Towards Automated Interactions between the Internet
and the Carrier-Grade Management Ecosystems

Small or medium scale focused research project (STREP) Co-funded by the European Commission within the Seventh Framework Programme
Grant Agreement no. 258300

Strategic objective: The Network of the Future (ICT-2009.1.1)

Start date of project: September 1st, 2010 (36 months duration)

Deliverable D2.2
Architectural design of the management adapter

Due date: 30/04/2012

Submission date: 4/29/2012

Deliverable leader: ADVA

Author list: ADVA: Maciej Maciejewski, Christine Brunn
UPC: Anny Martínez, Xavi Masip-Bruin, Marcelo Yannuzzi, Wilson Ramirez
TUBS: Mohit Chamania, Admela Jukan
SNU: Jörn Altmann, Mohammad Hassan
TID: Óscar González de Dios, Fernando Muñoz del Nuevo

Dissemination Level

☒	**PU:**	Public
☐	**PP:**	Restricted to other programme participants (including the Commission Services)
☐	**RE:**	Restricted to a group specified by the consortium (including the Commission Services)
☐	**CO:**	Confidential, only for members of the consortium (including the Commission Services)

Table of Contents

Figure Summary

1 Executive Summary

Architectural design is a challenging endeavour in modern IT environments. Product developers need to be compliant with existing infrastructures and the software already deployed in the field while the designers need to foresee the potential demands for the product and anticipate the possible evolution of the environment, so that the architecture can keep pace with evolution.

Today's telecom industry is growing faster than ever, mainly driven by bandwidth requirements and the increasing number of services that need to be managed by telecom carriers. In this environment, one of the main aspects in the design of new products is the flexibility and scalability offered by its software.

The Service Oriented Architecture (SOA) model offers a promising framework to provide enhanced agility of businesses processes, which is especially important in the context of the ONE adapter. Compared to former architectural choices, SOA allows for a better alignment between the software architecture and the way business activities are organized within telecom carriers. Moreover, SOA provides an approach to develop applications using independent and reusable modules called services.

This document updates the description of the functional modules that compose the ONE adapter, which was previously reported in deliverable D2.1.1 [D2.2.1]. Each of the modules in the adapter provides a basis for the creation and exposure of SOA services, which are essential to provide the functionality required to coordinate complex operations through the ONE adapter. In the course of the project, we will develop the modules defined in this document and will show how interaction between them can be used to achieve programmable multi-layer orchestration for the use cases specified in Deliverable D 2.1 [D2.1].

2 Introduction

This deliverable aims at defining the architecture of the ONE adapter to facilitate multi-layer coordination operations as that have been reported in deliverable D2.1 [D2.1]. In the current carrier networks, IP and transport networks are managed by separate entities and significant manual interaction is required between the operators to facilitate any multi-layer operation. As a result, even the simplest multi-layer operation such as the provisioning of a new IP link can take at least hours and typically days. As a result, all networks (IP as well as transport) are typically over-provisioned and critical operations such as protection are generally duplicated in both networks. This practice generally leads to a very high CAPEX in the network.

Integrated multi-layer management has proven to be a difficult task, primarily due to the tangential management and operation philosophy of IP and transport networks. Transport networks were designed to be highly reliable networks providing a small number of services, and as a result the design of the management systems for these networks has been standardized across different vendors and technologies. On the other hand, IP networks were designed to be flexible in order to provide a large number of existing and upcoming services, and the demand for new features generally overrides the requirement for standardized operations across vendors. As a result, most IP networks are configured using vendor-specific command-line interfaces which also vary with different IOS versions of the same vendor. Other solutions for multi-layer management include the use of control planes or middleware boxes to facilitate specialized operations.

Control plane technologies such as GMPLS were designed to facilitate specific management functions in a distributed fashion in an effort to reduce the complexity of network management systems. These functions include topology discovery, provisioning, and restoration. Topology discovery with the control plane works well in multi-layer transport networks but currently does not include IP network discovery in its purview. In terms of provisioning, the User Network Interface (UNI) can be used by the IP network to provision circuits in the transport network. However, in the absence of automated topology discovery, the use of UNI still requires significant manual intervention to make sure that the end-points requested in the transport match the corresponding end-points in the IP network. Finally, the restoration of services in the control plane is still limited to the restoration in transport networks which again implies that additional resources must also be reserved in IP to guarantee against failures of the IP infrastructure.

The use of middleware boxes has been the most popular approach to facilitate multi-layer coordination, with these systems attempting to automate only specific aspects of multi-layer coordination by interacting with the NMSs in the IP as well as the transport networks. However, current middleware boxes are limited in two major ways:

1) Multi-vendor support: Given the diversity in management mechanisms (especially in IP) it is difficult for middleware boxes to support multiple vendors for multi-layer coordination, and extending support to another vendor generally requires an expensive development cycle.

2) Limited operation capability: Multi-layer operation in current middleware systems is typically hard-coded and as a result inclusion of new operations or modification of existing coordination operations is difficult and expensive.

Another challenge in the development of multi-layer management functions was the required level of change in the network management and operations in order to implement them. As today's interconnected world does not accept any disruption, network providers avoid any kind of temporary or long-lasting disruption to their customers. Therefore, there is a need for an easy-to-deploy and cost-effective solution to address a set of management issues that have proven to be bottlenecks in current telecom operations.

The ONE adapter is devised to be useful in future networks, which are expected to be composed of heterogeneous technologies. Our design attempts to facilitate multi-layer coordination while addressing the issues of multi-vendor support and user-defined multi-layer coordination. The proposed architecture for the ONE adapter addresses three major features in order to facilitate multi-layer orchestration in commercial network settings, namely 1) programmable orchestration, 2) semantic multi-vendor adaptation and 3) integration of third-party systems.

Programmable orchestration provides the operator to flexibly define operational workflows for multi-layer operations that mimic the existing business operations in the network. The ONE adapter architecture also takes into cognizance the fact that the process for defining a workflow within ONE should be simple in order for it to be usable by operators, who may not be specialized programmers.

Semantic multi-vendor adaptation helps in the operation of multi-vendor networks and in supporting migration of underlying hardware while maintaining the business logic programmed within ONE. Multi-vendor support is challenging in the different ecosystems, with multi-vendor support for IP being especially difficult. Configuration syntax and context can change significantly between vendors and even across the IOS versions of the same vendors which makes implementing multi-vendor support in IP especially challenging. In the ONE adapter architecture, we propose the use of Ontological transformations to map operations from a given internal operation specification to the corresponding configuration operations on the different devices connected in the network. Ontological transformations are used during configuration operations on IP and transport networks. Ontological transformations are also applied on incoming information to ensure that information used by the different components within ONE (e.g. multi-layer topology information) is not affected with changes in the external systems used by the operator.

Third-party systems such as the Path Computation Element (PCE), Authentication Authorization and Accounting (AAA), and Service Level Agreement (SLA) management can significantly improve the performance of the network, either in terms of optimizing actual network operations or in reducing the OPEX involved in the management of the network. However, the integration of third-party systems, especially in multi-vendor settings is challenging as there always exist some minor discrepancies in the specifications and/or interpretation of standards, which in turn leads to different implementations that may not always match each other. The ONE adapter architecture uses the Service Oriented

Architecture in order to ensure that third-party systems can be easily integrated within the ONE adapter as services and can interact with other management subsystems via the ONE adapter.

The design of the ONE adapter architecture is based on three governing principles, which are described in Section 3 of this deliverable. These principles determine the separation between core and auxiliary modules within the ONE adapter.

In Section 4, we describe the core, and auxiliary modules, and their means of communication. The ONE core consists of three modules, namely the Ontology Mapper (OM), the Workflow Processor (WP), and the Management Controller (MC).

- The Ontology Mapper (OM) assures that the processes and configurations required during a workflow are correctly interpreted (i.e., without ambiguities) among the various components within the ONE adapter. The OM enables the ONE adapter to interoperate and coordinate actions in a scenario composed of different technologies and different network management layers.

- The Workflow Processor (WP) is responsible for the execution of a specific workflow. All workflows are stored inside the ONE adapter and provide a specification of a set of multi-layer operations. Workflows within ONE can be static for re-usable operations or could be one-time workflows to execute a specific multi-layer operation.

- The Management Controller (MC) is mainly responsible for coordinating the operations and configurations through the ONE adapter, and is also responsible for the initial actions inside the ONE adapter before execution of a workflow.

The auxiliary modules offer a pragmatic approach towards an easy-to-deploy adapter. These modules are required to fulfil the already defined use cases, and to provide the expected functionality of the ONE adapter. These required modules include:

- A module for topology lookup

- A module for gathering measurement information

- A Trigger Module

- An IP-NMS control module

- A T-NMS control module

We also include descriptions for some optional and future modules, such as the SLA control, the Path Computation Client (PCC), and the charging and billing module which can enhance the functionality of the ONE adapter in terms of both technical and business operations. Aspects such as security, fault tolerance, and resilience are also considered during the architectural design of each of these modules, not only independently but also in the ONE adapter as a whole.

The preliminary architecture design of the ONE adapter is based on the Service Oriented Architecture (SOA), with the internal communication using the Enterprise Service Bus (ESB) architecture. In addition, by using standardized protocols and interfaces such as SNMP, NETCONF, and MTOSI, the ONE adapter assures its integration into heterogeneous carrier and the IP networks without introducing significant changes to these management domains.

