one

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 D2.2.1
Preliminary report on architectural design of the management adapter

Due date:	01/07/2011
Submission date:	6/1/2011
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 MySoft: Gabriela Aronovici, Vlad Melinte, Dan Horhoianu, Viorel Ionescu, George Dan Culache SNU: Jörn Altmann, Mohammad Hassan TID: Carlos García Argos, Óscar González de Dios, Javier Jiménez Chico, 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)

SEVENTH FRAMEWORK PROGRAMME

Table of Contents

Figure Summary

1 Executive Summary

Architectural design is a challenging endeavour in modern IT environments. On the one hand, product developers need to be compliant with existing infrastructures and the software already deployed in the field. On the other hand, the designer needs 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 describes the set of functional modules composing the ONE Adapter. Each of these modules presents a basis for creation SOA services, as it targets a set of objectives, which are essential to provide the functionality required to coordinate complex operations through the ONE Adapter. In the following deliverables, we shall elaborate upon the details of these modules and show how to use them, and how to implement a set of services in order to deliver the required functionality.

2 Introduction

This deliverable aims at presenting the first preliminary architecture of the ONE Adapter, based on the definition of requirements and the use cases that have already been reported in deliverable D2.1.

As we stated in the first deliverable (D2.1), the motivation for the ONE Adapter, as being a mediator between the IP and the carrier-grade management layers, has its roots in a decade-long efforts of standardization bodies to resolve the shortcomings of both IP and carrier layers through the development of control plane frameworks. In view of the fact that network operators are always in quest for ways to reduce the operational cost of networks, as well as introduce flexibility and increase network utilization, the proposed past frameworks were not useful in addressing the operational needs of network management in both the IP and transport layers. Despite the many advantages of the proposed standardized control plane frameworks, the current gap between the IP and transport management layers remains as large as ever.

One of the main challenges behind the proposed frameworks 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 which have proven to be bottlenecks in current telecom operations.

To avoid the cost and challenges of deploying and putting on production disruptive management technologies, we take into consideration three primary design goals, namely, ease of integration and adoption, non-disruptiveness, and support for technological migration.

Our architecture design of the ONE Adapter is devised to be useful in future networks, which are expected to be composed of heterogeneous technologies. The preliminary architecture design of ONE Adapter also provides network operators with the possibility to adapt the mediator model according to their needs.

The design of the ONE Adapter architecture is based on two principles, which are described in Section 3 of this deliverable. These principles comprise the separation between core and auxiliary modules.

In Section 4, we describe the core, and auxiliary modules, and their means of communication. The ONE core consists of four modules, namely the Ontology Mapper (OM), the Workflow Processing Module (WPM), the Management Controller (MC), and the Workflow Description Database (DB).

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

- Workflow Processing Module (WPM): The WPM is responsible for the execution of a specific workflow.

- Management Controller (MC) is responsible for coordinating the configuration of the ONE adapter, and is also responsible for the initial operations inside the ONE adapter before execution of the workflow.

- Workflow Description Database (DB): All workflows supported by ONE are stored inside the Workflow Description Database and provide a specification of a set of multi-layer operations which can be used repeatedly.

As we shall discuss in this document, 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:

- modules for topology lookup,

- measurement information collection,

- trigger module, ONE administration,

- IP-NMS control module, and

- T-NMS control module.

We also include description of 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 independently and in the ONE adapter as a whole.

The preliminary architecture design of the ONE Adapter is based on the Service Oriented Architecture (SOA), so the internal communication complies with the Enterprise Service Bus (ESB) architecture. In addition, by using standardized protocols and interfaces such as SNMP, NETCONF, and MTOSI, the ONE Adapter's 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 auxiliary modules can be added or removed in accordance to the network providers needs, assures the management adapter to 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:** In order to facilitate large-scale deployment, it is essential for the ONE Adapter to have the capability to integrate with the varied technologies (and their management systems). At the same time, integration of the ONE Adapter into the management ecosystem should be very simple and should not require (extensive) modifications in these subsystems to facilitate interaction with the ONE Adapter.

2) **Replication of Business Processes:** The ONE Adapter should have the capability to facilitate multi-layer coordination by exactly 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 operations of 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 a novel 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 management of the ONE Adapter as well as the orchestration of multi-layer processes, which can be a composition of interactions between different auxiliary modules in the ONE Adapter.

The Auxiliary modules in the ONE Adapter are responsible for interactions with the external actors inside a carrier's management ecosystem. Note here that the separation of the process orchestration and 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 based on the diverse choices of external actors available in the management ecosystem, that the auxiliary modules themselves may vary significantly when changing (or coordinating operation) between different technologies, and can therefore have 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 a *Trigger*, which is received from an external actor through the *Trigger Module*. 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 orchestration and processing of the workflow for the operation and communicate with other auxiliary modules, when necessary. After processing of 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 interaction, the ONE Adapter as a whole must also support 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.

