
  </nmwg:datum>
  <nmwg:datum>
    <nml:port id="urn:ogf:network:pionier.net:WAR:Gig1/0/3.100">
      <nml:name>Gig1/0/3.100</nml:name>
      <nml-eth:vlan>100</nml-eth:vlan>
      <nml:relation type="contained-in">
        <nml:port idRef="urn:ogf:network:pionier.net:WAW:Gig1/0/3"/>
      </nml:relation>
    </nml:port>
  </nmwg:datum>
  <nmwg:datum>
    <nml:port id="urn:ogf:network:pionier.net:CRA:Gig1/0/5.100">
      <nml:name>Gig1/0/5.100</nml:name>
      <nml-eth:vlan>100</nml-eth:vlan>
      <nml:relation type="contained-in">
        <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/5"/>
      </nml:relation>
    </nml:port>
  </nmwg:datum>
  <nmwg:datum>
    <nml:port id="urn:ogf:network:pionier.net:CRA:Gig1/0/11.100">
      <nml:name>Gig1/0/11.100</nml:name>
      <nml:vlan-eth>100</nml:vlan-eth>
      <nml:relation type="contained-in">
        <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/11"/>
      </nml:relation>
    </nml:port>
  </nmwg:datum>

  <!-- Node definitions: -->

  <nmwg:datum>
    <nml:node id="urn:ogf:network:pionier.net:POZ">
      <nml:name>POZ</nml:name>
      <nml:location>
        <nml:country>Poland</nml:country>
        <nml:city>Poznan</nml:city>
      </nml:location>
      <nml:relation type="contains">
```

```
          <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/3"/>
          <nml:port idRef="urn:ogf:network:pionier.net:POZ:Gig1/0/11"/>
        </nml:relation>
      </nml:node>
    </nmwg:datum>
    <nmwg:datum>
      <nml:node id="urn:ogf:network:pionier.net:WAR">
        <nml:name>WAR</nml:name>
        <nml:location>
          <nml:country>Poland</nml:country>
          <nml:city>Warsaw</nml:city>
        </nml:location>
        <nml:relation type="contains">
          <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/2"/>
          <nml:port idRef="urn:ogf:network:pionier.net:WAR:Gig1/0/3"/>
        </nml:relation>
      </nml:node>
    </nmwg:datum>
    <nmwg:datum>
      <nml:node id="urn:ogf:network:pionier.net:CRA">
        <nml:name>CRA</nml:name>
        <nml:location>
          <nml:country>Poland</nml:country>
          <nml:city>Cracow</nml:city>
        </nml:location>
        <nml:relation type="contains">
          <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/5"/>
          <nml:port idRef="urn:ogf:network:pionier.net:CRA:Gig1/0/11"/>
        </nml:relation>
      </nml:node>
    </nmwg:datum>

  </nmwg:data>

</nmwg:message>
```

9.2.1.12 *Step 11*

```
<?xml version='1.0' encoding='UTF-8'?>

<!-- Request message from SIP to LS -->
<!-- Step 11 -->

<nmwg:message type="LSQueryRequest"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:xquery="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/lookup/xquery/1.0/">

  <nmwg:metadata id="meta1">
    <xquery:subject id="sub1">

      declare namespace nmwg='http://ggf.org/ns/nmwg/base/2.0/';

      for $m in /nmwg:store[@type='LSStore']/nmwg:metadata
        let $eventType := $m//nmwg:parameter
            where contains($eventType,'http://ggf.org/ns/nmwg/ops/storage/measurement/2.0')
              and contains($eventType,'http://ggf.org/ns/nmwg/event/status/2.0')
              return $m

    </xquery:subject>
    <nmwg:eventType>
      http://ogf.org/ns/nmwg/tools/org/perfsonar/service/lookup/query/xquery/2.0
    </nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data metadataIdRef="meta1" id="d1"/>

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!-- Response message from LS to SIP -->
<!-- Step 11 -->

<nmwg:message type="LSQueryResponse"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:psservice="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/1.0/"
              xmlns:perfsonar="http://ggf.org/ns/nmwg/tools/org/perfsonar/1.0/">

  <nmwg:metadata id="LSQueryResponseMetadata">
    <nmwg:eventType>success.ls.query</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="LSQueryResponseData"
             metadataIdRef="LSQueryResponseMetadata">

    <psservice:datum>
      <nmwg:metadata id="http://host:8080/services/MA">
        <perfsonar:subject id="commonParameters">
          <psservice:service id="serviceParameters">
            <psservice:serviceName>SQL MA</psservice:serviceName>
            <psservice:accessPoint>http://host:8080/services/ma</psservice:accessPoint>
            <psservice:serviceType>MA</psservice:serviceType>
          </psservice:service>
        </perfsonar:subject>
        <psservice:parameters>
          <nmwg:parameter name="eventType">
            http://ggf.org/ns/nmwg/event/status/2.0
          </nmwg:parameter>
          <nmwg:parameter name="eventType">
            http://ggf.org/ns/nmwg/ops/storage/measurement/2.0
          </nmwg:parameter>
        </psservice:parameters>
      </nmwg:metadata>
    </psservice:datum>

  </nmwg:data>

</nmwg:message>
```

```
<?xml version='1.0' encoding='UTF-8'?>

<!-- Request message from SIP to LS -->
<!-- Step 11 -->

<nmwg:message type="LSQueryRequest"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:xquery="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/lookup/xquery/1.0/">

  <nmwg:metadata id="meta1">
    <xquery:subject id="sub1">

      declare namespace nmwg='http://ggf.org/ns/nmwg/base/2.0/';
      <!--
        Xquery request which returns service metadata
        for specific eventType
      -->
      for $m in /nmwg:store[@type='LSStore']/nmwg:metadata
        let $eventType := $m//nmwg:parameter
            where contains($eventType,'http://ggf.org/ns/nmwg/ops/creation/measurement/2.0')
              and contains($eventType,'http://ggf.org/ns/nmwg/event/status/2.0')
              return $m

    </xquery:subject>
    <nmwg:eventType>
      http://ogf.org/ns/nmwg/tools/org/perfsonar/service/lookup/query/xquery/2.0
    </nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="data1" metadataIdRef="meta1"/>

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Response message from LS to SIP -->
<!-- Step 11 -->

<nmwg:message type="LSQueryResponse"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:psservice="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/1.0/"
              xmlns:perfsonar="http://ggf.org/ns/nmwg/tools/org/perfsonar/1.0/">

  <nmwg:metadata id="LSQueryResponseMetadata">
    <nmwg:eventType>success.ls.query</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="LSQueryResponseData"
             metadataIdRef="LSQueryResponseMetadata">

    <psservice:datum>
      <nmwg:metadata id="http://host:8080/services/MP">
        <perfsonar:subject>
          <psservice:service id="mp_1">
            <psservice:serviceName>Link Status MP</psservice:serviceName>
            <psservice:accessPoint>http://host:8080/services/mp</psservice:accessPoint>
            <psservice:serviceType>MP</psservice:serviceType>
            <psservice:serviceDescription>
              MP contacts devices to find statuses of links
            </psservice:serviceDescription>
            <psservice:organization>PIONIER</psservice:organization>
          </psservice:service>
        </perfsonar:subject>
        <psservice:parameters>
          <nmwg:parameter name="eventType">
            http://ggf.org/ns/nmwg/event/status/2.0
          </nmwg:parameter>
          <nmwg:parameter name="eventType">
            http://ggf.org/ns/nmwg/ops/creation/measurement/2.0
          </nmwg:parameter>
        </psservice:parameters>
      </nmwg:metadata>
    </psservice:datum>