In general, the fact that the ONE adapter’s auxiliary modules can be added or removed in accordance to the network providers’ needs, assures that the ONE adapter will be able to adapt and meet the requirements of the future Internet.

3 Architectural Concept

The architecture design of the ONE adapter takes into consideration three primary design goals, namely:

1) **Ease of Integration and Adoption (non-disruptiveness):** In order to facilitate large-scale deployment, it is essential for the ONE adapter to have the capability to integrate with different IP and transport technologies (and their management systems). At the same time, integration of the ONE adapter into the management ecosystem should not require (extensive) modifications in the existing subsystems.

2) **Replication of Business Processes:** The ONE adapter should have the capability to facilitate multi-layer coordination by replicating the current business and technical processes of a provider in order to ensure that the introduction of the ONE adapter leads to minimum disruption in the interactions between the different business units of the network provider.

3) **Support for Technology Migration:** The ONE adapter should remain relevant in the face of migration, and be relevant not only when changing technology in the IP or the transport network individually, but also in possible scenarios where providers may deploy hybrid devices supporting both IP and transport network infrastructure or in cases of multi-layer NMSs which can control both IP and transport networks from the same management system.

Driven by these design goals, we propose the architecture for the ONE adapter. As seen in Figure 1, the architecture is segregated into two primary sections, namely the *ONE Core* and the *ONE Auxiliary Modules*. The ONE Core is responsible for managing the ONE adapter itself as well as the orchestration of multi-layer processes, which basically consist of an ordered sequence of operations involving also the different auxiliary modules in the ONE adapter.

The Auxiliary modules in the ONE adapter are responsible for interacting with the external actors inside a carrier's management ecosystem. The separation of the process orchestration from the auxiliary modules means that integration of new modules does not affect the multi-layer process definitions themselves, thereby supporting easy integration and technology migration. The auxiliary modules are categorized based on the specific single/multi-layer functions (e.g. measurement or topology) and not by the external actor they communicate with. The categorization of auxiliary modules based on atomic network functions instead of external actors means that orchestration inside the ONE core defines processes based on a series of operations rather than a series of interactions between different external actors, and is therefore agnostic to the actual actors used in the management ecosystem.

It is clear that based on the diverse choices of external actors available in the management ecosystem, the auxiliary modules themselves may vary significantly when changing (or coordinating operations) between different technologies, and can therefore entail significantly different interfaces and data formats for the same function. To ensure that changes in the interfaces do not affect the process definition itself, we propose the use of an Ontology Mapper, which is responsible for simple data format translation as well as complex operation transformation for all communications between the ONE Core and the Auxiliary modules.

In this architecture, any operation of the ONE adapter is initiated by an *event*, which is received from an external actor in the *Trigger Module*. The Trigger Module uses one or more event notifications to compose an appropriate *trigger* for the ONE Core. The trigger can contain basic information necessary to initiate operations inside the ONE adapter based on an existing workflow, which is stored in the Workflow Database inside the ONE adapter or can contain instructions for orchestration of a series of internal workflows/actions with the necessary inputs. Once a trigger is received, the ONE Core Modules initiate the processing of the workflow for the operation and communicate with other Auxiliary modules, when necessary. After processing the workflow, the ONE adapter sends a notification of the status of the operation based on the configuration stored in the workflow or in the Management Controller.

Apart from features to support multi-layer interactions, the ONE adapter as a whole must also support a set of necessary features such as resilience, fault tolerance, and security to operate in a commercial management ecosystem. We shall address these features in Section 3.4.

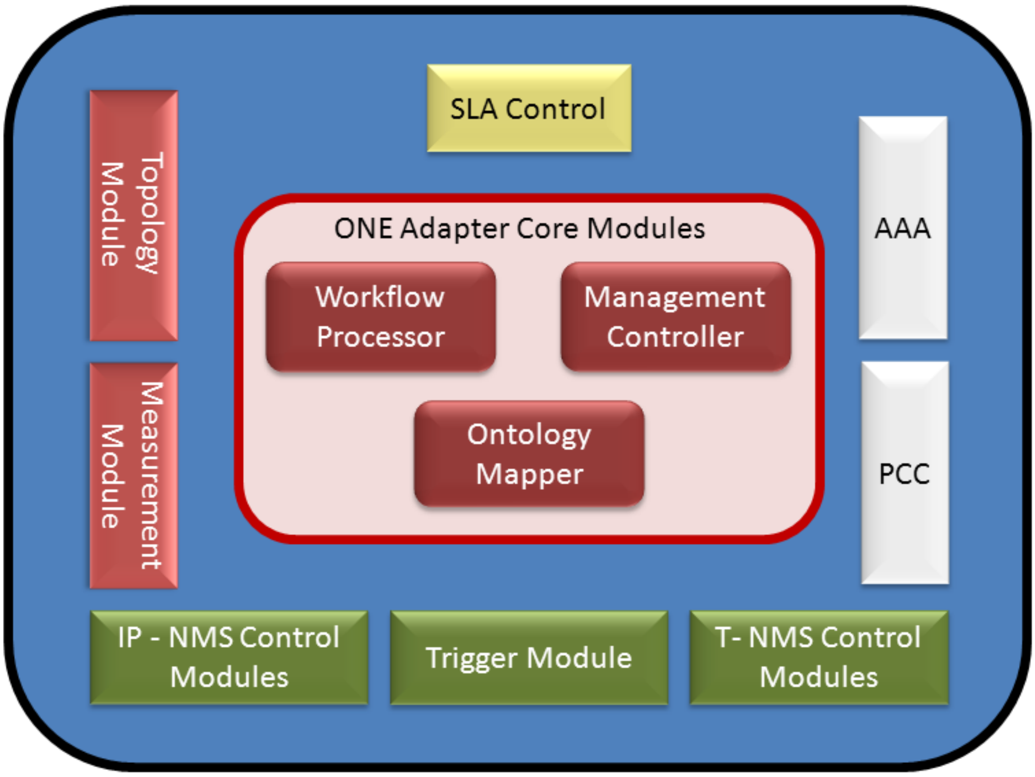


Figure 1 Architectural Overview of the ONE adapter.

3.1 ONE Core Modules

As stated above, the ONE Core is responsible for orchestrating a multi-layer process using the different Auxiliary modules available. In this architecture, the ONE Core must have the following features:

1) **Easy and Flexible Configuration**: Network Operators should have the option of defining a multi-layer process as an orchestration of interactions involving different Auxiliary modules. The architecture of the ONE Core must support flexible orchestration in order to provide a diverse set of processes, but at the same time must not be very complex to define.

2) **Ontology Support**: When creating orchestrations involving different auxiliary modules (while ensuring that network configurations have a high degree of reliability), the ONE Core must support Ontology based translation and transformations to compose and verify parameters exchanged with the auxiliary modules.

3) **Secure Operations**: The ONE Core is responsible for initializing any multi-layer process and must therefore support security functions to protect misuse against malicious attacks on the ONE adapter architecture. In the context of security, we plan to address the issues of verification of incoming triggers and authorization of actions for a given operation in more detail later in the upcoming deliverables D3.2 and D3.3.

4) **Policy Enforcement**: Policy enforcement in the ONE Core is required to solve contentions such as multiple simultaneous operation requests, and it is also needed to authorize multi-layer operations either via automated rules or manual intervention. These policy control mechanisms are also used in conjunction with policies programmed into external subsystems, which can trigger operations in the ONE adapter.

The major modules of the ONE Core are shown in Figure 1, namely, the Management Controller, the Ontology Mapper, and the Workflow Processor. As outlined above, the Workflow Processor is responsible for orchestration and execution of a series of functions defined through a workflow. The workflows are stored in a database within the Workflow Processor, and provide the specification for a set of multi-layer operations that can be used repeatedly. The only difference between two requests that are processed using the same workflow is in the input parameters. For instance, two requests for the provisioning of an IP link may differ in the interfaces used and/or in the router end-points, but the workflow used to orchestrate the provisioning of the link will be the same in both cases. The Ontology Mapper is responsible for the semantic interpretations and the transformation of operations and parameters as specified inside the workflow to specific actions based on the external actor (e.g., to configure IP routers from different vendors). The use of the Ontology Mapper ensures that, changes in an external actor or an auxiliary module, do not affect the process definitions inside the ONE Core. Finally, the Management Controller is responsible for configuration of the ONE adapter including configuration of the Core and the Auxiliary modules. The Management Controller is also responsible for routing and initial processing of triggers before execution of a workflow. For example, when a trigger is received and is classified, the trigger is sent to the Management Controller, which is responsible for facilitating authorization and workflow identification for the trigger. The Management Controller would also come into play to determine scheduling and pre-emption of workflows in case of concurrent requests for workflow executions.

3.2 ONE Auxiliary Modules

Our main design goal is to offer a pragmatic and easy-to-deploy adapter, enabling communications and coordinated operations between the IP and transport management layers, including automated provisioning of IP services over transport circuits and multi-layer self-healing operations. To this end, the adapter's core modules must be able to interact with external actors, such as human operators as well as with a set of systems that are typically deployed by telecom carriers, such as an IP-NMS, a T-NMS, a Monitoring and Measurement system, or a Multi-Layer Topology Database (MLTD). As we shall discuss in this document, our design targets a flexible and extensible adapter, which may also support interactions with newly emerged control and management sub-systems, such as a multi-layer Path Computation Element (PCE) together with its Traffic Engineering Database (TED), an Authentication, Authorization, and Accounting system (AAA), etc. In order to support interactions between these external actors and the adapter's core modules, a set of *auxiliary modules* are defined as shown in Figure 1.

