



## Deliverable D4.15

# Synthetic analysis of deployment results and impacts

<b>Grant Agreement</b>	257513	
<b>Date of Annex I</b>	08 October 2013	
<b>Dissemination Level</b>	Public	
<b>Nature</b>	Report	
<b>Work package</b>	WP4 - Deployment and Impacts	
<b>Due delivery date</b>	01 December 2013	
<b>Actual delivery date</b>	09 November 2013	
<b>Lead beneficiary</b>	TID	Beatriz Fuentes (fuentes@TID.ES)

<b>Authors</b>	<p>ALUD: Ingo Karla, Markus Gruber</p> <p>FT: Zwi Altman, Christian Destré, Imen Grida Ben Yahia</p> <p>iMinds: Vânia Gonçalves, Sander Spek</p> <p>NKUA: George Katsikas, Roi Arapoglou, Eleni Patouni , Nancy Alonistioni</p> <p>NTT: Wenyu Shen, Yukio Tsukishima</p> <p>TID: Beatriz Fuentes</p> <p>UCL: Stuart Clayman, Alex Galis, Lefteris Mamatas</p> <p>UPRC: Panagiotis Demestichas, Kostas Tsagkaris, Aristi Galani, Vasilis Foteinos, Vera Stavroulaki, Giorgios Poullos, Yiouli Kritikou, Marios Logothetis, Dimitris Karvounas, Andreas Georgakopoulos, Assimina Sarli, Evangelia Tzifa, Nikolaos Koutsouris, Petros Morakos, Alexandros Antzoulatos, Aimilia Bantouna</p> <p>VTT: Olli Mämmelä, Teemu Rautio, Jukka Mäkelä, Petteri Mannersalo</p>
----------------	---

## Executive summary

Deliverable D4.15 “Synthetic analysis of deployment results and impacts” reports on the demonstration results and presents an assessment of the technical and business impact of the solutions developed in UniverSelf. It brings together the simulation, prototyping, experimentation and demonstration work performed in Task 4.2, and the analysis of the technological and socio-economic impact of the technologies developed within the project (outcome of Task 4.3).

This document is structured into two main parts. The first one represents a consolidated report on the UniverSelf final integrated prototype, including a description of the testbeds, the deployment architecture of the UMF framework and the implemented Network Empowered Mechanisms (NEMs) and the three scenarios designed to showcase the prototype. Each of the scenarios seeks to highlight different aspects of network management in different network environments: QoS and SLA-aware self-management in SDN (Software Driven Networks) core and WLAN access network, conflict-free coordination of SON NEMs and UMF management of SDN. On one hand, the integrated prototype demonstrates how different network elements can be empowered with intelligent autonomic mechanisms in the form of NEMs; and on the other hand, it showcases how UMF enables the coordination, knowledge exchange and governance of these mechanisms. The results validate the efficient and reliable operation of the NEMs and their trustworthy interworking with the UMF core blocks. The technical study is complemented with a business assessment based on one of the scenarios, using quantitative and qualitative methodologies. This scenario was selected for the economic analysis as it is considered a representative case of challenges to be faced by operators in the future. The results of the OPEX analysis foresee an OPEX reduction for a typical network operator of 11% to 13% following the introduction of autonomic functions and UMF as management framework.

The second part of this document consists on a theoretical analysis of how UMF could be deployed in existing and emerging network architectures, and in standardized network management architectures. In the case of existing architectures, Metro Ethernet, FTTH and Heterogeneous Mobile networks were chosen as representative cases for the analysis. The study concludes that, although the incorporation of modifications in the management systems of current mature technologies is difficult in general, UMF could be deployed following a step-by-step migration strategy. The deployment analysis then moves to the emerging network infrastructures SdN (Software defined Networks) and NFV (Network Functions Virtualization), showing how those can benefit with the introduction of autonomic mechanisms and the management approaches offered by UMF. Finally, UMF is positioned with respect to existing network and management architectures (3GPP LTE and eTOM). The synthesis of the aforementioned assessment results establishes the UMF feasibility and readiness for deployment.