  </nmwg:data>

</nmwg:message>
```

9.2.1.13 *Step 12*

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Request message from SIP to DM -->
<!-- Step 12 -->

<nmwg:message type="MeasurementArchiveStoreRequest"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:perfsonar="http://ggf.org/ns/nmwg/tools/org/perfsonar/1.0/"
              xmlns:psservice="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/1.0/">

  <nmwg:metadata id="meta1">
    <nmwg:eventType>http://ggf.org/ns/nmwg/event/status/2.0</nmwg:eventType>
    <nmwg:eventType>http://ggf.org/ns/nmwg/ops/storage/measurement/2.0</nmwg:eventType>
    <!-- possible other metrics: -->
    <!--
    <nmwg:eventType>http://ggf.org/ns/nmwg/characteristic/utilization/2.0</nmwg:eventType>
    <nmwg:eventType>http://ggf.org/ns/nmwg/characteristic/errors/2.0</nmwg:eventType>
    -->
  </nmwg:metadata>

  <nmwg:data id="data1" metadataIdRef="meta1">
    <nmwg:metadata id="meta1">
      <perfsonar:subject id="1">
        <psservice:service id="ma 1">
          <psservice:serviceName>SQL MA</psservice:serviceName>
          <psservice:accessPoint>
            http://host:8080/services/ma
          </psservice:accessPoint>
          <psservice:serviceType>MA</psservice:serviceType>
        </psservice:service>
      </perfsonar:subject>
      <psservice:parameters>
        <nmwg:parameter name="eventType">
          http://ggf.org/ns/nmwg/event/status/2.0
        </nmwg:parameter>
        <nmwg:parameter name="eventType">
          http://ggf.org/ns/nmwg/ops/storage/measurement/2.0
        </nmwg:parameter>
      </psservice:parameters>
    </nmwg:metadata>
  </nmwg:data>

  <!--
  <nmwg:data id="data2" metadataIdRef="meta1">
    <nmwg:metadata id="meta1">
      <perfsonar:subject id="1">
        <psservice:service id="maad 1">
          <psservice:serviceName>MA-2</psservice:serviceName>
          <psservice:accessPoint>http://host-2:8080/services/ma</psservice:accessPoint>
          <psservice:serviceType>MA</psservice:serviceType>
          <psservice:serviceDescription>MA</psservice:serviceDescription>
          <psservice:organization>PIONIER</psservice:organization>
        </psservice:service>
      </perfsonar:subject>
      <psservice:parameters>
        <nmwg:parameter name="eventType">
          http://ggf.org/ns/nmwg/characteristic/utilization/2.0
        </nmwg:parameter>
        <nmwg:parameter name="eventType">
          http://ggf.org/ns/nmwg/characteristic/errors/2.0
        </nmwg:parameter>
        <nmwg:parameter name="eventType">
          http://ggf.org/ns/nmwg/ops/storage/measurement/2.0
        </nmwg:parameter>
      </psservice:parameters>
    </nmwg:metadata>
  <nmwg:data>
  -->

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Response message from SIP to DM -->
<!-- Step 12 -->

<nmwg:message type="SetupDataResponse"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/">

  <nmwg:metadata id="resultCodeMetadata">
    <nmwg:eventType>success.ma.store</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="resultDescriptionData_for_resultCodeMetadata"
                metadataIdRef="resultCodeMetadata">
    <nmwgr:datum xmlns:nmwgr="http://ggf.org/ns/nmwg/result/2.0/">
      Successfully executed request
    </nmwgr:datum>
  </nmwg:data>

</nmwg:message>
```

9.2.1.14 *Step 13*

```
<?xml version="1.0" encoding="UTF-8" ?>

<!-- Request message from SIP to MP -->
<!-- Step 13 -->

<nmwg:message type="SetupDataRequest"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:select="http://ggf.org/ns/nmwg/ops/select/2.0/"
              xmlns:store="http://ggf.org/ns/nmwg/ops/store/2.0/">

  <nmwg:metadata id="meta1">
    <nmwg:subject id="sub1">

      <!-- The content of this block depends on the implementation of MP service. -->
      <!-- Needed information can be copied from the segment descriptor. -->

    </nmwg:subject>
    <nmwg:eventType>http://ggf.org/ns/nmwg/event/status/2.0</nmwg:eventType>
    <!-- The parameter below may be useless if the segment id is already located -->
    <!-- inside the subject block -->
    <nmwg:paremeters>
      <nmwg:parameter name="segmentId">
        urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER
      </nmwg:parameter>
      <!-- one can be more parameters -->
    </nmwg:paremeters>
  </nmwg:metadata>

  <nmwg:metadata id="meta2">
    <select:subject id="iusub2" metadataIdRef="meta1"/>
    <select:parameters id="param1">
      <nmwg:parameter name="time">now</nmwg:parameter>
    </select:parameters>
    <nmwg:eventType>http://ggf.org/ns/nmwg/ops/select/2.0</nmwg:eventType>
  </nmwg:metadata>

  <!-- SIP informs MP where to store data -->
  <nmwg:metadata id="meta3">
    <store:subject id="iusub3" metadataIdRef="meta2"/>
    <store:parameters id="param1">
      <nmwg:parameter name="accessPoint">http://host:8080/services/ma</nmwg:parameter>
    </store:parameters>
    <nmwg:eventType>http://ggf.org/ns/nmwg/ops/store/2.0</nmwg:eventType>
  </nmwg:metadata>
```

```
  <nmwg:data id="data1" metadataIdRef="meta3" />

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8" ?>

<!-- Response message from MP to SIP -->
<!-- Step 13 -->

<nmwg:message type="SetupDataResponse"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/">

  <nmwg:metadata id="resultCodeMeta meta1">
    <nmwg:subject id="resultCodeSubj_meta1"
                  metadataIdRef="meta1" />
      <!-- the value of eventType below depends on the implementation of MP -->
      <nmwg:eventType>success.mp.action</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data
    id="resultCodeData_meta1"
    metadataIdRef="resultCodeMeta_meta1">
        <nmwgr:datum xmlns:nmwgr="http://ggf.org/ns/nmwg/result/2.0/">
      Successfully executed request
    </nmwgr:datum>
  </nmwg:data>

</nmwg:message>
```