The auxiliary modules can be divided in the following categories: 1) required modules; 2) optional modules; and 3) future modules. In the first category we include those modules that are mandatory to provide the functionality expected from the ONE adapter, and which are defined within the use cases in this project. More specifically, this group contains the following modules:

- ***Topology Module***: this module is in charge of obtaining the individual topologies in the IP and transport networks as well as the required correlation between a (node, interface) pair at the IP layer and a (node, interface) pair at the transport layer. This information is required for the coordination of tasks requiring multi-layer connection provisioning.

- ***Measurement Module***: The operations orchestrated through the core modules of the ONE adapter may require the collection of specific network measurements as part of the internal workflows provided by the adapter. Thus, this is the module in charge of handling the communication with an external measurement system. Through this interaction, the ONE adapter may request information about specific metrics and statistics in the network, such as the current load on a certain link, the status of a given Network Element (NE) interface, etc.

- ***Trigger Module***: this module is in charge of receiving notifications of external events that may initiate any coordinated operation through the ONE adapter. In our design, the coordination of operations is initiated by a trigger, which is generated from one or more event notifications coming from human operators, IP-NMSs or monitoring and measurement systems in the form of SNMP traps or from other external actors. The capability to correlate multiple event notifications in order to generate a trigger provides the operator flexibility in defining initiation mechanisms for operations in the ONE adapter.

- ***IP-NMS Control Module***: this module enables the communication between the ONE adapter and an IP-NMS to perform configuration functions on the IP network. In case that an external IP-NMS with configuration capabilities of NEs is not present in the network, the ONE adapter can be installed with an *IP-NMS Control Service* which provides support for automating the configuration via direct interfaces to the IP network elements. There might be more than one IP-NMS Control Module operating within the ONE adapter.

- ***T-NMS Control Module*****:** this module enables the communication between the ONE adapter and a T-NMS. Unlike the IP network, transport networks typically employ a network management system and provide standardized interfaces for the same. The ONE adapter will use these interfaces to communicate with the T-NMS in order to perform configurations (e.g. service provisioning) in the transport network. There might be more than one T-NMS Control Module operating within the ONE adapter.

- ***Path Computation Client (PCC)*****:** this module is responsible for the communication with an external multi-layer Path Computation Element (PCE) to facilitate path calculations required by the workflows.

The second category of auxiliary modules contains modules that are not necessarily required for the operation of the ONE adapter, but which may be desirable to include in order to bring our prototype implementation closer to the telecom carrier market. These modules include (and are not limited to):

- ***SLA Control*****:** this module is responsible for communication of Service Level Agreement (SLA) information to external systems of a network operator.

- ***Authentication, Authorization and Accounting (AAA) Module*****:** this module is in charge of authorizing the operations initiated through the adapter, and keeping records of the actions taken. Different profiles of "external actors" can be defined; in particular, the operators can be classified according to their administrative privileges, meaning that a given operator might be allowed or forbidden to initiate certain operations.

The third category encompasses modules that are foreseen as plausible candidates for inclusion in future releases of the ONE adapter. Among this group, we can include the interactions with business-related systems, such as a Billing System.

A high-level description of the internal communication between the auxiliary modules and the core modules of the ONE adapter can be found in Section 4.3.

3.3 Communication with external systems

External systems or actors are existing management subsystems in the carrier's ecosystem that will be used by the ONE adapter in some fashion. The basic model of the ONE adapter considers several actors including the IP-NMS, T-NMS, PCE, and the operator. Due to its modular architecture and the service-oriented approach, new actors can be easily integrated by extending the ONE adapter's auxiliary modules.

The interaction with the T-NMS should cover the following aspects:

- Configuration and provisioning requests through the MTOSI interface.

- Requesting information (topology, resource availability) over the MTOSI interface.

- Handling SNMP and MTOSI notifications.

The T-NMS will mediate between the Network Elements and the ONE adapter. For proper notification handling, we shall assume that the traps received by the network elements may be enriched with MTOSI naming, which would facilitate positioning them correctly within the network topology model.

In the IP network, there is no standardised or unified network management system. Thus, the ONE adapter should be prepared to perform the required actions, considering not only the mediation provided by the IP NMS tools described in the Deliverable D2.1, but also by directly accessing IP network elements if needed. Therefore communication may be supported over the following options:

- SNMP

- Command Line Interface (CLI).

- Network Configuration Protocol (NETCONF, RFC 4741).

The third actor considered is the human operator, who should be able to trigger workflows, and may be contacted for approval before specific operations are carried out via the ONE adapter. There are several possibilities for the interaction with a human operator, such as a Graphical User Interface (GUI), a command line interface, script inputs, etc. At this point, the exact interface which will be implemented (in the time frame) is not specified, but the architecture supports the integration of all the interfaces specified above.

In addition, other external systems can be integrated through ONE adapter to facilitate specific functions. Some of the possibilities are a Measurement OSS, AAA, SLA, Billing, etc.

The separation between the communication with external actors and the auxiliary modules inside the ONE adapter architecture makes it very flexible towards the continuous evolution in terms of communication with future external OSSs and Network Elements. The architecture of the ONE adapter is suitable for coordination both with multi-layer NMS systems as well as with external customer OSSs. Upcoming integrated NEs consisting of a hybrid IP router and optical switch[JUN11] can also be easily integrated within ONE, where it can interact directly with the NEs or can cooperate with the NMS that will drive these NEs.

3.4 Additional Features of ONE Adapter Architecture

In this section, we will present the ability of the ONE adapter to support necessary features for operation in a commercial management ecosystem. The primary features discussed here include Fault Tolerance, Resilience, and Security.

3.4.1 Fault Tolerance

In the context of the ONE adapter, fault tolerance can be seen as the ability of the ONE adapter to identify and deal with errors during operations. In the context of the architecture, the ONE adapter will provide basic features for fault tolerance, and will also allow users to embed capabilities of fault tolerance and verification inside the workflow definitions.

In the presented architecture, the ONE adapter will attempt to primarily use existing services provided by external actors for implementing auxiliary modules, thereby relying on the fault tolerance capabilities of existing management subsystems to ensure smooth operations of these modules.

Basic fault tolerance mechanisms can also be integrated into the Ontology definitions, and the ONE adapter will use these definitions to verify constraints on interactions with the ONE auxiliary modules. In case that an auxiliary module is developed specifically for the ONE adapter, the design of the module should incorporate mechanisms for fault tolerance.

Finally, the ONE adapter will also inherently support features for rollback during the execution of a workflow in order to ensure that the network returns to its original configuration in case of failure during the network configuration. Network operators can also incorporate fault tolerance against operations in their workflow definitions, by requiring verification of certain actions.

3.4.2 Resilience

Resilience in the ONE adapter architecture will deal with the ability of the adapter to operate in case that some of the core/auxiliary modules fail. The ONE adapter architecture is designed so that the modules can be distributed across multiple systems and can interact with each other. The design can be used to reduce the probability of all modules (auxiliary as well as core) failing simultaneously. The ONE architecture will further address scenarios where one or more of either the Auxiliary or the core services fail.

Resilience mechanisms for failures in auxiliary as well as core modules may include but are not limited to duplication of individual modules at different sites. In the current architecture, the ONE adapter's *core modules* are designed to be state-less in-between workflows so as to avoid issues of state synchronization in case of failures. While the design will also attempt to develop state-less auxiliary modules when possible, that may not be the case at all times and specific solutions (e.g. maintaining state in a separate database) would be employed for modules which require that state be maintained.

In case of failure of one or more modules, another challenge faced is the affect on the inter-module communication within the ONE adapter. As the system is designed on the service-oriented architecture, we can employ an Enterprise Service Bus (ESB) to overcome this problem. The ESB would act as a communication mediator between any two modules in the ONE adapter architecture, and in case a module has failed, the ESB would transparently re-route the communication to the duplicate module (if available).

3.4.3 Security

Security features in the ONE adapter architecture will primarily address issues of authentication of incoming triggers to initiate operations in the ONE adapter as well as implementation of authorization functions for specific operations. We assume that the interaction between the ONE Auxiliary modules and the actors that are part of the existing management ecosystem (such as IP/T-NMS) are trusted.

The ONE adapter will integrate with the carriers' AAA infrastructure, and will require that all incoming triggers contain information to authenticate themselves against the provider's AAA infrastructure. The primary authentication will be used to determine the identity of the event/user responsible for initiating operations inside the ONE adapter.

Secondly, configurations via the Auxiliary modules may require authorization information. The request for authorization can come from 1) the external actors such as the T-NMS before performing any action or 2) from the Auxiliary module itself and can be controlled by the network operator. In order to facilitate these authorization requests, the ONE adapter may use the provider's AAA to provide authorization information when communicating with these Auxiliary modules to ensure that the initiator of the event has the appropriate authorization to carry out the necessary network configurations.

Note that, in the scope of this project, the ONE adapter will not address security concerns arising from the inclusion of a potentially malicious workflow into the ONE adapter that can significantly affect the regular operation of the network. Note that this does not introduce additional threats to the management ecosystem, since an operator authorized to develop and execute a malicious workflow can actually produce a similar level of harm today, though in a manual fashion.