# Table of Content

<b>1</b>	<b>Introduction</b>	<b>7</b>
<b>2</b>	<b>Deployment Assessment of Key UniverSelf Solutions</b>	<b>8</b>
2.1	Testbeds description	8
2.1.1	SDN-based Core and WLAN Access Testbed	8
2.1.2	SON Coordination LTE RAN Testbed	9
2.1.3	Very Lightweight Software Driven Network and Services Platform (VLSP)	9
2.2	Deployment architecture	11
2.2.1	UMF Deployment	11
2.2.2	NEM Deployment	20
2.3	Scenarios and Deployment Results	27
2.3.1	Scenario 1- A Unified Framework for QoS and SLA-aware multi-domain self-management	27
2.3.2	Scenario 2 - Conflict-free Coordination of SON NEMs	34
2.3.3	Scenario 3 - UMF Management of Software Driven Networks (SDN)	38
2.4	Conclusions	43
<b>3</b>	<b>OPEX Impact Analysis</b>	<b>45</b>
3.1	Toy model methodology	45
3.1.1	Overview	45
3.1.2	NEM metric inputs	47
3.1.3	Conversion tables	48
3.2	Model results	52
3.2.1	Individual NEM results	52
3.2.2	NEM impact aggregation	53
3.3	Discussion of results	54
3.4	Conclusions	56
<b>4</b>	<b>UMF deployment in existing network infrastructures</b>	<b>57</b>
4.1	Deployment Study of UMF in Metro Ethernet Service	57
4.1.1	Introduction to Target Metro Ethernet Service	57
4.1.2	Current Network Management System	57
4.1.3	Basic Principles of Deploying New Technologies	60
4.1.4	UMF's Deployment Scenario	61
4.2	Deployment of UMF in FTTH network	64
4.2.1	GPON Architecture	64
4.2.2	FTTH-related Support Systems	65
4.2.3	UMF deployment on FTTH Networks	67
4.3	Deployment of UMF in Heterogeneous Mobile Network infrastructure	69
4.3.1	Context – heterogeneous network with small cells	69
4.3.2	UMF deployment in heterogeneous cell networks	70
4.4	Migration of UMF and NEMs from Legacy Networks	71
4.5	Conclusions	73

<b>5</b>	<b>UMF deployment in emerging network infrastructures</b>	<b>74</b>
5.1	UMF deployment in Software Defined Networks	74
5.1.1	SdN main concepts	74
5.1.2	Connecting UMF to SdN enabled infrastructures	75
5.2	UMF deployment in Network Functions Virtualization Architecture	79
5.3	Conclusions	80
<b>6</b>	<b>UMF positioning on existing network and management architectures</b>	<b>81</b>
6.1	UMF functions in 3GPP-LTE systems	81
6.1.1	Governance	81
6.1.2	Coordination	83
6.1.3	Knowledge	84
6.2	UMF mapping in eTOM	84
6.3	Conclusions	85
<b>7</b>	<b>Conclusions</b>	<b>86</b>
<b>8</b>	<b>References</b>	<b>87</b>

## Foreword

This deliverable concludes the implementation, integration and demonstration work performed in WP4, and reports on the deployment and impact analysis results, outcomes of the work in Tasks 4.2 and 4.3, respectively. At the same time, this document closes the design-implementation-assessment cycle defined in UniverSelf project, which involved activities in WP2 – Unified Management Framework, WP3 – Network Empowerment and WP4-Deployment and Impacts.

WP2 produced the specifications of the Unified Management Framework (UMF) in Deliverable D2.4 [1]. UMF is a management framework that aims to solve actual network problems and address the growing management complexity of future networks. The novel characteristics are achieved through the smooth embodiment of empowered autonomic mechanisms into both services and networks, and the specification of core blocks for their governance, their coordination and the management of the knowledge produced and consumed by those functions. To this effect, UMF introduces the concept of Network Empowerment Mechanisms (NEMs), which encapsulate autonomic functions that can be embedded into legacy and future systems, and the UMF core blocks, namely Governance, Knowledge and Coordination. The ultimate goal of the UMF is to ensure the trustworthy integration (plug and play), operation and interworking (conflict avoidance and knowledge sharing) of multiple NEMs destined to the management of networks and services within the operator's environment.

WP3 designed and evaluated the methods/algorithms to be embedded as NEMs, which solve particular operational problems of a given network segment or service architecture. This work-package focused on algorithms for parameter configuration and optimization, learning solutions for the network to trigger responsive actions, and strategies for cooperation between NEMs. The designed methods were implemented and tested in isolation using analytical modelling, simulation/tests, and emulations. The results of the development and evaluation of these methods have been reported in the WP3 Deliverables [2][3][4][5].