9.2.1.15 *Step 14*

```
<?xml version='1.0' encoding='UTF-8'?>

<!-- Request message from MP to MA -->
<!-- Step 14 -->

<nmwg:message type="MeasurementArchiveStoreRequest"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:ifevt="http://ggf.org/ns/nmwg/event/status/base/2.0/">

  <nmwg:metadata id="meta1">
    <nmwg:subject id="sub1">

      <!-- The content of this block depends on the implementation of MP service. -->

    </nmwg:subject>
    <nmwg:eventType>http://ggf.org/ns/nmwg/event/status/2.0</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="data1" metadataIdRef="meta1">
    <ifevt:datum timeType="unix" timeValue="1170084623">
      <ifevt:stateAdmin>Unknown</ifevt:stateAdmin>
      <ifevt:stateOper>down</ifevt:stateOper>
    </ifevt:datum>
  </nmwg:data>

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Response message from MA to MP -->
<!-- Step 14 -->

<nmwg:message type="MeasurementArchiveStoreResponse"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/">

  <nmwg:metadata id="resultCodeMetadata">
    <nmwg:eventType>success.ma.write</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="resultDescriptionData_for_resultCodeMetadata"
             metadataIdRef="resultCodeMetadata">
    <nmwgr:datum xmlns:nmwgr="http://ggf.org/ns/nmwg/result/2.0/">
      Data have been stored
    </nmwgr:datum>
  </nmwg:data>

</nmwg:message>
```

9.2.1.16 *Step 15*

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Request message from DM to MA -->
<!-- Step 15 -->

<nmwg:message type="SetupDataRequest"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:select="http://ggf.org/ns/nmwg/ops/select/2.0/">

  <nmwg:metadata id="meta1">
    <nmwg:subject id="sub1">

      <!-- The content of this block depends on the implementation of MP service. -->
      <!-- Needed information can be copied from the segment descriptor. -->

    </nmwg:subject>
    <nmwg:eventType>http://ggf.org/ns/nmwg/event/status/2.0</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:metadata id="meta2">
    <select:subject id="sub2" metadataIdRef="meta1"/>
    <select:parameters id="par1">
      <nmwg:parameter name="time">now</nmwg:parameter>
    </select:parameters>
    <nmwg:eventType>http://ggf.org/ns/nmwg/ops/select/2.0</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="data1" metadataIdRef="meta2"/>

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Response message from MA to DM -->
<!-- Step 15 -->

<nmwg:message type="SetupDataResponse"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:ifevt="http://ggf.org/ns/nmwg/event/status/base/2.0/">

  <nmwg:metadata id="meta1">
    <nmwg:subject id="sub1">

      <!-- The content of this block depends on the implementation of MP service. -->
      <!-- Needed information can be copied from the segment descriptor. -->

    </nmwg:subject>
    <nmwg:eventType>http://ggf.org/ns/nmwg/event/status/2.0</nmwg:eventType>
    <!-- The parameter below may be useless if the segment id is already located -->
    <-- inside the subject block -->
    <nmwg:paremeters>
      <nmwg:parameter name="segmentId">
        urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER
      </nmwg:parameter>
      <!-- one can be more parameters -->
    </nmwg:paremeters>
  </nmwg:metadata>

  <nmwg:data id="data1" metadataIdRef="meta1">
    <ifevt:datum timeType="unix" timeValue="1170084623">
      <ifevt:stateAdmin>Unknown</ifevt:stateAdmin>
      <ifevt:stateOper>down</ifevt:stateOper>
    </ifevt:datum>
  </nmwg:data>

</nmwg:message>
```

9.2.1.17 *Step 16*

```
<?xml version='1.0' encoding='UTF-8'?>

<!-- Request message from DM to LS -->
<!-- Step 16 -->

<nmwg:message type="LSQueryRequest"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:xquery="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/lookup/xquery/1.0/">

  <nmwg:metadata id="meta1">
    <xquery:subject id="sub1">

      declare namespace nmwg='http://ggf.org/ns/nmwg/base/2.0/';
      for $m in /nmwg:store[@type='LSStore']/nmwg:metadata
        let $eventType := $m//nmwg:parameter
          where contains($eventType,'http://ggf.org/ns/nmwg/event/status/2.0')
            and contains($eventType,'http://ggf.org/ns/nmwg/ops/storage/measurement/2.0')
     and contains($eventType,'http://ggf.org/ns/nmwg/ops/storage/measurement/aggregated/autobahn/2.0')
            return $m

    </xquery:subject>
    <nmwg:eventType>
      http://ogf.org/ns/nmwg/tools/org/perfsonar/service/lookup/query/xquery/2.0
    </nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="data1" metadataIdRef="meta1"/>

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Response message from LS to DM -->
<!-- Step 16 -->

<nmwg:message type="LSQueryResponse"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:psservice="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/1.0/"
              xmlns:perfsonar="http://ggf.org/ns/nmwg/tools/org/perfsonar/1.0/">

  <nmwg:metadata id="LSQueryResponseMetadata">
    <nmwg:eventType>success.ls.query</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="LSQueryResponseData"
             metadataIdRef="LSQueryResponseMetadata">

    <psservice:datum>
      <nmwg:metadata id="http://host:8080/axis/services/ma_ad">
        <perfsonar:subject id="1">
          <psservice:service id="maad 1">
            <psservice:serviceName>Aggregated Data MA</psservice:serviceName>
            <psservice:accessPoint>http://host:8080/services/ma_ad</psservice:accessPoint>
            <psservice:serviceType>MA</psservice:serviceType>
            <psservice:serviceDescription>E2E Circuit Aggregated Data</psservice:serviceDescription>
            <psservice:organization>PIONIER</psservice:organization>
          </psservice:service>
        </perfsonar:subject>
        <psservice:parameters>
          <nmwg:parameter name="eventType">
            http://ggf.org/ns/nmwg/event/status/2.0
          </nmwg:parameter>
          <nmwg:parameter name="eventType">
            http://ggf.org/ns/nmwg/ops/storage/measurement/2.0
          </nmwg:parameter>
          <nmwg:parameter name="eventType">
            http://ggf.org/ns/nmwg/ops/storage/measurement/aggregated/autobahn/2.0
          </nmwg:parameter>
        </psservice:parameters>
      </nmwg:metadata>
    </psservice:datum>

  </nmwg:data>

</nmwg:message>
```

9.2.1.18 *Step 17*

```
<?xml version='1.0' encoding='UTF-8'?>

<!-- Request message from DM to MA(ad) -->
<!-- Step 17 -->

<nmwg:message type="MeasurementArchiveStoreRequest"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:ifevt="http://ggf.org/ns/nmwg/event/status/base/2.0/"
              xmlns:nml="http://schemas.ogf.org/nml/base/201103/beta/">

  <nmwg:metadata id="meta1">
    <nmwg:subject id="sub1">
      <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER:10.10.64.1">
        <nml:name>10.10.64.1</nml:name>
        <nml:relation type="next">
          <nml:link idRef="urn:ogf:network:geant.net:PIONIER@1275987204 res 1:PIONIER:10.10.64.2"/>
        </nml:relation>
        <nml:relation type="contained-in">
          <nml:link idRef="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER"/>
        </nml:relation>
      </nml:link>
    </nmwg:subject>
    <nmwg:eventType>http://ggf.org/ns/nmwg/event/status/2.0</nmwg:eventType>
```

```
    <nmwg:eventType>http://ggf.org/ns/nmwg/ops/aggregation/autobahn/2.0</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="data1" metadataIdRef="meta1">
    <ifevt:datum timeType="unix" timeValue="1170084623">
      <ifevt:stateAdmin>Unknown</ifevt:stateAdmin>
      <ifevt:stateOper>up</ifevt:stateOper>
    </ifevt:datum>
  </nmwg:data>