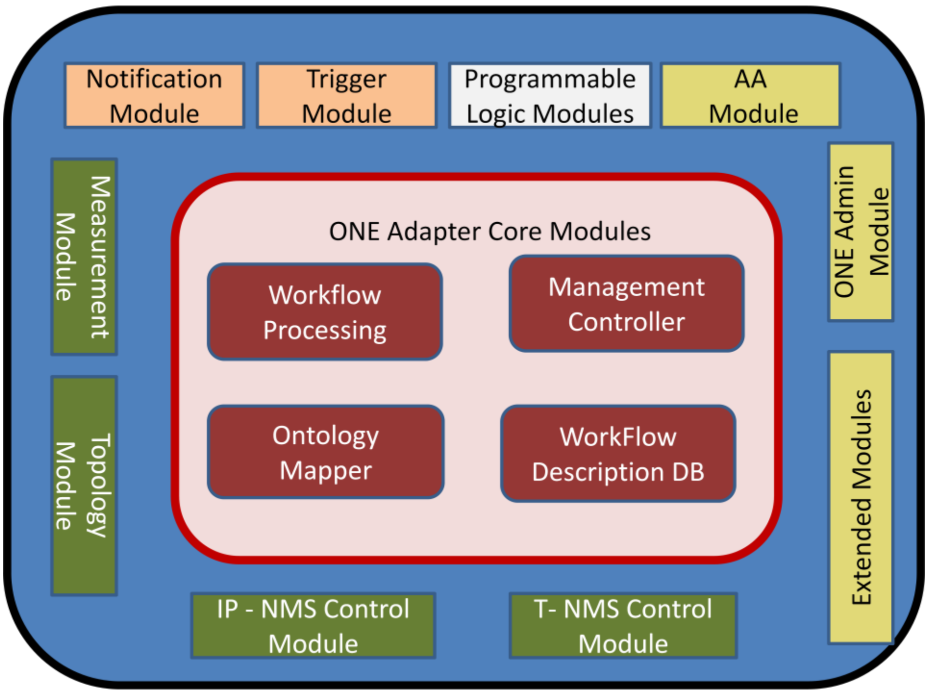

Figure 1: Architectural Overview of the ONE Adapter

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 functionalities:

1) **Easy and Flexible Configuration**: Network Operators should have the option of defining a multi-layer process as an orchestration of interactions between 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 orchestration between 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 in the later versions of the architecture document.

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

The four major modules of the ONE Core are shown in

Figure 1, namely, the Management Controller, the Ontology Mapper, the Workflow Processing and the Workflow Description Database. The Workflow Processing Module is responsible for orchestration and execution of a series of functions defined through a workflow. The workflows are stored inside the Workflow Description Database and provide a specification of a set of multi-layer operations which 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 makes sure that changes in an external actor or an auxiliary module does 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 operations before execution of a workflow. For example, when a trigger is received and is classified by the Ontology Mapper, the trigger is sent to the management controller which is responsible for facilitating authentication and authorization functions on the trigger. The management controller would also come into play to determine contentions inside the adapter in case two different triggers arrive at the same time at the ONE adapter.

3.2 ONE Auxiliary Modules

Our main design goal is to offer a pragmatic and easy-to-deploy adapter, enabling communication 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 which 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 make possible the interactions between these external actors and the adapter's core modules, a set of *auxiliary modules* is defined, which is depicted in

Figure **1** as a pool of satellite services around the core modules.

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:

- ***Trigger Module*:** this is the module in charge of receiving the external triggers that are required to initiate any coordinated operation through the ONE Adapter. In our design, the triggers may be initiated by a human operator, may come either from an IP-NMS or a Monitoring and Measurement system in the form of SNMP traps or from other external actors.

- ***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 service 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.

- ***Topology Module*:** this module is basically 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.

- ***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.

- ***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.

- ***Administration Module*****:** this module is responsible for managing the configuration of the ONE adapter per se. Updates in the workflow database, the inclusion of a new ontology, a new release of the mapping schemes, software upgrades, adding a new auxiliary module, etc., are all handled through this module.

- ***Notification Module:*** In the course of an operation by the ONE adapter, the workflow may dictate events requiring interactions (e.g. authorization) from human actors. Also, after completion of events, the workflows may dictate that the adapter notify the operator via some means. The Notification module will consist of mechanisms to facilitate the same.

The second category of auxiliary modules comprises those 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 may include (and are not limited to):

- ***Authentication, Authorization and Accounting (AAA) Module*****:** this module is the one 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.

- ***Programmable Logic Module:*** This module is a devised as a connector to external logic, network planning and optimization functions which may be necessary for computation of multi-layer operations. The module is designed specifically for operations consisting of complex computation based on the state of the network, which may be out of the scope of the ONE adapter core. Examples of the same may include, for instance, a PCC which may communicate with a PCE to support complex multi-layer path computation, and a connector to network optimization functions to facilitate IP offloading. By externalizing the logic functions, the operators may re-configure the algorithmic logic behind network operations without disturbing the operation logic programmed into the workflows.

The third category encompasses the modules which 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 with a Service-Level Agreement sub-system, a Billing System, etc.

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**Error! Reference source not found.**.

3.3 External Systems

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

The communication with the transport network should cover the following aspects:

- Configuration and provisioning requests through the MTOSI interface.
- Requesting information through MTOSI interface.
- Handling SNMP notifications.

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

In the IP network, on the other hand, 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 router network elements if needed. Therefore communication may be supported with either of the following options:

- The SNMP protocol,
- The Command Line Interface (CLI), and,
- The Network Configuration Protocol (NETCONF, RFC 4741)

The third actor considered is the human operator, which should be able to trigger certain workflows, and could also be contacted to get the approval before specific operations are carried out through 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, it is not yet defined which ones of the above will be supported in the time frame of this project.