3.4.4 Scalability

Scalability of the ONE adapter can be viewed from two different aspects, namely network size and operation frequency. In terms of network size, the scalability of the ONE adapter is limited primarily by the scalability of the management subsystems it uses to perform multi-layer operations. The ONE adapter will only communicate with network elements directly when necessary and hence can scale to the network sizes currently supported by existing NMSs.

The scalability in terms of operation frequency has to be addressed within the ONE adapter architecture. If all modules in the ONE adapter are completely stateless, then, in theory, the operations in the ONE adapter can be parallelized, and scalability can therefore be increased simply by increasing the computing resources available. However, specific operations in the ONE adapter, especially IP configurations, cannot be made completely stateless: for example, if two parallel workflows attempt to configure the IP address on the same interface at the same time, the result of the operation may be unpredictable. The scalability of the ONE adapter is therefore dependent on the implementation used for individual auxiliary modules and external actors. In case that the resolution of competing operations is not provided by the external actors or the auxiliary modules, the ONE adapter would be restricted to execute a single workflow at a time and the scalability would be restricted by the execution time for a workflow in the ONE adapter ecosystem.

It should be stated here that multi-layer operations are very rare in the network today. As a consequence, the ONE adapter architecture is not optimized for frequency of orchestrations supported, and instead focuses primarily on dealing with policy enforcement, contention and pre-emption of workflow executions rather than concurrency and parallelization in execution.

4 Modules Details

4.1 Core Modules

As stated previously, the ONE Core consists of three major functional modules: the Workflow Processor, the Ontology Mapper and the Management Controller. These three modules interact with each other to facilitate policy enforcement and process orchestration in a technology agnostic way.

4.1.1 Workflow Processor

The Workflow Processor module is used to execute a process orchestration inside the ONE core. The Workflow Processor module contains a workflow database, which stores the workflows or orchestration definitions of the different processes defined by network operators. In the database, workflows are mapped to specific triggers and upon arrival of a trigger to the ONE Core, the Management Controller requests a specific workflow definition based on the received trigger format. A central database also allows large workflows to request smaller workflows stored in the database, thereby reducing the complexity of the workflow definition. The Workflow Processor receives a request to execute a specific workflow in the workflow database from the Management Controller uses the workflow definition in the database to orchestrate the execution of operations, e.g., by gathering system information and performing configurations through the auxiliary modules. In a complex workflow, the input parameters for an operation request to an auxiliary module are generated from a combination of one-or-more outputs from requests to other modules. The Workflow Processor works in conjunction with the Ontology Mapper to facilitate the information exchange from the Auxiliary modules. More precisely, the Workflow Processor using the workflow definition to determine the information exchange based on the ONE adapter data model definition, while the Ontology Mapper helps in the semantic translation of information coming from the external actors via the auxiliary modules into the ONE adapter information model.

4.1.1.1 *Roll back*

The Workflow Processor is also responsible for rollback operations in case of an error during the orchestration. Smooth and accurate execution of rollback operations is critical as unfinished or erroneous configuration of network elements can be catastrophic for smooth network operations.

Roll back is a specific function of the Workflow Processor. It allows a system to return into a state that is well defined if a workflow cannot be completed successfully. The rollback function monitors the completion of each step of the workflow. A workflow rollback is initiated if a certain event occurs. For instance, such event could be the delay in the completion of a step within a workflow or a failure of a configuration operation during the workflow execution.

One way to implement such a mechanism is to use "if else" conditions in the workflow definitions. In this case, all possible events of a workflow step need to be studied and the corresponding actions need to be defined ahead of the workflow execution. Another possible mechanism is to design policies for each possible event during a workflow execution and use them to decide the next operation in the workflow. For instance, if a configuration action fails, the policy could direct the workflow to retry the operation or initiate a rollback operation.

Depending on the design of the Workflow Processor, the roll back function can be part of the workflow itself or it can be a separate decision making system that oversees the workflow execution.

4.1.2 Ontology Mapper

The fundamental role of the ONE adapter is to enable the interoperability between two network management layers that are currently isolated. The heterogeneity of the NMSs used by telecom carriers at the IP and transport layers poses complex challenges in the design of the adapter, requiring solutions for data model adaptations, as well as the need to communicate unambiguously with the different actors involved, in order to perform coordinated management operations between the IP and the transport layer. To achieve these goals, a formal way of representing concepts is needed, which can endow the adapter with the capability of solving the semantic interoperability problem when management operations involve the configuration of devices both at the IP and transport layers. In our design, the formal representation of concepts is based on a set of *ontologies*. In the path of our research we are exploring two possible approaches for solving the semantic interpretations and interoperability issues for multi-vendor IP configurations. On one hand, we consider solving these issues by means of *mappings* between ontologies [LVA03] [WRP05] [TLL06] [DMD02], while on the other hand, a different approach is based on the automatic instantiation of vendor-specific ontologies[BUI05] [BON03] [CEL04] [MAY08], which aim to represent the specifics for configuring a given device. Despite the fact that both approaches aim at the same goal, which is, achieving semantic interpretation and syntactic adaptation of high-level and agnostic requests, they differ on the way in which this is accomplished. For the former approach, we assume to have individual representations of the concepts of each vendor domain, placing then, the complexity on the algorithmic component of the mappings between concepts of different ontologies. On the other hand, the later approach aims at building an automatic instance by extracting knowledge, terms and information from external meta-data (e.g. from the routers' HELP command set) allowing the generation of a vendor-specific instance of the generic ontology. Both approaches pose major challenges as well as different limitations for achieving a truly ontology-based driven solution. For the mapping approach beforehand we can assert that a one-to-one mapping is not guaranteed, fact that increases the level of complexity of the algorithmic solution. For the automatic instantiation approach, which is based on the belief that there is a common shared knowledge of the subject domain, it highly depends on the quantity and quality of the meta-information that is handled to build from scratch and populate in an automated form the specific ontology.

Therefore, the “Ontology Mapper” is one of the key building blocks in the architecture of the ONE adapter. This block will provide the necessary means to enable the automatic mappings between ontologies or instance generation, and it shall be performed in such a way that the semantics between specific concepts embedded in different ontologies can be aligned accordingly. In a nutshell, the Ontology Mapper will supply the algorithmic engine for finding the correspondence between concepts belonging to different vendor domains.

In our design, the main application of the Ontology Mapper module is for processing and adapting the required configurations to the specific command set of the equipment present in the network, from one of either of the two approaches previously exposed. The mapping approach is illustrated in Figure 2. In this case, the module provides both, the semantic interpretation of the configurations required and the mappings to the corresponding command set of the devices involved in the operation that has been requested—the example shows the mappings needed for automated configuration of different router models. The second approach referred to the automatic instantiation of vendor-specific ontologies is illustrated in Figure 3.

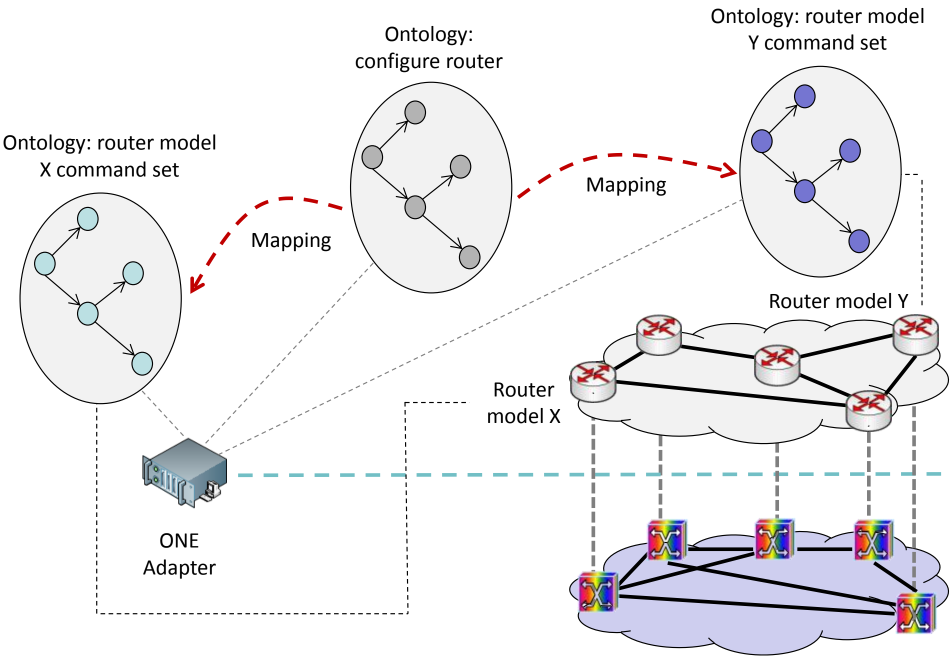

Ontology: router model Y command set
Ontology: configure router
Ontology: router model X command set
Mapping
Mapping
Router model Y
Router model X
ONE Adapter

Figure 2 Semantic interpretation of the configurations required and mapping them to the specific command set of the devices involved.