  <nmwg:metadata id="meta2">
    <nmwg:subject id="sub2">
      <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER:10.10.64.2">
        <nml:name>10.10.64.2</nml:name>
        <nml:relation type="next">
          <nml:link idRef="urn:ogf:network:geant.net:PIONIER@1275987204 res 1:PIONIER:10.10.64.3"/>
        </nml:relation>
        <nml:relation type="contained-in">
          <nml:link idRef="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER"/>
        </nml:relation>
      </nml:link>
    </nmwg:subject>
    <nmwg:eventType>http://ggf.org/ns/nmwg/event/status/2.0</nmwg:eventType>
    <nmwg:eventType>http://ggf.org/ns/nmwg/ops/aggregation/autobahn/2.0</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="data2" metadataIdRef="meta2">
    <ifevt:datum timeType="unix" timeValue="1170084623">
      <ifevt:stateAdmin>Unknown</ifevt:stateAdmin>
      <ifevt:stateOper>up</ifevt:stateOper>
    </ifevt:datum>
  </nmwg:data>

  <nmwg:metadata id="meta3">
    <nmwg:subject id="sub3">
      <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER:10.10.64.3">
        <nml:name>10.10.64.3</nml:name>
        <nml:relation type="contained-in">
          <nml:link idRef="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER"/>
        </nml:relation>
      </nml:link>
    </nmwg:subject>
    <nmwg:eventType>http://ggf.org/ns/nmwg/event/status/2.0</nmwg:eventType>
    <nmwg:eventType>http://ggf.org/ns/nmwg/ops/aggregation/autobahn/2.0</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="data3" metadataIdRef="meta3">
    <ifevt:datum timeType="unix" timeValue="1170084623">
      <ifevt:stateAdmin>Unknown</ifevt:stateAdmin>
      <ifevt:stateOper>up</ifevt:stateOper>
    </ifevt:datum>
  </nmwg:data>

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Response message from MA(ad) to DM -->
<!-- Step 17 -->

<nmwg:message type="MeasurementArchiveStoreResponse"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/">

  <nmwg:metadata id="resultCodeMetadata">
    <nmwg:eventType>success.ma.write</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="resultDescriptionData_for_resultCodeMetadata"
             metadataIdRef="resultCodeMetadata">
    <nmwgr:datum xmlns:nmwgr="http://ggf.org/ns/nmwg/result/2.0/">
      Data have been stored
    </nmwgr:datum>
  </nmwg:data>

</nmwg:message>
```

9.2.2 External consumer

The message examples in the following sections refer to the workflow steps presented in section 5.3.2.

9.2.2.1 *Step 1*

```
<?xml version='1.0' encoding='UTF-8'?>

<!-- Request message from VC to gLS -->
<!-- Step 1 -->

<nmwg:message type="LSQueryRequest"
          xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
          xmlns:summary="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/lookup/summarization/2.0/"
          xmlns:nml="http://schemas.ogf.org/nml/base/201103/beta/">

  <nmwg:metadata id="meta1">
    <summary:subject>
      <nml:domain id=”pionier.net”/>
    </summary:subject>
    <nmwg:eventType>
      http://ogf.org/ns/nmwg/tools/org/perfsonar/service/lookup/discovery/summary/2.0
    </nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="data1" metadataIdRef="meta1"/>

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Response message from gLS to VC -->
<!-- Step 1 -->

<nmwg:message type="LSQueryResponse"
          xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
          xmlns:perfsonar="http://ggf.org/ns/nmwg/tools/org/perfsonar/1.0/"
          xmlns:summary="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/lookup/summarization/2.0/"
          xmlns:psservice="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/1.0/"
          xmlns:nml="http://schemas.ogf.org/nml/base/201103/beta/">

  <nmwg:metadata id="meta1">
    <summary:subject>
      <nml:domain id=”pionier.net”/>
    </summary:subject>
    <nmwg:eventType>
      http://ogf.org/ns/nmwg/tools/org/perfsonar/service/lookup/discovery/summary/2.0
    </nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="data1"
             metadataIdRef="meta1">
    <psservice:datum>
      <nmwg:metadata id="http://host:8080/services/ls">
        <perfsonar:subject>
          <psservice:service>
            <psservice:serviceName>PIONIER LS</psservice:serviceName>
            <psservice:accessPoint>http://host:8080/services/ls</psservice:accessPoint>
            <psservice:serviceType>LS</psservice:serviceType>
            <psservice:serviceDescription>PIONIER LS</psservice:serviceDescription>
            <psservice:organization>PIONIER</psservice:organization>
          </psservice:service>
        </perfsonar:subject>
      </nmwg:metadata>
```

```
    </psservice:datum>
  </nmwg:data>

</nmwg:message>
```

9.2.2.2 *Step 2*

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Request message from VC to hLS -->
<!-- Step 2 -->

<nmwg:message type="LSQueryRequest"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:xquery="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/lookup/xquery/1.0/">

  <nmwg:metadata id="meta1">
    <xquery:subject id="sub1">

      <!-- xquery searching the MA(at) -->

      declare namespace nmwg='http://ggf.org/ns/nmwg/base/2.0/';

      for $m in /nmwg:store[@type='LSStore']/nmwg:metadata
        let $eventType := $m//nmwg:parameter
          where contains($eventType,'http://ggf.org/ns/nmwg/ops/storage/topology/aggregated/2.0')
            return $m

    </xquery:subject>
    <nmwg:eventType>
      http://ogf.org/ns/nmwg/tools/org/perfsonar/service/lookup/query/xquery/2.0
    </nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data metadataIdRef="meta1" id="d1"/>

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Response message from hLS to VC -->
<!-- Step 2 -->

<nmwg:message type="LSQueryResponse"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:perfsonar="http://ggf.org/ns/nmwg/tools/org/perfsonar/1.0/"
              xmlns:psservice="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/1.0/">

  <nmwg:metadata id="LSQueryResponseMetadata">
    <nmwg:eventType>success.ls.query</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="LSQueryResponseData"
             metadataIdRef="LSQueryResponseMetadata">
    <psservice:datum>
      <nmwg:metadata id="http://host:8080/services/ma at">
        <perfsonar:subject>
          <psservice:service>
            <psservice:serviceName>PIONIER Aggregated Topology MA</psservice:serviceName>
            <psservice:accessPoint>http://host:8080/services/ma at</psservice:accessPoint>
            <psservice:serviceType>MA</psservice:serviceType>
            <psservice:serviceDescription>
              PIONIER Aggregated Topology MA
            </psservice:serviceDescription>
            <psservice:organization>PIONIER</psservice:organization>
          </psservice:service>
        </perfsonar:subject>
        <psservice:parameters id="p1">
          <nmwg:parameter name="eventType">
```

```
            http://ggf.org/ns/nmwg/ops/storage/topology/aggregated/2.0
          </nmwg:parameter>
        </psservice:parameters>
      </nmwg:metadata>
    </psservice:datum>
  </nmwg:data>

</nmwg:message>
```

9.2.2.3 *Step 3*

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Request message from VC to MA(at) -->
<!-- Step 3 -->

<nmwg:message type="TSQueryRequest"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:xquery="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/lookup/xquery/1.0/">

  <nmwg:metadata id="meta1">
    <xquery:subject id="sub0">

    <!-- xquery statement that fetches the reservation descriptor with a given domain name -->

      declare namespace nmwg='http://ggf.org/ns/nmwg/base/2.0/';
      declare namespace nml='http://schemas.ogf.org/nml/base/201103/beta/';

      for $lk in /nml:topology/nml:link
        where contains($lk/@id, "pionier.net")
          return $lk