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 is very flexible towards the constant evolution in terms of communicating with future external OSSs and Network Elements. The architecture of the ONE Adapter is inherently suitable for coordination both with multi-layer NMS systems as well as with external customer OSSs. Upcoming integrated NEs consisting of an hybrid IP router and optical switch [JUN11] should be no challenge to ONE, as it is capable to interact with those directly and/or 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 preliminarily discussed here, and 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 of 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 of fault tolerance.

Finally, the ONE adapter will also inherently support features of 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. Fault tolerance against operations can also be incorporated by network operators into their workflow definitions by requiring verification of actions.

3.4.2 Resilience

Resilience in the ONE adapter architecture will deal with the ability of the ONE adapter to operate in case all or 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 *core modules* are designed to be state-less in-between workflows so as to avoid issues of state synchronization in case of failures. Future iterations of the architecture will also address issues of synchronization (if any) between duplicate auxiliary modules at different locations so as to ensure stable operation in case of module failures.

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 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 authentication 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 AAA to provide authorization information when communicating with these Auxiliary modules to ensure that the initiator of the event has sufficient authorization to carry out the necessary network configurations.

Note here 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 which can significantly affect the regular operation of the network.

4 Modules Details

4.1 Core Modules

As stated previously, the ONE Core consists of four major functional modules: the Workflow Processor, the Workflow Description Database, the Ontology Mapper and the Management Controller. These four modules interact with each other to facilitate policy enforcement and process orchestration in the ONE Adapter.

4.1.1 Workflow Processor

The Workflow processor module is used to execute a process orchestration inside the ONE core. The processor receives the workflow definition from the workflow database and based on these definitions contacts the external modules to get configuration parameters or initiate configuration actions. Note that in a complex workflow, input parameters for a request are generated from a combination of one-or-more outputs from previous requests: here the workflow processor works in conjunction with the ontology mapper to facilitate the same.

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.

4.1.2 Workflow Description Database

The Workflow Description Database stores the workflows or orchestration definitions of the different processes defined by the 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.

4.1.3 Ontology Mapper

The fundamental role of the ONE Adapter is to enable the interoperability between two network management layers, which 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 in issues such as the semantic interpretation of the information contained in the triggers, the need 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*, and the semantic interpretations and interoperability issues are solved by means of *mappings* between ontologies [LVA03] [WRP05] [TLL06] [DMD02].

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, 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 two different ontologies.

In our design, we consider two possible applications of the Ontology Mapper module, namely, one for interpreting the information contained in the triggers, and the other for processing and adapting the required configurations to the specific command set of the equipment present in the network. To illustrate the first application, consider Figure 2, which shows one possible sequence of processing upon receiving an external trigger. When the request arrives to the Trigger Service (1), it first needs to be authorized (2). If successful, the message must be processed by the Ontology Mapper (3), which will semantically interpret the information contained in the trigger and will map it to a specific workflow (4) for subsequent processing and execution (5). It is worth highlighting that we are also exploring the possibility of endowing the adapter with a “Trigger Builder” tool, which may provide a programmable framework for telecom carriers through which they can build their own triggers. With this approach, carriers could be able to orchestrate a set of coordinated operations, which may be launched and driven by a sequence of actions embedded in the triggers. In this latter case, the complexity entailed in the first ontology mapping phase could be relaxed, since some of the requests received by the adapter are expected to match with carrier-defined triggers. For instance, a carrier-defined trigger might carry a tag or an ID, so the interpretation of the trigger can be reduced to a trivial check in this case. The capacity of semantically interpreting triggers must be present in the architecture, since the ONE Adapter is expected to interact with several external actors, some of which may request coordinated operations through the adapter, e.g., in the form of an SNMP trap, Web Service messages, etc.

The second phase of application of the Ontology Mapper module is illustrated in Figure 3. 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.