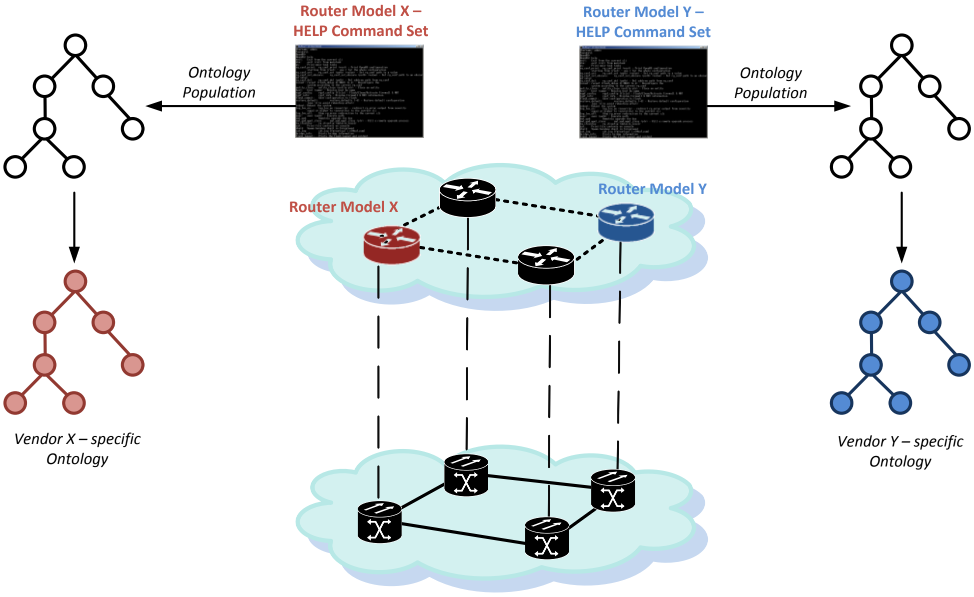


Figure 3 Semantic interpretation of the configurations required and automatic instantiation of vendor-specific ontologies.

4.1.3 Management Controller

The Management Controller (MC) manages the overall actions carried out by the ONE adapter, including the configuration of the core and the auxiliary services, the workflow execution authorization, the accounting of workflow executions, workflow policies and prioritization, as well as the communication with the ONE administrator and the network operators. The Management Controller gets request from the Trigger Module.

In order for the Management Controller to accomplish the above-mentioned tasks, the architecture of the Management Controller consists of three sub-modules, namely the Workflow Triggering (WFT), the Logging and Monitoring (LM), and the Analytics module.

To achieve these tasks, the Management Controller interacts with the Trigger Module, the AAA module, the Workflow execution module, and has a user interface (GUI) to the network operators (Figure 4). The user interface allows giving feedback on the workflow triggering process and the workflow prioritization.

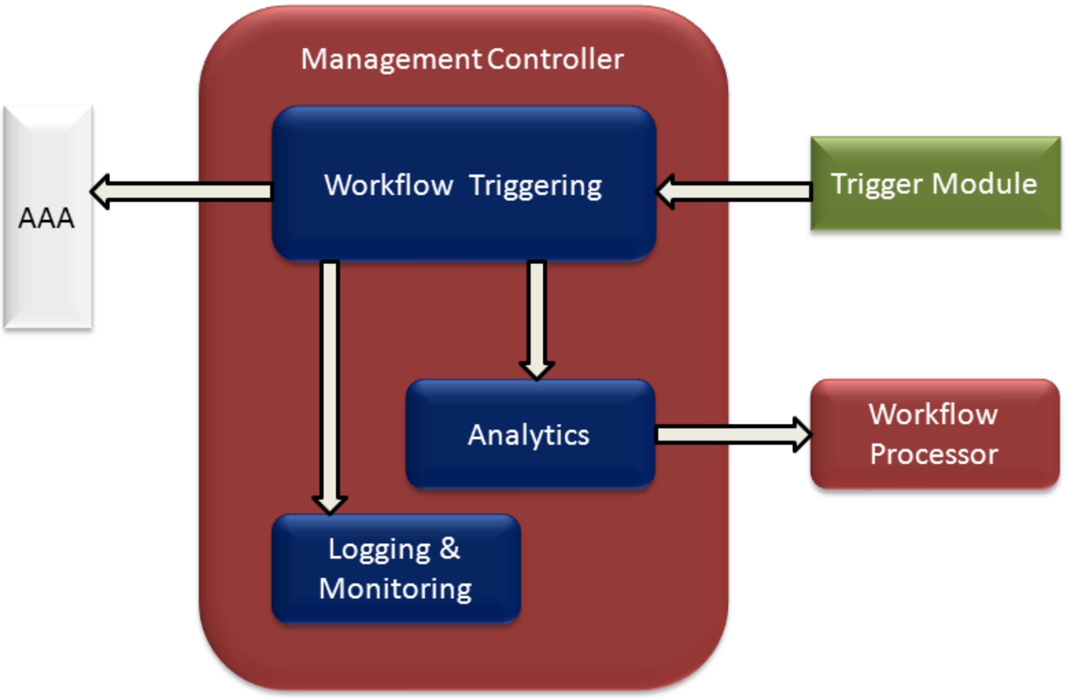


Figure 4 Architecture of the Management Controller, workflow triggering module, and the analytics

4.1.3.1 *Workflow triggering*

The different components of the Management controller are shown in Figure 4. The primary function of the Workflow Triggering sub-module is to receive Triggers from the Trigger module, assign a workflow to the trigger and check whether the workflow initiator (i.e., person or device) is authorized to request the execution of a specific workflow. This function is accomplished with the help of the AAA module. The primary objective here is to check if the initiator is authorized to execute a workflow or not, which must be specified in advance.

As shown in Figure 4, the Trigger Module (TM) sends the trigger information to the MC. The workflow triggering sub-module assigns a workflow using the workflow database in the Workflow Processor, and uses the AAA module to check if the initiator is authorized to request the execution of the workflow. If the AAA approves the execution, then the WFT passes the received information to the Analytics sub-module.

4.1.3.2 *Logging and Monitoring*

The logging and monitoring sub-module consists of a database that stores all logging information sent by any of the modules (both the core and the auxiliary modules). The stored data can then be accessed by the ONE administrator to analyse statistics or irregular behaviour of the ONE adapter. It is possible that the ONE administrator is supported in this activity by a log-analyser, which checks for certain events in the logs.

4.1.3.3 *Analytics*

The analytics sub-module prioritizes the execution of workflows and may initiate pre-emptive scheduling of workflows. Pre-emption is defined using a static priority associated with each workflow and policy which dictates if a workflow should/can pre-empt the execution of an already running workflow.

4.2 Auxiliary Modules

As described earlier, the auxiliary modules enable the interactions between external actors (e.g., IP-NMS, T-NMS) and the core modules in the adapter's architecture. These modules consist of a set of interfaces and protocols that provide the necessary support to allow the communication between the adapter and the different actors involved during any operation. In the design phase, we are trying to rely as much as possible on standardized and well-accepted protocols and interfaces for the implementation of the auxiliary modules. This will ensure that the internal representation is flexible to facilitate integration of new auxiliary modules in the future, which would not only allow the adapter to evolve in time but would also expand the horizon of possible use cases and applications. Note that by using auxiliary modules, we can clearly ease the process of integrating new management technologies and add-on features to the adapter.

In the high-level design proposed in this document, every auxiliary module is actually devised as a functional block that provides an atomic set of tasks. The modules conceived at this stage are described further in this section.

Required indicates that the module is mandatory in order to provide the functionality expected from the adapter.

Optional covers a set of modules that are not necessarily required for the operation of the adapter, but which may be desirable to include. In particular, in a more advanced phase of our design we plan to investigate the interactions with a PCE, which will be handled by the Programmable Logic Module.

Future modules, on the other hand, represent those that we see as prospect incorporations but which will not be explored during this project.

We now proceed to describe in more detail the needs, the design objectives, and the features of each of these auxiliary modules, starting first with those that are required, then going through some optional modules, and finally outlining the ones that we foresee as candidates for future implementations within the ONE adapter.

4.2.1 Topology module

Topological information functions are required to leverage the existing mechanisms used by the network operators to discover and identify network elements both through the IP and the transport NMSs. The *Topology Module* will gather the necessary information matching a (node, interface) pair at the IP layer with a (node, interface) pair at the transport layer and store the correlation between an IP link and the corresponding circuit in the transport layer. This information is the basis for constructing and maintaining updated the multi-layer topology of the carrier's network. To this end, topological modules will also interact with other external actors such as a multi-layer PCE or the inventory databases used by telecom carriers.

In our preliminary design, the Topology Module in the ONE adapter will be in charge of obtaining the necessary topological information from an external source (e.g., by querying a multi-layer PCE or an inventory database). In this initial phase of the design, we will start from a basic setting where the multi-layer topology (and hence the correlation of (node, interface) pairs) is obtained from an external source. In a more advanced scenario, we plan to explore other alternatives, and investigate if we can come up with a practical solution, through which the adapter can automatically "discover" and keep updated these correlations without the need of an external repository. If successful, our research can bring new possibilities and add value to the adapter, since the latter can actually become a "provider" of the multi-layer topology. In this case, the adapter could be used for keeping inventory databases updated, performing crosschecks, or may become one of the sources that might feed the Traffic Engineering Database (TED) of a multi-layer PCE. While we plan to explore mechanisms to automatically discover multi-layer topology, for the moment the multi-layer topology is assumed to be obtained from an external actor.