    </xquery:subject>
    <!-- or we can use xpath -->
    <!--
    <xpath:subject xmlns:xpath="http://ggf.org/ns/nmwg/tools/org/perfsonar/xpath/1.0/">
      /*:topology/*:link[contains(@id,"pionier.net")]
    </xpath:subject>
    -->
    <nmwg:eventType>http://ggf.org/ns/nmwg/topology/201103</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data metadataIdRef="meta1" id="data1" />

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Response message from MA(at) to VC -->
<!-- Step 3 -->

<nmwg:message type="TSQueryResponse"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:nml="http://schemas.ogf.org/nml/base/201103/beta/"
              xmlns:xquery="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/lookup/xquery/1.0/">

  <nmwg:metadata id="meta1">
    <xquery:subject id="sub0">
      declare namespace nmwg='http://ggf.org/ns/nmwg/base/2.0/';
      declare namespace nml='http://schemas.ogf.org/nml/base/201103/beta/';
      for $lk in /nml:topology/nml:link
        where contains($lk/@id, "pionier.net")
          return $lk
    </xquery:subject>
    <nmwg:eventType>http://ggf.org/ns/nmwg/topology/201103</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data metadataIdRef="meta1" id="data1" />
    <nmwg:datum>
```

```
      <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1">
        <nml:relation type="serialcompound">
          <nml:link idRef="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER">
            <nml:relation type="next">
              <nml:link idRef="urn:ogf:network:geant.net:PIONIER@1275987204 res 1:GEANT" />
            </nml:relation>
          </nml:link>
          <nml:link idRef="urn:ogf:network:geant.net:PIONIER@1275987204_res_1:GEANT">
            <nml:relation type="next">
              <nml:link idRef="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:HEANET" />
            </nml:relation>
          </nml:link>
          <nml:link idRef="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:HEANET" />
        </nml:relation>
      </nml:link>
    </nmwg:datum>
  </nmwg:data>

</nmwg:message>
```

9.2.2.4 *Step 4*

```
<?xml version='1.0' encoding='UTF-8'?>

<!-- Request message from VC to gLS -->
<!-- Step 4 -->

<nmwg:message type="LSQueryRequest"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"

xmlns:summary="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/lookup/summarization/2.0/"
              xmlns:nml="http://schemas.ogf.org/nml/base/201103/beta/">

  <nmwg:metadata id="meta1">
    <summary:subject>
      <nml:domain id=”pionier.net” />
      <nml:domain id=”geant.net” />
      <nml:domain id=”heanet.net” />
    </summary:subject>
    <nmwg:eventType>
      http://ogf.org/ns/nmwg/tools/org/perfsonar/service/lookup/discovery/summary/2.0
    </nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="data1" metadataIdRef="meta1"/>

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Response message from gLS to VC -->
<!-- Step 4 -->

<nmwg:message type="LSQueryResponse"
          xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
          xmlns:perfsonar="http://ggf.org/ns/nmwg/tools/org/perfsonar/1.0/"
          xmlns:summary="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/lookup/summarization/2.0/"
          xmlns:psservice="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/1.0/"
          xmlns:nml="http://schemas.ogf.org/nml/base/201103/beta/">

  <nmwg:metadata id="meta1">
    <summary:subject>
      <nml:domain id=”pionier.net” />
    </summary:subject>
    <nmwg:eventType>
      http://ogf.org/ns/nmwg/tools/org/perfsonar/service/lookup/discovery/summary/2.0
    </nmwg:eventType>
  </nmwg:metadata>
```

```
  <nmwg:data id="data1"
             metadataIdRef="meta1">
    <psservice:datum>
      <nmwg:metadata id="http://host:8080/services/ls">
        <perfsonar:subject>
          <psservice:service>
            <psservice:serviceName>PIONIER LS</psservice:serviceName>
            <psservice:accessPoint>http://host:8080/services/ls</psservice:accessPoint>
            <psservice:serviceType>LS</psservice:serviceType>
            <psservice:serviceDescription>PIONIER LS</psservice:serviceDescription>
            <psservice:organization>PIONIER</psservice:organization>
          </psservice:service>
        </perfsonar:subject>
      </nmwg:metadata>
    </psservice:datum>
  </nmwg:data>

  <nmwg:metadata id="meta2">
    <summary:subject>
      <nml:domain id=”geant.net” />
    </summary:subject>
    <nmwg:eventType>
      http://ogf.org/ns/nmwg/tools/org/perfsonar/service/lookup/discovery/summary/2.0
    </nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="data2"
             metadataIdRef="meta2">
    <psservice:datum>
      <nmwg:metadata id="http://geanthost:8080/services/ls">
        <perfsonar:subject>
          <psservice:service>
            <psservice:serviceName>GEANT LS</psservice:serviceName>
            <psservice:accessPoint>http://geanthost:8080/services/ls</psservice:accessPoint>
            <psservice:serviceType>LS</psservice:serviceType>
            <psservice:serviceDescription>GEANT LS</psservice:serviceDescription>
            <psservice:organization>GEANT</psservice:organization>
          </psservice:service>
        </perfsonar:subject>
      </nmwg:metadata>
    </psservice:datum>
  </nmwg:data>

  <nmwg:metadata id="meta3">
    <summary:subject>
      <nml:domain id=”heanet.net” />
    </summary:subject>
    <nmwg:eventType>
      http://ogf.org/ns/nmwg/tools/org/perfsonar/service/lookup/discovery/summary/2.0
    </nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="data3"
             metadataIdRef="meta3">
    <psservice:datum>
      <nmwg:metadata id="http://heanethost:8080/services/ls">
        <perfsonar:subject>
          <psservice:service>
            <psservice:serviceName>HEANET LS</psservice:serviceName>
            <psservice:accessPoint>http://heanethost:8080/services/ls</psservice:accessPoint>
            <psservice:serviceType>LS</psservice:serviceType>
            <psservice:serviceDescription>HEANET LS</psservice:serviceDescription>
            <psservice:organization>HEANET</psservice:organization>
          </psservice:service>
        </perfsonar:subject>
      </nmwg:metadata>
    </psservice:datum>
  </nmwg:data>