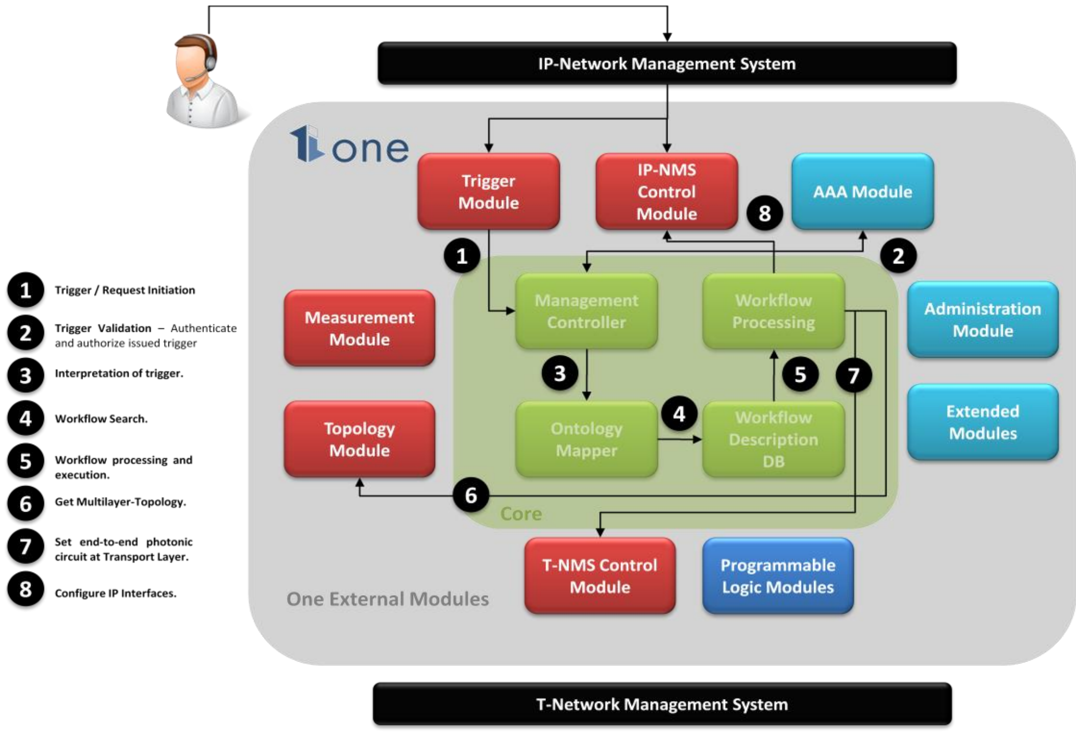

Figure 2: Semantic interpretation of the information contained in the trigger, and mapping the request to the corresponding workflow for internal processing and execution.

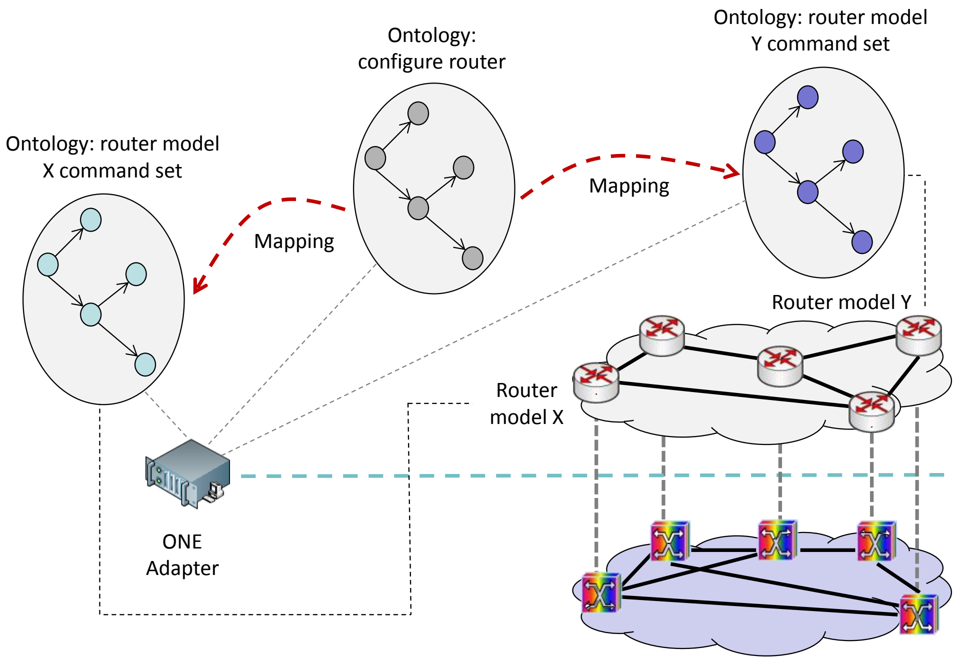

Figure 3: Semantic interpretation of the configurations required and mapping them to the specific command set of the devices involved.

4.1.4 Management Controller

The management Controller is primarily responsible for the configuration of the ONE adapter including the core and the auxiliary services. The management controller also acts as the initial point of contact after a trigger is received and is identified by the Ontology mapper. For example, after identifying a trigger, the Management controller may validate the authenticity of a trigger by communicating with an external AAA system. The management controller is also responsible for solving contentions inside the ONE adapter: for example, if a trigger arrives when a workflow is being processed, the management controller must determine if the current execution must be terminated and the new trigger processed first, or if the new trigger should be queued/discarded.

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 which 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 correlations between an IP link and the corresponding circuit in the transport layer. This information is the basis for constructing and maintaining updated 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 cross-checks, 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. For instance, upon receiving an external trigger, the ONE Adapter will start a coordinated process, during which the *Measurement Module* could be used to verify whether the event that triggered the operation is actually happening in the network (e.g., if the routers' interfaces indicated for IP offloading are effectively congested).

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.

4.2.3 Trigger module

It is worth emphasizing that any coordinated operation initiated through the ONE Adapter will be triggered by an external message. The *Trigger Module* is precisely the one that receives the external triggers, which may be issued by a human operator, an IP-NMS, a T-NMS, or by a management subsystem such as a Monitoring and Measurement system.

In our preliminary design, we consider basically two different types of triggers: 1) those initiated through Web Services; and 2) those coming in the form of SNMP traps. The first type covers the operation requests generated by a T-NMS, an IP-NMS (if present), and a set of operator-initiated tasks which could be facilitated through a front-end management server of the ONE Adapter. The second type gathers triggers that may be issued by a management subsystem, such as a Monitoring and Measurement system, a TE system, or a Policy control tool, where SNMP traps are sent either to alert that a network element has failed or that a policy rule is met.

Overall, the basic role of the *Trigger Module* is to enable the communication via Web Services and the SNMP protocol with a set of external actors, and provide support to process and route the external message, so that it can reach the corresponding internal block for subsequent processing and execution of the operation requested by the trigger. The information conveyed in a trigger could be as simple as "provision a new IP link supported by a transport circuit between two routers", or it can encode more complex requests, such as an alarm triggering coordinated restoration actions in the multi-layer network.