4.2.2 Measurement Module

During its operation, the ONE adapter must be able to get information about the state of specific network resources, since this is essential not only for coordinating the provisioning of IP services over a transport network, but also for endowing the network with self-healing capabilities. As an example, the adapter must be able to check whether a given interface of a given router is used before a provisioning operation, or if a certain link has sufficient remaining capacity during a self-healing process. The *Measurement Module* is the one that enables this functionality, by allowing the ONE adapter to communicate with external monitoring and measurement subsystems.

It is important to note that the adapter per se will not perform any kind of network measurement, but it will use this module to get the necessary state information from an external system. Moreover, most of the interactions with a monitoring and measurement system that we conceive at this stage of our design are mainly for consistency checks during the execution of workflows, as well as to gather information about the utilization and traffic performance on certain interfaces in the network.

The ONE adapter may also gather information from monitoring systems to proactively prepare the ground for a self-healing action. For example, alarms such as a neighbour loss in the IP layer and link/interface failures in the transport layer can be correlated by the ONE adapter, which could suggest or even make a restoration decision based on the state and measurements proactively obtained from an external monitoring system.

The Measurement Module will use internal definitions of types of measured data. Those data types along with the measurement period information, and the location of the data to be measured, will serve for all ONE adapter internal measurement data communication. Correlation of those to the actual network obtained data is the responsibility of the appropriate IP-NMS module, T-NMS module or External Measurement Systems module.

4.2.3 Trigger Module

It is worth emphasizing that any coordinated operation initiated through the ONE adapter will be triggered by an external event. The *Trigger Module* is the module that receives the external event notifications, which may be issued by a human operator, by management sub-systems such as an IP-NMS or a Monitoring and Measurement system, or by any other external actor.

A *trigger* may be generated by one or more event notifications, and the Trigger Module contains an event correlation function. Here, event correlation rules generated automatically or defined by the operator can be defined, and the event correlation function will then wait for a specific occurrence of multiple events in order to initiate a trigger. Such functions are typically necessary for complex operations such as multi-layer restoration, where the ONE adapter must ensure that there are multiple failures that cannot be recovered by the management systems alone and require a multi-layer restoration operation.

Note that the event notifications coming to the Trigger Module can be very different. For example, notifications can come in the form of web service messages from GUIs, SNMP traps/messages either from transport or IP systems or other protocols such as the PCEP protocol for notifications from the PCE. The Trigger Module is also responsible for composing this information in a standard format, which can then be used within the Management Controller to initiate a workflow.

4.2.4 IP-NMS Control modules

Management functionality in IP networks can be split into two functionalities: monitoring and configuration. The *Monitoring* functionality is in charge of receiving alarms and other information related to the IP routers while the *Configuration* functionality facilitates configuration and service provisioning operations in the IP network.

Today, the standards related to the configuration process of IP network elements are not sufficiently mature. Current approaches are attempting to use standardized XML-based approaches and definitions for operations through protocols such as NETCONF (RFC 4741), which have had a strong impact on the network management industry, even inspiring commercial implementations like JUNOScript [JUR10]. However, while there is not a dominant standardized technology for management, existing mechanisms of SNMP and Proprietary CLI will continue to be used in the future years. On similar lines, IP NMSs also use proprietary interfaces (unlike transport NMSs) and can vary significantly between different NMSs.

The IP-NMS control module must therefore have the capability to perform configuration operations over a variety of interfaces. As shown in Figure 5, the ONE adapter must have the capability to interact with the IP NMS over a variety of protocols in order to facilitate the configuration.

The configuration process becomes significantly more challenging in the absence of an IP NMS. Given the diversity between interfaces to configure IP network elements, the control module shall rely on the Ontology Mapper to facilitate transformation of complex configuration operations in a generalized representation (used in the workflow) into a sequence of controlled configuration steps based on the equipment vendor or the interface used to configure IP network elements.

Primary objectives of IP-NMP control module are:

- Interact with IPNMS ecosystem despite the management's protocols used by the IPNMS (**Error! Reference source not found.**).
- Provide management information from, and to the IPNMS ecosystem to leverage its functionalities.
- Offer automation of configuration of NEs at the IP layer in case of the absence of an IPNMS that offers this service.
- Offer support at the time of the provisioning of a new service.

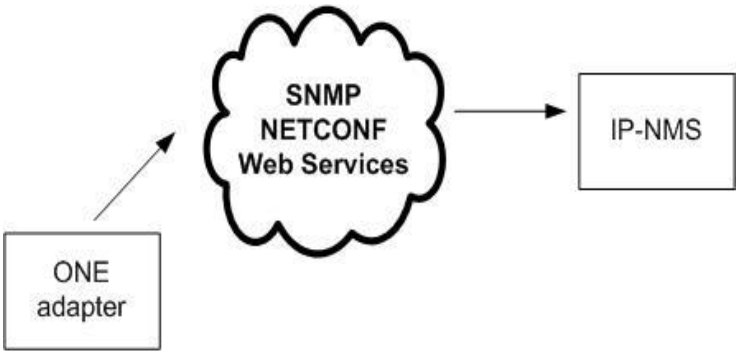


Figure 5 Example of the interactions involvingthe ONE adapterand anIP-NMS (Note: the communication is bidirectional)

4.2.5 T-NMS Control modules

The primary objective of the T-NMS control module is to facilitate the communication with the network management system of the transport network. It should be capable of cooperating with other ONE adapter modules, namely it has to be able to:

- Handle the requests from the *Topology Module* to acquire the transport network topology
- Cooperate with the *Ontology Mapper* to support various network models
- Execute requests coming from the *Workflow Processor* regarding path provisioning
- Provide measurement data requested by the *Measurement Module*

- Cooperate with the *Management Controller* in terms of logging

Interfacing to T-NMS systems depends on their capabilities of exposing the equipment functionality. Current operator trends often turn to the MTOSI interface, as the one with well-defined functionality and data models for all levels of interest, ranging from inventory to service support. The modularity of the ONE adapter lets it adopt easily control over older T-NMS applications with a different set of management interfaces, like CORBA TMF814 or SNMP.

4.2.6 Path Computation Client (PCC)

The Path Computation Client is an instance of a Programmable Logic Module, which may be used for computing multi-layer paths in the network. The Path Computation Client Module will interact with an external Path Computation Element (PCE), which is a centralized server used to compute paths.

A PCE is a node that has specialized path computation capabilities and receives path computation requests from entities or clients called Path Computation Clients (PCCs). The use of a PCE eliminates the need for path computation capabilities in every node within the network. For instance, there is no need for every node in the network to maintain a path computation database; there is now a central database, which can be used for path computation purposes. In order integrate the PCE with the ONE adapter, the communication between the ONE adapter and the PCE can be accomplished using the PCEP protocol [OKI10]. The communication process between the ONE adapter and PCE is shown in **Error! Reference source not found.**. One of the auxiliary modules of ONE adapter will act as a PCC enabling access to the features and strengths offered by a PCE.

When requesting a path computation to a PCE, in the request will be embedded the following information:

- Origin and destination points.
- Bandwidth requirements.
- Cost limits.
- SLAs.
- QoS parameters.
- Layer Levels (in which the path computation will be calculated).

In this architecture, the Path Computation Client part of the ONE adapter will act as a gateway for path computation requests directed to the PCE, and will convert the response from the PCE into the standardized format used inside the ONE architecture.

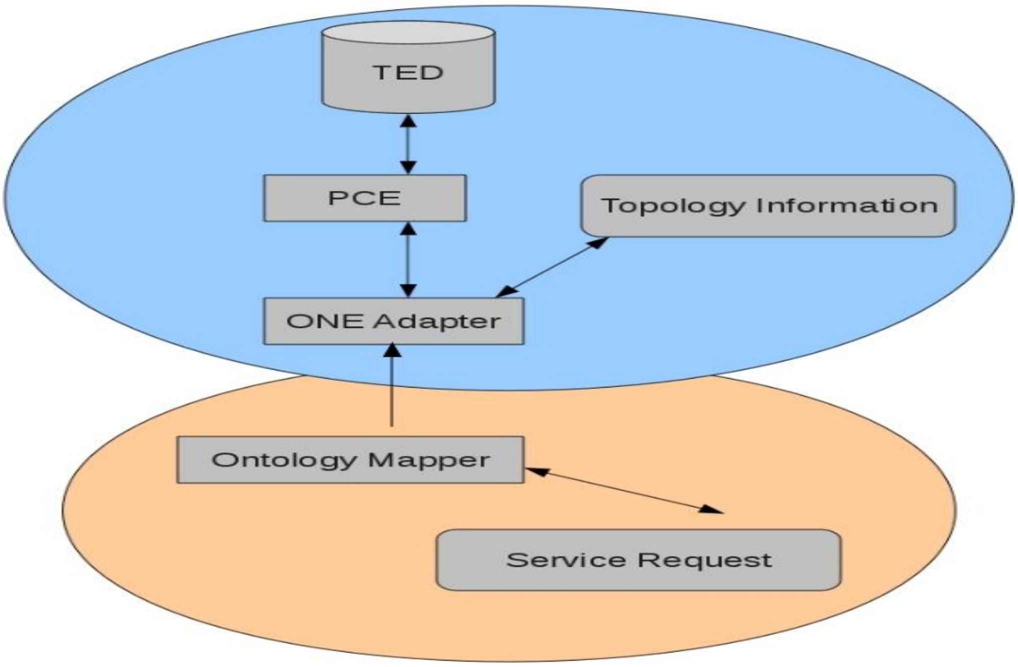


Figure 6 Inter-Communication between the ONE adapter and the PCE

4.2.7 SLA Control

A Service Level Agreement is a key feature in any network that offers quality of service to users. The communication with this type of subsystems represents an aggregated value to the ONE adapter architecture.