WP4 was responsible for implementing, deploying, assessing and analysing the technical and business impact of UniverSelf solutions to a set of identified problems, defined in the form of use cases. In this context, the core blocks of UMF specified in WP2 in terms of functions and operations were developed and integrated with the appropriate NEMs. Finally, the solutions to the use cases were used to build three use case prototypes and one integrated prototype. In particular, this document is related to the following WP objectives:

- Contribute to project full integration
- Demonstrate that UniverSelf solutions are deployable and produce an impact
- Assess, according to key success criteria, UniverSelf impacts in terms of solving problems, reducing costs, creating new business opportunities
- Develop strategies for Network Operators to adopt autonomic solutions
- Develop migration strategies to provide an incremental integration of autonomics into current and future networks

Previous WP4 deliverables have already partially reported on the work performed in this work package: Deliverables D4.6 [6] and D4.12 [7] on deployment results, D4.7 [8] and D4.14 [9] on the impact analysis of UniverSelf solutions, and D4.4/D4.5 [10]-[11], D4.8/D4.9 [12]-[13], D4.10/D4.11 [14]-[15] and D4.13 [16] on the prototypes.

In this context, the main goal of this deliverable is to provide a final public overview of the deployment results as outcome of Task 4.2, and the business and impact analysis as outcome of Task 4.3. The analysis of UMF deployment is not limited to the scenarios developed in the project, but it is completed with a theoretical study of how UMF could be deployed in existing and emerging network architectures. This document gathers not confidential material from deliverables D4.12 and D4.14.

# 1 Introduction

The main objective of this deliverable is to present an overview of the assessment of UniverSelf solutions, both from a technical and economical perspective. The analysis is not limited to the scenarios developed in the project as part of the implementation and integration work, but it is completed with a theoretical study of how UMF could be deployed in existing and emerging network architectures. These two elements conform the two parts in which this deliverable is structured.

The first part presents an assessment of the technical and business impact of the solutions developed in the project. In UniverSelf, a significant number of NEMs have been implemented, and can be classified into three groups: NEMs that are not part of UniverSelf Use Case prototypes [10][12] or the integrated prototype, the NEMs that are part of Use Case prototypes, and the NEMs that are part of the integrated prototype. The deployment analysis in this deliverable will focus on the integrated prototype, given that it integrates a selected set of NEMs with the UMF core, and it is built upon the use case prototypes. Moreover, the results of each of the individual implemented NEMs, being them part of prototypes or not, have already been reported in public WP3 deliverables [2]-[5]. This first part of the document comprises Chapters 2 and 3. The former presents the assessment of deployment of key UniverSelf solutions for the integrated prototype, and presents the topology of the testbeds involved, the deployment architecture of the UMF and NEMs on the testbeds, the scenarios designed for the integrated prototype, and the deployment results. Chapter 3 contains the OPEX impacts of one of the scenarios of the integrated prototype, using a so-called Toy Model that helps to calculate the impact of autonomics in a Network Operator's OPEX.

The second part of the deliverable presents the possible deployment of UMF functions on existing network and management architectures. The ultimate goal is to demonstrate that the project solutions could be deployed in real networks at production level. Chapter 4 includes an analysis of UMF deployment in existing network infrastructures, and in particular in the Metro Ethernet service, in the FTTH access network, and in Heterogeneous Mobile networks. This chapter also presents an elaboration on the migration from legacy networks and management systems to a network empowered with NEMs and governed by UMF core, based on information learned from the UniverSelf demonstrators and from field trials with commercial network operators. Chapter 5 presents UMF deployment in emerging network infrastructures, Software Defined Networks (SdN) and Network Functions Virtualization (NFV) Architecture. Chapter 6 gives an overview of the UMF positioning on existing network and management architectures; specifically UMF is mapped to 3GPP LTE network architecture and eTOM. This second part of the document aims to present the information in a handbook-like structure, where the reader can look up the UMF deployment architecture for a set of representative existing or emerging network infrastructures.

Finally, Chapter 7 concludes the deliverable by summarising the outcomes of this synthesis of deployment results and impact analysis.

## 2 Deployment Assessment of Key UniverSelf Solutions

During the life of the UniverSelf project, quite an important number of elements have been implemented and tested. This chapter focuses on the available results coming from the final prototype, which integrates the UMF with NEMs from different use cases. The ultimate goal of this prototype is twofold: on one hand, to demonstrate how different network elements and network segments can be empowered with intelligent autonomic mechanisms; on the other hand, to illustrate how these NEMs can be coordinated, can exchange knowledge and be governed by the UMF core blocks, considering a common stable implementation of them.