4.2.4 IP-NMS Control module

Management is a crucial functionality for Network Operators. This functionality can be split in two phases: monitoring and configuration. The Monitoring phase is in charge of receiving alarms and others manage information of the nodes in the network. The Configuration phase, as the name suggests, is in charge of the configuration and service provisioning operations in the IP network.

To date, standardization in the configurations of IP network is not mature. Current approaches are attempting to use standardized XML-based approaches definitions for operations and protocols like NETCONF (RFC 4741) have had a strong impact on 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 4, the ONE adapter must have the capability to interact with the IP NMS over a variety of protocols in order to facilitate configuration.

The process of configuration becomes significantly more challenging in the absence of an IP NMS. Given the diversity between interfaces to configure IP network elements, the control module relies 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. In short, the primary objectives of the IP-NMS control module are

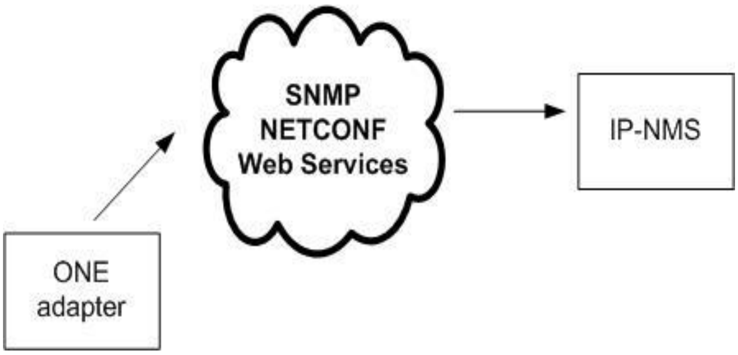

Figure 4: Example interactions from ONE Adapter to IP-NMS (Note: the communication is bidirectional)

4.2.5 T-NMS Control module

The ONE Adapter needs an interface with the capabilities of interacting with MTOSI interfaces but also with optional support to older interfaces available around T-NMSs like CORBA. The primary objective of the T-NMS control module is to facilitate communication with the T-NMS for configuration and service provisioning operations on the transport network. These functions are provided by standard interfaces such as MTOSI and will be exploited by the ONE adapter. As an example, MTOSI provides support for the activation and release of transport network services. While the interface to the transport network may be standardized, the layering and service activation requirements in transport networks may change significantly with the technology used, and the T-NMS control module must also have the capability to automatically provide these parameters, thereby reducing the visible complexity to operators when designing workflows.

4.2.6 Path Computation Client

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 a 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. The PCE eliminates the need for every node within the network to compute the path. There is no need for every node in the network to maintain a path computation database; there is now a central database for path computation. In order to do this 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 Figure 5. One of the auxiliary modules of ONE Adapter will act as a PCC offering access to features of 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 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.

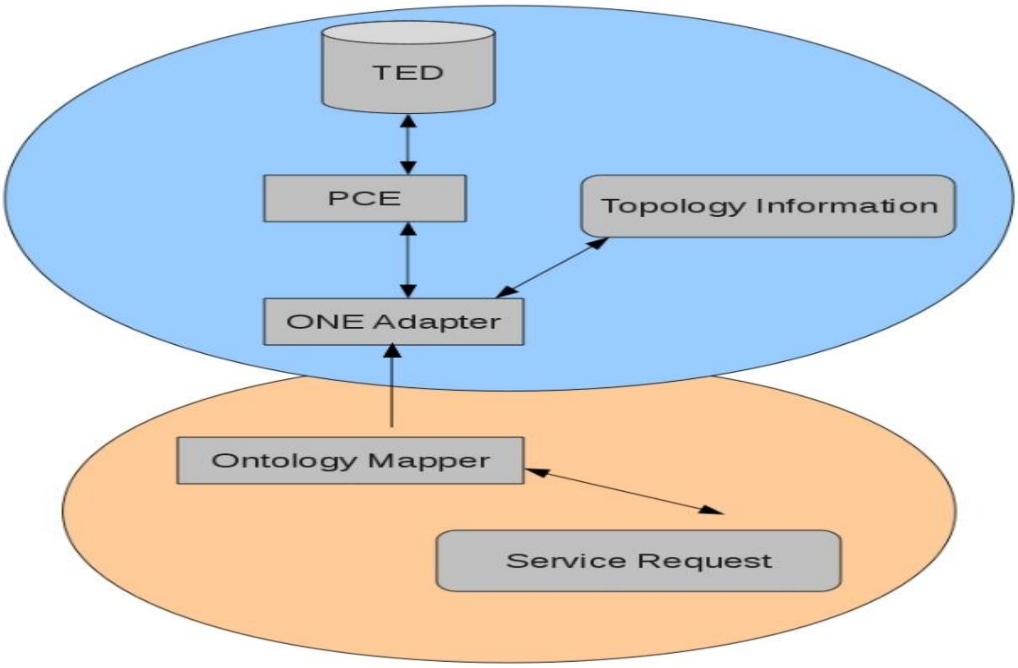

Figure 5: Inter-Communication between the ONE Adapter and the PCE

4.2.7 SLA Control

Service Level Agreement is a key feature in any network that considers offering 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 could be taken from external business systems or SLA parameters embedded in triggers. Depending on these parameters a provisioning of a service can be accomplished or not. Going further, a warning could be sent to the user when the demanded 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 percent rebate of service fees.