</nmwg:message>
```

9.2.2.5 *Step 5*

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Request message from VC to hLS -->
<!-- Step 5 -->

<nmwg:message type="LSQueryRequest"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:xquery="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/lookup/xquery/1.0/">

  <nmwg:metadata id="meta1">
    <xquery:subject id="sub1">

      <!-- xquery searching the MA(at) -->

      declare namespace nmwg='http://ggf.org/ns/nmwg/base/2.0/';

      for $m in /nmwg:store[@type='LSStore']/nmwg:metadata
        let $eventType := $m//nmwg:parameter
          where contains($eventType,'http://ggf.org/ns/nmwg/ops/storage/topology/aggregated/2.0')
            return $m

    </xquery:subject>
    <nmwg:eventType>
      http://ogf.org/ns/nmwg/tools/org/perfsonar/service/lookup/query/xquery/2.0
    </nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data metadataIdRef="meta1" id="d1"/>

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Response message from hLS to VC -->
<!-- Step 5 -->

<nmwg:message type="LSQueryResponse"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:perfsonar="http://ggf.org/ns/nmwg/tools/org/perfsonar/1.0/"
              xmlns:psservice="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/1.0/">

  <nmwg:metadata id="LSQueryResponseMetadata">
    <nmwg:eventType>success.ls.query</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="LSQueryResponseData"
             metadataIdRef="LSQueryResponseMetadata">
    <psservice:datum >
      <nmwg:metadata id="http://host:8080/services/ma_at">
        <perfsonar:subject>
          <psservice:service>
            <psservice:serviceName>PIONIER Aggregated Topology MA</psservice:serviceName>
            <psservice:accessPoint>http://host:8080/services/ma at</psservice:accessPoint>
            <psservice:serviceType>MA</psservice:serviceType>
            <psservice:serviceDescription>
              PIONIER Aggregated Topology MA
            </psservice:serviceDescription>
            <psservice:organization>PIONIER</psservice:organization>
          </psservice:service>
        </perfsonar:subject>
        <psservice:parameters>
          <nmwg:parameter name="eventType">
            http://ggf.org/ns/nmwg/ops/storage/topology/aggregated/2.0
          </nmwg:parameter>
        </psservice:parameters>
      </nmwg:metadata>
    </psservice:datum>
  </nmwg:data>
```

```
</nmwg:message>
```

9.2.2.6 *Step 6*

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Request message from VC to MA(at) -->
<!-- Step 6 -->

<nmwg:message type="TSQueryRequest"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:xquery="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/lookup/xquery/1.0/">

  <nmwg:metadata id="meta1">
    <xquery:subject id="sub0">

      <!-- xquery statement that fetches the aggregated segment descriptor  -->
      <!-- with a given segment identifier -->

      declare namespace nmwg='http://ggf.org/ns/nmwg/base/2.0/';
      declare namespace nml='http://schemas.ogf.org/nml/base/201103/beta/';

      for $lk in /nml:topology/nml:link
        where contains($lk/@id, "urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER")
          return $lk

    </xquery:subject>
    <!-- or we can use xpath -->
    <!--
    <xpath:subject xmlns:xpath="http://ggf.org/ns/nmwg/tools/org/perfsonar/xpath/1.0/">
      /*:topology/*:link[contains(@id,"urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER")]
    </xpath:subject>
    -->
    <nmwg:eventType>http://ggf.org/ns/nmwg/topology/201103</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data metadataIdRef="meta1" id="data1" />

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Response message from MA(at) to VC -->
<!-- Step 6 -->

<nmwg:message type="TSQueryResponse"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:nml="http://schemas.ogf.org/nml/base/201103/beta/"
              xmlns:xquery="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/lookup/xquery/1.0/">

  <nmwg:metadata id="meta1">
    <xquery:subject id="sub0">
      declare namespace nmwg='http://ggf.org/ns/nmwg/base/2.0/';
      declare namespace nml='http://schemas.ogf.org/nml/base/201103/beta/';
      for $lk in /nml:topology/nml:link
        where contains($lk/@id, "urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER")
          return $lk
    </xquery:subject>
    <nmwg:eventType>http://ggf.org/ns/nmwg/topology/201103</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data metadataIdRef="meta1" id="data1" >

    <nmwg:datum>
      <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER">
        <nml:relation type="bidirectional">
          <nml:link idRef="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER AtoB"/>
          <nml:link idRef="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER_BtoA"/>
```

```
        </nml:relation>
      </nml:link>
    </nmwg:datum>

    <nmwg:datum>
      <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER_AtoB">
        <nml:relation type="serialcompound">
          <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER:10.10.64.1">
            <nml:name>10.10.64.1</nml:name>
            <nml:relation type="next">
             <nml:link idRef="urn:ogf:network:geant.net:PIONIER@1275987204 res 1:PIONIER:10.10.64.2"/>
            </nml:relation>
          </nml:link>
          <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER:10.10.64.2">
            <nml:name>10.10.64.2</nml:name>
            <nml:relation type="next">
             <nml:link idRef="urn:ogf:network:geant.net:PIONIER@1275987204_res_1:PIONIER:10.10.64.3"/>
            </nml:relation>
          </nml:link>
          <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER:10.10.64.3">
            <nml:name>10.10.64.3</nml:name>
          </nml:link>
        </nml:relation>
      </nml:link>
    </nmwg:datum>

    <nmwg:datum>
      <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204_res_1:PIONIER_BtoA">
        <nml:relation type="serialcompound">
          <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER:10.10.64.3">
            <nml:name>10.10.64.3</nml:name>
            <nml:relation type="next">
             <nml:link idRef="urn:ogf:network:geant.net:PIONIER@1275987204 res 1:PIONIER:10.10.64.2"/>
            </nml:relation>
          </nml:link>
          <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER:10.10.64.2">
            <nml:name>10.10.64.2</nml:name>
            <nml:relation type="next">
             <nml:link idRef="urn:ogf:network:geant.net:PIONIER@1275987204_res_1:PIONIER:10.10.64.1"/>
            </nml:relation>
          </nml:link>
          <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER:10.10.64.1">
            <nml:name>10.10.64.1</nml:name>
          </nml:link>
        </nml:relation>
      </nml:link>
    </nmwg:datum>

  </nmwg:data>

</nmwg:message>
```

9.2.2.7 *Step 8*

```
<?xml version='1.0' encoding='UTF-8'?>

<!-- Request message from VC to hLS -->
<!-- Step 8 -->

<nmwg:message type="LSQueryRequest"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:xquery="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/lookup/xquery/1.0/">

  <nmwg:metadata id="meta1">
    <xquery:subject id="sub1">

      declare namespace nmwg='http://ggf.org/ns/nmwg/base/2.0/';

      for $m in /nmwg:store[@type='LSStore']/nmwg:metadata
        let $eventType := $m//nmwg:parameter
        where contains($eventType,'http://ggf.org/ns/nmwg/event/status/2.0')
```

```
          and contains($eventType,'http://ggf.org/ns/nmwg/ops/storage/measurement/2.0')
     and contains($eventType,'http://ggf.org/ns/nmwg/ops/storage/measurement/aggregated/autobahn/2.0')
               return $m

    </xquery:subject>
    <nmwg:eventType>
      http://ogf.org/ns/nmwg/tools/org/perfsonar/service/lookup/query/xquery/2.0
    </nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="data1" metadataIdRef="meta1"/>

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Response message from hLS to VC -->
<!-- Step 8 -->

<nmwg:message type="LSQueryResponse"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:perfsonar="http://ggf.org/ns/nmwg/tools/org/perfsonar/1.0/"
              xmlns:psservice="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/1.0/">

  <nmwg:metadata id="LSQueryResponseMetadata">
    <nmwg:eventType>success.ls.query</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="LSQueryResponseData"
             metadataIdRef="LSQueryResponseMetadata">