Parameters like jitter, throughput, and percentage of available time, mean time between failures, etc, could be taken from external business systems or SLA parameters embedded in triggers. Depending on these parameters, the provisioning of a service can be accomplished or not. Going further, a warning could be sent to the user when the SLA is not being accomplished due to variations of the network state.

To compete successfully, companies must proactively manage the quality of their services. Since provisioning of those services is dependent on multiple partners, management of partner services SLAs become critical for success. SLAs are used to define and manage expectations among partners for performance, customer care, billing, service provisioning, and other critical business areas. SLA Management can also be used to assess predefined penalties when SLA parameters, such as failure to meet performance, time-line, or cost requirements, are not met. For example, if network downtime exceeds one hour, the penalty is a 10% rebate of service fees.

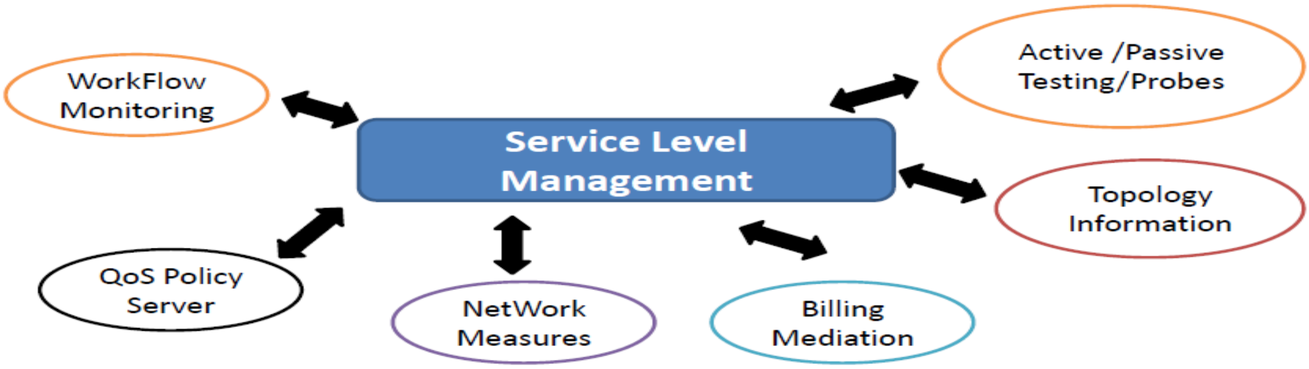


Figure 7 SLM Architecture

Figure 7 shows that a number of different data sources are required to support effective service level management.

Objectives

- Extrapolate SLA agreements defined in outside subsystems to the network.
- Construct an interface to dispatch service level specifications.
- Support static and Dynamic SLAs definitions.
- Define an XML template of SLA specifications that can be received by the ONE adapter.

4.2.8 Authentication Authorization and Accounting (AAA)

The AAA module will support authentication, authorization and accounting capabilities. The authentication and authorization functions may be used for granting or denying permission to execute a workflow. The access profiles stored in the AAA module define levels of access to certain functions and operations of the ONE adapter.

The ONE adapter may also facilitate exchange of accounting information to facilitate accounting of multi-layer operations performed by the ONE adapter.

The AAA module architecture, which is shown below in Figure 8, consists of three sub-modules, namely the AAA engine, the Policy DB, and a sub-module to deal with the applications having specific characteristics. As stated before, the Trigger Module and the Management Controller communicate with the AAA module to authenticate and to check the trigger initiator’s rights for executing a specific workflow. In addition to this, the analytics sub-module will report usage information to the AAA module. The usage information is related to the SLA agreement between the IP network and the transport network. For this communication, we will use protocols like DIAMETER and RADIUS.

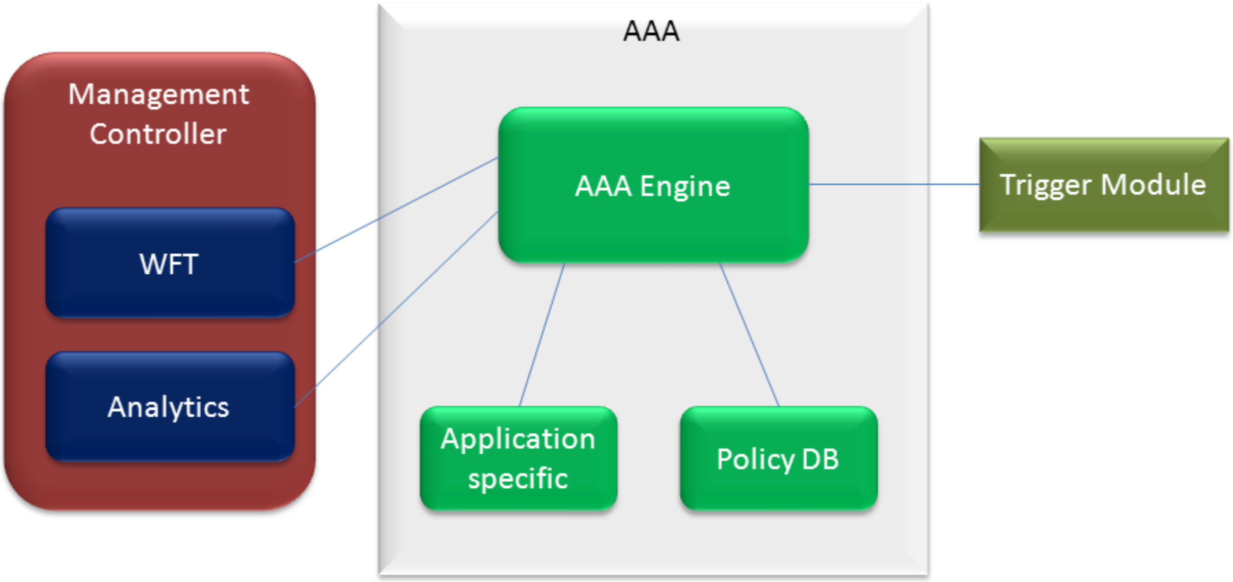


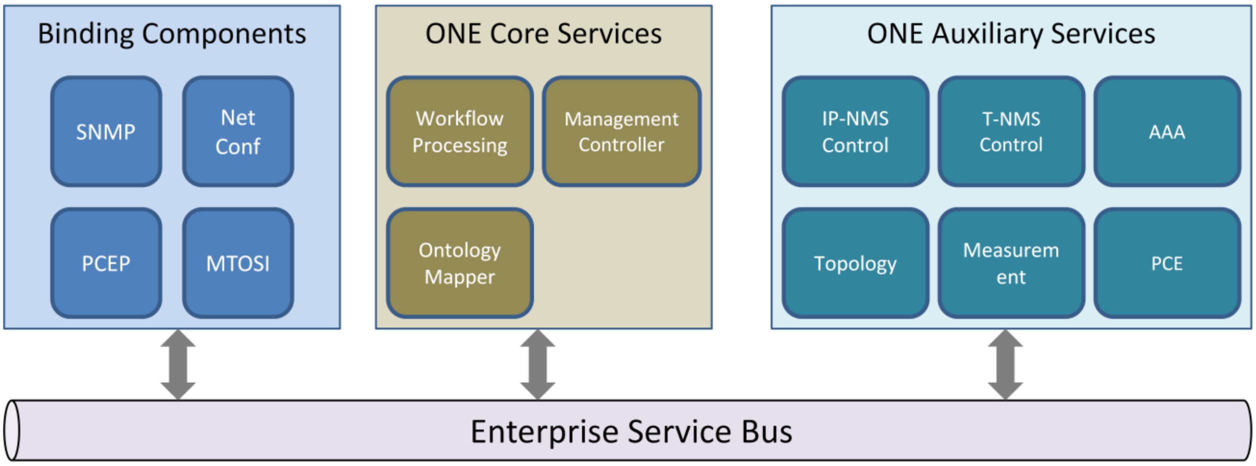
Figure 8 AAA module architecture

4.3 Internal Communication

From the design shown in Figure 1, and the subsequent description in Sections 3 and 4, it is clear that the ONE adapter architecture can facilitate multi-layer coordination with a degree of independence from the actual external subsystems employed in the operator's management ecosystem which is largely down to the SOA [ERL07] [HEW09] approach where the different modules are implemented as services. However, we must also consider the mechanism for communication between these modules as that can largely affect factors such as the capability for migration, resilience, load-balancing etc. in the ONE adapter.

In the implementation, we will employ the Enterprise Service Bus Architecture, which acts as a communication channel between different services in the network. Note that during the implementation, the modules in the ONE adapter will be referred to as services.

A Web-service is associated with one or more Service Endpoints, which define the location of the service and the protocol (SOAP/REST). In order to interact with a service, the *consumer* of the service must be aware of the actual end-point. In case of the ONE adapter, for example, if end-points for auxiliary services were directly encoded into the workflows, the migration of a service from one end-point to another would require alteration of every workflow in the Workflow Database. Similarly, if an auxiliary service was not available and a backup service was to be used, the information about the backup service would also have to be programmed into the workflow which would significantly increase the complexity in defining the workflow.