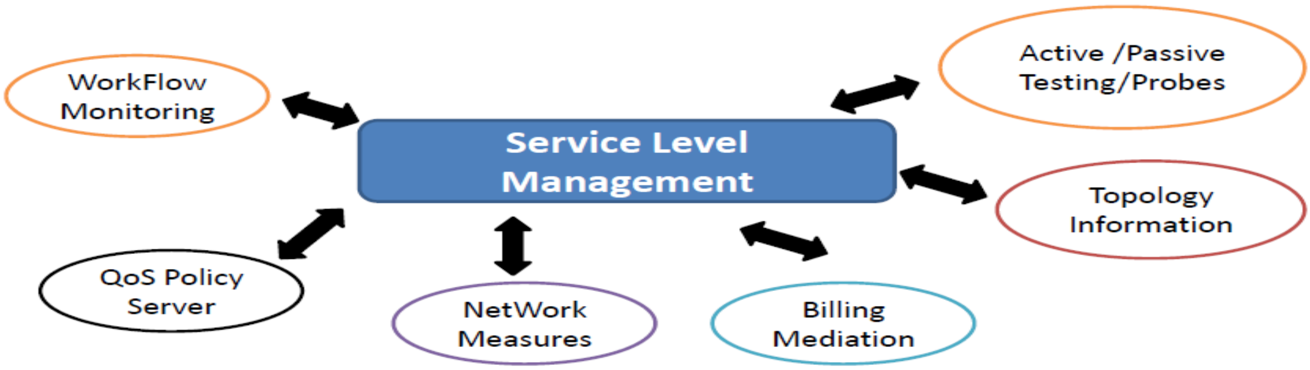

Figure 6: SLM Architecture

Figure 6 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 a XML template of SLA specifications that can be received by the ONE Adapter.

4.2.8 Accounting & Billing

Billing is the key commercial component for ISPs. A novel billing support is very important to open the doors and competing with seizing earlier opportunities, innovate tariff policies and in general flexibility in context of billing. Cost effective solutions demand from operators to change their billing policies depending on user’s localization, activity, market fluctuations, temporal situations, and type of payment.

The main objectives of this block are to support admission of protocols like DIAMETER and RADIUS to provide among many things billing, allowing communication to outside billing systems, such as AVABILL [AVA11] for offline charging and/or online charging. Billing functional block is related to SLA functional block. Depending of SLAS parameters, the value charged to user will vary. A charging function might depend on time, upload/download bytes, and transactions by the user.

This functional module may also facilitate exchange of network parameters as defined in the SLA as well as accounting information with the billing systems so that they can perform billing functions for specific multi-layer operations.

4.2.9 Authentication Authorization and Accounting (AAA)

Security is perhaps the most important issue in a network, besides its correct functioning. This block will provide access control to the ONE Adapter functionalities.

The AAA module will support authentication, authorization and accounting capabilities by facilitating communication with the providers AAA server. The authentication and authorization functions may be used to grant or deny permission to process a requested action, or for retrieving user access profiles. These access profiles define levels of access to certain functions and operations of the ONE Adapter. At second instance in a future, the ONE Adapter will support secure communications, in terms of validating possible requests. The ONE adapter may also facilitate initiation of accounting sessions and exchange of accounting information with billing and charging systems to facilitate accounting of multi-layer operations performed by the ONE adapter.

4.2.10 Administration module

As any network equipment, the ONE Adapter has to be able to offer the possibility to execute administration tasks over it.

This module has to give capabilities to perform reconfiguration actions to the ONE Adapter. These actions can be grouped as the ones that affect the external functions of the ONE Adapter and the ones that affect the internal functions. External functions are those related to the interaction of the ONE Adapter to external systems. Internal functions affect the functioning of the ONE Adapter, for example: regular management task (access passwords) and tuning Ontology definitions of the Ontology Mapper.

4.3 Internal Communication

4.3.1 Service Engines and Binding Components

ONE will be designed on a Service Oriented Architecture (SOA) paradigm [ERL07] [HEW09], so internal communication complies with Enterprise Service Bus (ESB) architecture. In this chapter as well as throughout the document "services" and "modules" are interchangeable concepts.

Service engines (SEs), service units (SUs) and binding components (BCs) are the components in the ONE Adapter environment. SEs and SUs provide services to other components and BCs allow external systems to communicate with the adapter environment. An SU provides functionality to other components and can consume services provided by other components. Already existing custom components that provide reusable functionality can easily be made available as a service engine. However, when a service will be available through an SE, only other adapter services can access this service.

The binding component (BC) is a component that provides connectivity to existing applications and services that are located outside the adapter environment. If ONE wants to communicate with existing applications, it will be possible to do that by using binding components. The same is true for the integration with communication protocols (such as HTTP, SNMP, JMS, etc.). Besides providing access to external services, BCs can be used to expose internal services to the outside world.

Generally speaking from a functional perspective, SEs map to ONE core modules and SUs to ONE auxiliary modules as described further in ONE building blocks.