    <psservice:datum>
      <nmwg:metadata id="http://host:8080/services/ma_ad">
        <perfsonar:subject>
          <psservice:service>
            <psservice:serviceName>
              PIONIER Aggregated Data MA
            </psservice:serviceName>
            <psservice:accessPoint>
              http://host:8080/services/ma ad
            </psservice:accessPoint>
            <psservice:serviceType>MA</psservice:serviceType>
            <psservice:serviceDescription>
              PIONIER Segment Aggregated Data
            </psservice:serviceDescription>
            <psservice:organization>PIONIER</psservice:organization>
          </psservice:service>
        </perfsonar:subject>
        <psservice:parameters>
          <nmwg:parameter name="eventType">
            http://ggf.org/ns/nmwg/event/status/2.0
          </nmwg:parameter>
          <nmwg:parameter name="eventType">
            http://ggf.org/ns/nmwg/ops/storage/measurement/2.0
          </nmwg:parameter>
          <nmwg:parameter name="eventType">
            http://ggf.org/ns/nmwg/ops/storage/measurement/aggregated/autobahn/2.0
          </nmwg:parameter>
        </psservice:parameters>
      </nmwg:metadata>
    </psservice:datum>

  </nmwg:data>

</nmwg:message>
```

9.2.2.8 *Step 9*

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Request message from VC to MA(ad) -->
<!-- Step 9 -->

<nmwg:message type="SetupDataRequest"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:select="http://ggf.org/ns/nmwg/ops/select/2.0/"
              xmlns:nml="http://schemas.ogf.org/nml/base/201103/beta/">

  <nmwg:metadata id="meta1">
    <nmwg:subject id="sub1">

      <!-- The content of this block depends on the implementation of MA(ad) service. -->
      <!-- Needed information can be copied from the segment descriptor. -->
      <!-- Example: -->
      <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER"/>

    </nmwg:subject>
    <nmwg:eventType>http://ggf.org/ns/nmwg/event/status/2.0</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:metadata id="meta2">
    <select:subject id="sub2" metadataIdRef="meta1"/>
    <select:parameters id="par1">
      <nmwg:parameter name="time">now</nmwg:parameter>
    </select:parameters>
    <nmwg:eventType>http://ggf.org/ns/nmwg/ops/select/2.0</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="data1" metadataIdRef="meta2"/>

</nmwg:message>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Response message from MA(ad) to VC -->
<!-- Step 9 -->

<nmwg:message type="SetupDataResponse"
              xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
              xmlns:ifevt="http://ggf.org/ns/nmwg/event/status/base/2.0/"
              xmlns:nml="http://schemas.ogf.org/nml/base/201103/beta/">

  <nmwg:metadata id="meta1">
    <nmwg:subject id="sub1">
      <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER:10.10.64.1">
        <nml:name>10.10.64.1</nml:name>
        <nml:relation type="next">
          <nml:link idRef="urn:ogf:network:geant.net:PIONIER@1275987204_res_1:PIONIER:10.10.64.2" />
        </nml:relation>
        <nml:relation type="contained-in">
          <nml:link idRef="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER"/>
        </nml:relation>
      </nml:link>
    </nmwg:subject>
    <nmwg:eventType>http://ggf.org/ns/nmwg/event/status/2.0</nmwg:eventType>
    <nmwg:eventType>http://ggf.org/ns/nmwg/ops/aggregation/autobahn/2.0</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="data1" metadataIdRef="meta1">
    <ifevt:datum timeType="unix" timeValue="1170084623">
      <ifevt:stateAdmin>Unknown</ifevt:stateAdmin>
      <ifevt:stateOper>up</ifevt:stateOper>
    </ifevt:datum>
  </nmwg:data>
```

```
  <nmwg:metadata id="meta2">
    <nmwg:subject id="sub2">
      <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER:10.10.64.2">
        <nml:name>10.10.64.2</nml:name>
        <nml:relation type="next">
          <nml:link idRef="urn:ogf:network:geant.net:PIONIER@1275987204_res_1:PIONIER:10.10.64.3" />
        </nml:relation>
        <nml:relation type="contained-in">
          <nml:link idRef="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER"/>
        </nml:relation>
      </nml:link>
    </nmwg:subject>
    <nmwg:eventType>http://ggf.org/ns/nmwg/event/status/2.0</nmwg:eventType>
    <nmwg:eventType>http://ggf.org/ns/nmwg/ops/aggregation/autobahn/2.0</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="data2" metadataIdRef="meta2">
    <ifevt:datum timeType="unix" timeValue="1170084623">
      <ifevt:stateAdmin>Unknown</ifevt:stateAdmin>
      <ifevt:stateOper>up</ifevt:stateOper>
    </ifevt:datum>
  </nmwg:data>

  <nmwg:metadata id="meta3">
    <nmwg:subject id="sub3">
      <nml:link id="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER:10.10.64.3">
        <nml:name>10.10.64.3</nml:name>
        <nml:relation type="contained-in">
          <nml:link idRef="urn:ogf:network:pionier.net:PIONIER@1275987204 res 1:PIONIER"/>
        </nml:relation>
      </nml:link>
    </nmwg:subject>
    <nmwg:eventType>http://ggf.org/ns/nmwg/event/status/2.0</nmwg:eventType>
    <nmwg:eventType>http://ggf.org/ns/nmwg/ops/aggregation/autobahn/2.0</nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="data3" metadataIdRef="meta3">
    <ifevt:datum timeType="unix" timeValue="1170084623">
      <ifevt:stateAdmin>Unknown</ifevt:stateAdmin>
      <ifevt:stateOper>up</ifevt:stateOper>
    </ifevt:datum>
  </nmwg:data>

</nmwg:message>
```

9.2.3 Communication between the hLS and the gLS

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Request message from hLS to gLS-->

<!-- summarisation -->

<nmwg:message type="LSRegisterRequest"
          xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
          xmlns:perfsonar="http://ggf.org/ns/nmwg/tools/org/perfsonar/1.0/"
          xmlns:psservice="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/1.0/"
          xmlns:summary="http://ggf.org/ns/nmwg/tools/org/perfsonar/service/lookup/summarization/2.0/"
          xmlns:nml="http://schemas.ogf.org/nml/base/201103/beta/">

  <nmwg:metadata id="meta1">
    <perfsonar:subject>
      <psservice:service>
        <psservice:serviceName>PIONIER LS</psservice:serviceName>
        <psservice:accessPoint>http://host:8080/services/ls</psservice:accessPoint>
        <psservice:serviceType>LS</psservice:serviceType>
        <psservice:serviceDescription>PIONIER home LS</psservice:serviceDescription>
        <psservice:organization>PIONIER</psservice:organization>
      </psservice:service>
    <nmwg:eventType>
      http://ogf.org/ns/nmwg/tools/org/perfsonar/service/lookup/registration/service/2.0
    </nmwg:eventType>
  </nmwg:metadata>

  <nmwg:data id="data1" metadataIdRef="meta1">
    <nmwg:metadata>
      <summary:subject>

        <nml:domain id="pionier.net" />

        <nmwg:eventType>http://ggf.org/ns/nmwg/topology/201103</nmwg:eventType>
        <nmwg:eventType>http://ggf.org/ns/nmwg/topology/aggregated/autobahn/2.0</nmwg:eventType>

      </summary:subject>
      <nmwg:eventType>
        http://ogf.org/ns/nmwg/tools/org/perfsonar/service/lookup/discovery/summary/2.0
      </nmwg:eventType>
    <nmwg:metadata>
  </nmwg:data>