Binding components provide functionality to communicate with external systems using standard protocols

Figure 9 The Enterprise Service Bus for Internal communication in the ONE adapter

The Enterprise Service Bus, as shown in Figure 9, acts as a mediator between the consumer of a service and the actual service itself. Here, a service is published via the ESB, and the consumer of the service uses a service endpoint advertised by the ESB to consume this service. This mediation mechanism can help address a large number of implementation as well as design issues, such as:

1) **Service Endpoint Migration:** In case a service must be re-located to a different location, the operator only needs to configure the new end-point in the ESB, while configuration of all service consumers remains unaffected.

2) **Load-balancing and Resilience:** The current generation of ESB platforms offer in-built mechanisms for load-balancing of service requests. For example, a service may be implemented on multiple physical machines (with different end-points) and service requests coming to the ESB can be balanced between these end-points. Similar mechanisms can also be used to facilitate resilience in case one of multiple endpoints of a service becomes inactive.

3) **Logging, Monitoring and Alarm Generation:** As all messages are directly routed over an ESB, the mediation service can also log inputs/outputs to the service as well as monitor the requests/responses for faults. Most ESB implementations support user-defined logging and monitoring capabilities and can also be used to generate alarms to the operator in case of a fault.

4) **Data Transformation and Message normalization:** Data formats can vary with the web service implementations used, which in turn adds complexity to workflow processing if not addressed elsewhere. A basic example can be a scenario auxiliary modules can use two different formats (SOAP and REST) for their implementations. As the web-service calls are different in different formats, consumers cannot make a SOAP call to a RESTful web service and vice versa. However, when routed over an ESB, the consumer can make a SOAP call to

the endpoint published by the ESB. The ESB then translates the incoming call to a RESTful call to the actual service, performs transformation on the response received from the actual service end-point to make it SOAP compliant before responding to the service consumer. This particular example is commonly referred to as *"Message Normalization"* in SOA. The current generation of ESBs also support a large number of binding components which help facilitate normalization across other protocols that may not be a Web service. For example, using an SNMP binding connector, a web service may make a web-service call to the ESB, which in turn can communicate with a router over SNMP using the SNMP binding connector. The availability of binding connectors can significantly reduce the implementation overhead when developing integration with existing subsystems.

5) **Access control:** Finally, the ESB can provide a level of access control to the services in the ONE adapter, where access to the auxiliary services can be restricted to the core services only, and vice versa, while the user only has access to specific services exposed by the Trigger module to initiate a workflow and interfaces exposed by the Management Controller for administration of the ONE adapter.

The choice of using ESB for internal communication within the ONE adapter is primarily motivated by the abovementioned functionalities. We also note that the ESB architecture has been proven to be highly scalable, with large e-commerce entities employing the architecture for managing inter-communication in their back-ends [WS01].

5 Conclusions

The architectural design of the ONE adapter, which is reported in this deliverable, has been derived basically from two needs that are present in today's telecom market. Firstly, network providers are requiring flexible tools enabling coordinated operations and orchestrations involving the IP and transport networks. Secondly, it must be possible for network providers to integrate and exploit such tools without disruptions on their production environment. It is not acceptable for network service providers if the integration of a network management framework requires major changes and disruptions to their networks. The consideration of three design goals, namely, ease of integration and adaption, replication of current business processes, and the support for technology migration is reflected in the architectural design of the adapter, and promises to pave the way for its acceptance and potential deployment.

The flexibility of the ONE adapter to add or drop auxiliary modules, based on the need of providers, allows the ONE adapter to be useful in heterogeneous carrier and IP networks (i.e., networks with different capabilities and based on different technologies), and be adaptable and also useful in the future network environment.

The use of standardized protocols and interfaces allows the ONE adapter to be easily adapted to the different network management systems (in terms of technology, languages, and processes).

This vision toward an easy to deploy, flexible, and cost effective mediator has also been reflected in the design of the ONE adapter’s modules.

6 References

[ASH10] A. Gumaste, N. Krishnaswamy. Proliferation of the Optical Transport Network:A Use Case Based Study, IEEE Communications Magazine, September 2010.

[BON03] K. Bontcheva and H. Cunningham. 2003. The Semantic Web: A New Opportunity and Challenge for HLT. In Proceedings of the Workshop HLT for the Semantic Web and Web Services at ISWC 2003.

[BRE07] Brendan Jenning et al. Towards Autonomic Management of Communications Networks..IEEE Communications Magazine, October 2007.

[BUI05] P. Buitelaar, P. Cimiano, and B. Magnini, editors. 2005. Ontology Learning from Text: Methods, Evaluation and Applications. IOS Press, Amsterdam, The Netherlands.

[CEL04] D. Celjuska, and M. Vargas-Vera, (2004) Ontosophie: A Semi-Automatic System for Ontology Population from Text. In *Proceedings International Conference on Natural Language Processing ICON 2004*, Hyderabad, India.

[D2.1] Deliverable D2.1, Definition of requirements and use cases, http://www.ict-one.eu/images/deliverables/onefp7-infso-ict-258300d2_1.pdf

[D2.2.1] Deliverable D2.2.1, Preliminary report on architectural design of the management adapter, http://www.ict-one.eu/images/deliverables/one_fp7-infso-ict-258300_d_2_2_1.pdf

[DMD02] A. Doan, J. Madhavan, P. Domingos, and A. Halevy, “Learning to Map between Ontologies on the Semantic Web,” In Proc. of WWW 2002, May 2002, Honolulu, Hawaii, USA.

[ERL07] Thomas Erl, SOA Principles of Service Design, Prentice Hall; 1 edition (July 28, 2007), ISBN-10: 9780132344821

[HEW09] Eben Hewitt, Java SOA Cookbook, O'Reilly Media; 1 edition (April 2, 2009), ISBN-10: 0596520727

[JUN11] http://www.juniper.net/us/en/dm/supercore/

[JUR03] Jurgen Schonwalder et al. On the Future of Internet Management Technologies.IEEE Communications Magazine 2003.

[JUR10] Jurgen Schonwalder et al.Network Configuration Management Using NETCONF and YANG. IEEE Communications Magazine,September 2010.

[LVA03] J. E. Lopez de Vergara, V. A. Villagra, J. I. Asensio, and J. Berrocal, “Ontologies: giving semantics to network management models,” IEEE *Network*, Vol. 17, No. 3. (2003), pp. 15-21.

[MAY08] Maynard, D., Li, Y.and Peters, W., NLP techniques for term extraction and ontology population. In: Buitelaar, P. and Cimiano, P. (eds.), Ontology Learning and Population: Bridging the Gap between Text and Knowledge, pp. 171-199, IOS Press, Amsterdam (2008)

[OKI10] E. Oki, T. Takeda, A. Farrel. Extensions to the Path Computation Element communication Protocol (PCEP) for Inter-Layer MPLS and GMPLS Traffic Engineering (Draft : draft-ietf-pce-inter-layer-ext-04.txt), July 2010.

[PIO07] Piotr Cholda et al.A Survey of Resilience Differentiation Frameworks in Communication Networks,IEEE Communications Surveys & Tutorials,2007.

[TLL06] J. Tang , J. Li , B. Liang , X. Huang , Y. Li , K. Wang , “Using Bayesian Decision for Ontology Mapping,” Journal of Web Semantics, Elsevier, Vol. 4, No. 4, 2006.

[WRP05] A. K. Y. Wong, P. Ray, N. Parameswaran, and J. Strassner, “Ontology mapping for the interoperability problem in network management,” Selected Areas in Communications, IEEE Journal on, Volume 23, Number 10, p.2058-2068 (2005).

[WS01] WSO2 ESB Success Stories, http://wso2.com/about/customers

7 Acronyms

[AAA]	Authentication, Authorization, Accounting
[BC]	Binding Component
[CAPEX]	Capital expenditures
[CLI]	Command Line Interface
[CORBA]	Common Object Request Broker Architecture
[ESB]	Enterprise Service Bus
[GMPLS]	Generalized Multiprotocol Label Switching
[HTTP]	Hypertext Transfer Protocol
[HTTPS]	Hypertext Transfer Protocol Secure
[IGP]	Interior Gateway Protocol
[IP-NMS]	Internet Protocol Network Management System
[ISO]	International Organization of Standardization
[JBI]	Java Business Integration
[JMS]	Java Message Service
[MPLS]	Multiprotocol Label Switching
[MTNM]	Multi-Technology Network Management
[MTOSI]	Multi-Technology Operations System Interface
[NE]	Network Element
[NMS]	Network Management System
[OPEX]	Operating expenditures
[PCC]	Path Computation Client
[PCE]	Path Computation Element
[PCEP]	Path Computation Element Protocol
[QoR]	Quality of Resilience
[QoS]	Quality of Service
[REST]	Representational State Transfer
[SE]	Service Engine
[SLA]	Service Level Agreement
[SNMP]	Simple Network Management Protocol
[SOAP]	Simple Object Access Protocol
[SU]	Service Unit
[TE]	Traffic Engineering
[TED]	Traffic Engineering Database
[T-NMS]	Transport Network Management System
[XML]	Extensible Markup Language