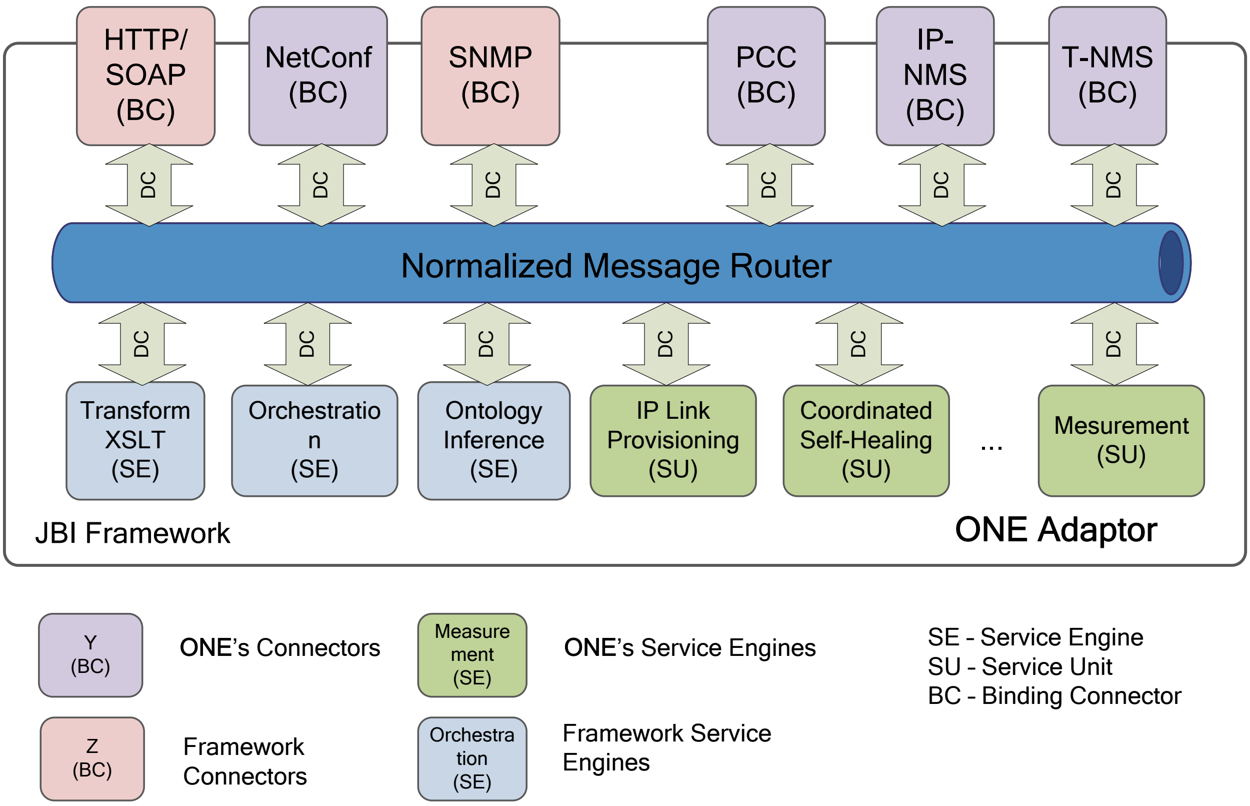

Figure 7: ONE's Internal Communication via Delivery Channel

Consumer and *provider* are the two roles a component inside the ONE Adapter can have. If a component provides services to another component, the component's role is the provider. If a component uses a service provided by another component, it consumes this service, and the component is called a consumer.

Figure 7 shows the ONE adaptor in which several components are installed. Among the BCs we note SNMP BC, HTTP/SOAP BC and ONE specific BCs such as NetConf BC, PCE BC, IPNMS BC, TNMS BC. As shown BCs do not just allow incoming communications; it also provides a way for the other components in the container to communicate with external technologies.

Inside the adapter we see several SEs. An XSLT SE that transforms XML, an Orchestration SE, an Ontology Inference SE and several ONE specific SUs such as IP Provisioning SU.

4.3.2 Services and Endpoints

A service can't be accessed directly and it is necessary to use an endpoint for the same. Each service must have at least one or more endpoints. In order to utilize a service provided by a component, it is necessary to know the name of the service and the name of the endpoint to invoke. This combination of a service and a specific endpoint on that service is called a service endpoint.

4.3.3 Normalized Message Router (NMR)

The NMR is the core of ONE's internal communication. It contains ESB functionality including the Routing SE which fulfils the Management Controller functionality in the ONE Core Modules. It is based on a Rules Engine and involves the necessary logic to determine the ultimate destination for an incoming message and executes the appropriate workflow/service.

In Figure 8**Error! Reference source not found.**, at the centre of all the components is a component called the normalized message router (NMR). This means that the components (SE. SU or BC) don't directly communicate with each other—they communicate using the NMR. The components don't connect directly to this NMR, but instead use a delivery channel (DC). It's the NMR's job to make sure that the messages are exchanged correctly among the various components in the Java Business Integration (JBI) environment.

The NMR can exchange messages in a number of ways, or *patterns*. The following is the list of patterns that must be supported by a JBI implementation. Consider each pattern from the provider's point of view.

In-Only —with this pattern the consumer makes a request but doesn't expect a response back from the provider. This also means that if a fault occurs, this fault isn't sent back to the consumer.

Robust-In-Only —this pattern is similar to the previous one, only this time the provider can send a fault message if something goes wrong.

In-Out —in this traditional request/response scenario, the consumer sends a request and expects a response from the provider. If an error occurs, the provider is free to send a fault message.

In-Optional-Out —this pattern is similar to the previous one, only this time the response is optional, and during the message interaction both parties can send a fault message.