</nmwg:message>
```

Appendix A Implementation Considerations

A.1 Visualisation Tool and E2EMon Usage

As during GN2 an E2EMon monitoring solution was developed for AutoBAHN, it seems reasonable to use and update it for data visualisation.

E2EMon can display measurement data within each per-domain segment, as well as the status of an end-to-end circuit. As such, it is suitable for displaying aggregated network status information gathered by perfSONAR. It can be integrated with the AutoBAHN Client Portal to provide users with a single management console that can be used to set up required paths and to check their performance.

It is suggested to use E2EMon for presenting data from the designated perfSONAR Measurement Archive, MA(ad), to users.

A.2 Changes to perfSONAR Services

The integration of the AutoBAHN and perfSONAR systems requires some modifications to be made to perfSONAR services. Some of them are minor (e.g. addition of new fields), but the modifications that have to be made to MP (e.g.: SSH/Telnet MP) services are significant. SSH/Telnet MP registers information with the LS that is not useful for the integration. This problem has to be analysed during the implementation phase. The SNMP MP service is currently evaluated by SA2 Task 3 to match the multi-domain monitoring use case and become part of the perfSONAR MDM service.

While deciding which MP services could be used for the pilot deployment, it is suggested to consider also services implemented in Perl by US partners (perfSONAR-PS [perfSONAR-PS]

A.3 Monitoring the Edge Domain Nodes

This section contains updated examples of messages (compared with the original steps in the section 9.2) which refer to the second use case described in section **Błąd! Nie można odnaleźć źródła odwołania.** (the MP service only has access to one end of a link; the second end is located in the neigbouring domain and protected by its administrators).

Not all messages require modifications to support the use case.

A.3.1 Messages

TODO

Appendix B The General Standardised Client Access

The following workflow is the result of discussion between Internet2 and GN3 partners and the existing form was formulated by Aaron Brown (Internet2). It is the general procedure without any assumptions allowing acces to the measurement data by the client application[8].

Step 1: The client contacts one or more gLS(es) and asks which hLS(es) know about services containing circuit descriptors for the domain in the circuit identifier.

Step 2: The gLS(es) responds to the client with a list of hLS(es).

Step 3: The client contacts each hLS (in parallel or in serial), and asks for services containing circuit descriptors for the domain.

Step 4: The hLS(es) respond to the client with a list of TS(es).

Step 5: The client contacts each TS (in parallel or in serial), and asks for the circuit descriptor.

Step 6: The TS(es) respond to the client with the circuit descriptor.

Step 7: The client contacts one or more gLS(es) and asks which hLS(es) know about services containing segment descriptors for the domains in each of the segment identifiers.

Step 8: The gLS(es) respond to the client with a list of hLS(es) for each domain.

Step 9: The client contacts each hLS (in parallel or in serial), and asks for the services containing the segment descriptors.

Step 10: The hLS(es) respond to the client with a list of TS(es).

Step 11: The client contacts each TS (in parallel or in serial), and asks for the segment descriptors.

Step 12: The TS(es) respond with the segment descriptors.

[8] Terminology in the PS-pS solution is different for two elements. The GN3 solution uses the names “reservation descriptor” and “MA(at)” instead of “circuit descriptor” and TS.

Step 13: The client contacts one or more gLS(es) and asks which hLS(es) know about services containing the topology information pointed to by the segment descriptors.

Step 14: The gLS(es) responds to the client with a list of hLS(es).

Step 15: The client contacts each hLS (in parallel or in serial), and asks for the services containing topology information for the domain.

Step 16: The hLS(es) respond to the client with a list of TS(es).

Step 17: The client contacts each TS (in parallel or in serial), and asks for the topology information referenced by the segment descriptors.

Step 18: The TS(es) respond to the client with the topology information

Step 19: The client contacts one or more gLS(es) and asks which hLS(es) know about services containing measurement information for the domains in the topology elements.

Step 20: The gLS(es) respond to the client with a list of hLS(es).

Step 21: The client contacts each hLS (in parallel or in serial), and asks for the services containing measurement information for the topology elements.

Step 22: The hLS(es) respond with a list of MA(s).

Step 23: The client contacts each MA (in parallel or in serial), and asks for measurement information about the topology elements.

Step 24: The MA(s) respond with measurement information about the topology elements.

References

[perfSONAR]	perfSONAR project home page http://www.perfsonar.net/
[AutoBAHN]	GN2 website containing information about AutoBAHN project http://www.geant2.net/server/show/nav.756& https://forge.geant.net/forge/display/autobahn
[GN2]	GN2 project home page http://www.geant2.net/
[GN3]	GN3 project home page http://www.geant.net/pages/home.aspx
[DICE]	DICE collaboration home page http://wiki.geant.net/bin/view/DICE http://www.geant2.net/server/show/conWebDoc.1308
[GLIF]	Global Lambda Integrated Facility home page http://www.glif.is/
[ABClientPortal]	Information about the AutoBAHN Client Portal https://forge.geant.net/forge/display/autobahn/Screenshots
[ABAPI]	Information about the AutoBAHN API https://intranet.geant.net/sites/Services/SA2/T5/Documents/uap-interface-description.doc
[IDM]	Definition of Bandwidth on Demand Framework and General Architecture http://www.geant2.net/upload/pdf/GN2-05-208v7_DJ3-3-1_GEANT2_Initial_Bandwidth_on_Demand_Framework_and_Architecture.pdf
[cNIS]	cNIS project home page http://cnis.geant.net/
[MDM]	MDM release notes https://wiki.man.poznan.pl/perfsonar-mdm/index.php/PerfSONAR_v3.2 https://wiki.man.poznan.pl/perfsonar-mdm/index.php/PerfSONAR-Bundle-Release-Notes#perfSONAR_LHCOPN
[MDMAdminGuide]	perfSONAR MDM 3.2 Administrator's Guide http://www.perfsonar.net/download/bundles/perfsonar-mdm-bundle/3.2/perfSONAR_MDM_3.2_Admin_Guide_1.2.pdf
[MRTG]	The Multi Router Traffic Grapher project home page http://oss.oetiker.ch/mrtg/
[RelaxNG]	Releax NG schema language home page http://www.relaxng.org/
[WSDL]	WSDL language home page http://www.w3.org/TR/wsdl
[perfSONAR-PS]	perfSONAR-PS project home page http://psps.perfsonar.net/

Acronyms

API	Application Programming Interface
AutoBAHN	Automated Bandwidth Allocation across Heterogeneous Networks
BoD	Bandwidth on Demand
BWCTL	Bandwidth Controller
cNIS	Common Network Information Service
DM	Domain Manager
gLS	global LS
GUI	Graphical User Interface
hLS	home LS
IDM	Inter-Domain Manager
IP	Internet Protocol
hLS	Home Lookup Service
gLS	Global Lookup Service
MA	Measurement Archive
MA(ad)	Measurement Archive for aggregated data
MA(at)	Measurement Archive for abstract topology
MA(t)	Measurement Archive for real topology
MDM	Multi-Domain Monitoring
MP	Measurement Point
perfSONAR	Performance Focused Service Oriented Network Monitoring Architecture
RC	Reservation Client
SIP	System Integration Point
SNMP	Simple Network Management Protocol
SSH	Secure Shell
TS	Topology Service
VC	Visualisation Client
WSDL	Web Services Definition Language