Figure 9 shows a detailed description of the interaction between the provider, consumer and the NMR for an *In-Out* message exchange. The steps executed here are:

1) The consumer creates a message exchange for a specific service and endpoint (this combination of service name and endpoint is called the Service-Endpoint).

2) The consumer sets the “in” part of the message exchange with the request body. After this step, it sends the message by putting it on its delivery channel, and thus sending it to the NMR.

3) The NMR determines to which provider this exchange needs to be sent and queues it for delivery to the provider.

4) The provider accepts this message exchange.

5) The provider executes its business logic.

6) After the provider has finished processing, the response message is added to the “out” part of the message exchange and the message exchange is again presented to the NMR.

7) The NMR once again queues the message for delivery to the consumer.

8) The consumer accepts the message exchange.

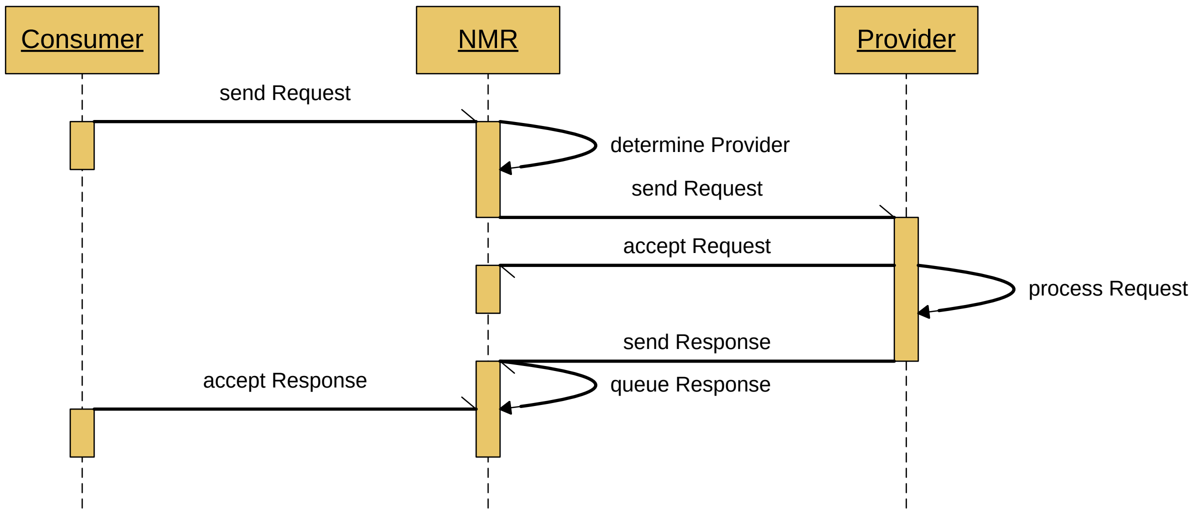

Figure 8: In-Out Message Exchange

The Normalization and De-Normalization processes are depicted in more detail in **Error! Reference source not found.**.

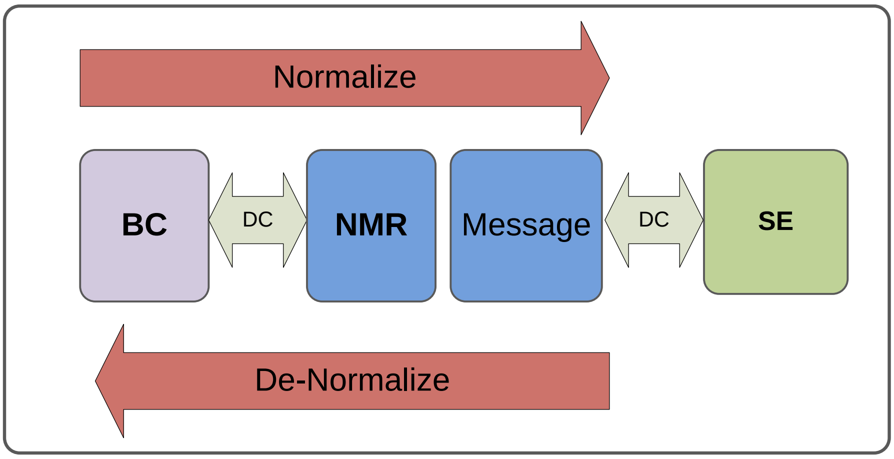

Figure 9: Close Up Messaging

4.3.4 Message Normalization

All message exchange patterns in ONE adaptor comply with the Web Services Description Language (WSDL) 2.0 specification. As for the external messages they are normalized to WSDL 2.0 format as well. As it can be seen from the previous patterns, they're all written from the perspective of the provider.

5 Conclusions

The preliminary design of the ONE Adapter architecture, which is reported in this deliverable, has been derived from the fact that today's network providers do not accept disruptions of their production environment. It is not acceptable to network service providers, if a network management framework is not affordable and requires major changes to their network. The consideration of three design goals (i.e., ease of integration and adaption, replication of current business processes, and the support for technology migration) in the preliminary architecture design of the ONE Adapter assures its desirability and wide 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), adaptable and 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 modules.

6 References

[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

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

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

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

[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.

[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.

[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.

[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).

[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.

[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.

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

[AVA11] http://www.avabill.com/

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
[PCE]	Path Computation Element
[QoR]	Quality of Resilience
[QoS]	Quality of Service
[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