A pre-final integrated prototype, shown at FUNEMS 2013, has already been reported in D4.13 [16]. The main advancements of the final integrated prototype are:

- The integration of the complete NEM lifecycle management. The NEM lifecycle, defined in D2.4 [1], is the set of phases that a NEM goes through its existence. The lifecycle describes the way a NEM can be dynamically instantiated, started, activated, halted and stopped.
- The development of the UniverSelf dashboard. The dashboard is a Graphical User Interface that incorporates the Governance, Coordination and Knowledge building mechanisms and provides to the operator the capability to instantiate, deploy and monitor NEMs. The NEM lifecycle management is enabled by the visualization of the virtual and physical infrastructure. Screenshots of the dashboard will be included in the different sections of this chapter (see Figure 15 and Figure 39).

Furthermore, while in D4.13 the reporting mainly focused on the development and integration activities, in this document the focus is set on deployment and results.

The final prototype is presented in the following subsections:

- The underlying network topology consisting of three testbeds.
- The deployment architecture, describing how the mapping between the software solutions and the available hardware.
- The three scenarios which conform the demonstration of the final prototype and their assessment, describing the results obtained with the different elements and how they work together to achieve the objectives in each of the scenario.

### 2.1 Testbeds description

This subsection presents a brief description of the available network testbeds that will constitute the playground for the UniverSelf integrated prototype: SDN-based core and WLAN access testbed, SON coordination LTE RAN Testbed and the Very Lightweight Software Driven Network and Services Platform testbed.

Software Driven Networks (SDN) is used on purpose instead of Software defined Networks (SdN). Usually, SdN is linked to the OpenFlow protocol that made it popular, while using SDN we want to highlight that the work performed in UniverSelf is not tight to a particular protocol, but corresponds to a more general concept in terms of the different protocols that could be used (e.g. FORCES, wireless controls, etc). In the rest of this deliverable, SDN will be used for Software Driven Networks, while SdN will stand for Software defined Networks.

#### 2.1.1 SDN-based Core and WLAN Access Testbed

This testbed targets to validate UMF functionality over a multi-domain, multi technology data plane, respecting also QoS and SLA constraints. Specifically, the network topology under consideration consists of a real Wi-Fi access segment that is comprised of one router and two Access Points (APs), and an SDN-based core network topology that emulates an IP core network and is comprised of seven virtual routers. A video server is also used to inject the service while three end users are connected to the Wi-Fi access part to consume the video. The overall topology is depicted in Figure 1.



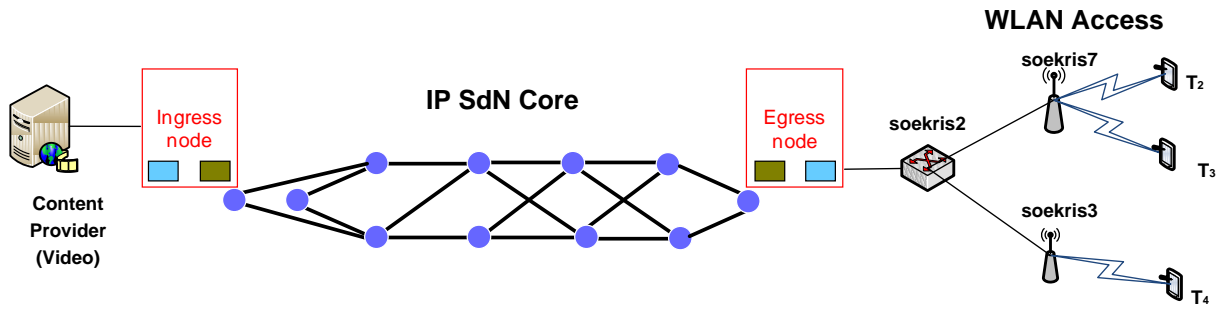


Figure 1: SDN-based Core and WLAN Access Data plane.

### 2.1.2 SON Coordination LTE RAN Testbed

The SON Coordination scenario takes place on a simulated LTE RAN testbed. The simulator mechanics are developed in Matlab while the view of the network is visualized using Java (using the Matlab built-in interoperability mechanism).

In the scenario, the network consists of 1 LTE macro cell (named *CellB* in Figure 2) and 4 small cells (Femtos, see *FemtoB.{1-4}* in Figure 2) backhauled through the Macro cell using wireless links (in-band relaying). Omnidirectional antennas are modelled on a cell radius of 2 km, a Rayleigh fast-fading model and 46dBm/30dBm transmit power for macro/small cell antennas respectively. OFDMA is the access technology used, while 10 resource blocks of 180 kHz each are assumed. Users (see shades of blue in Figure 2) arrive randomly in the cells following a spatial Poisson process with, initially fixed and equal arrival rates for each Femto. Each user is assumed to download a 10MB file and then leave the network.

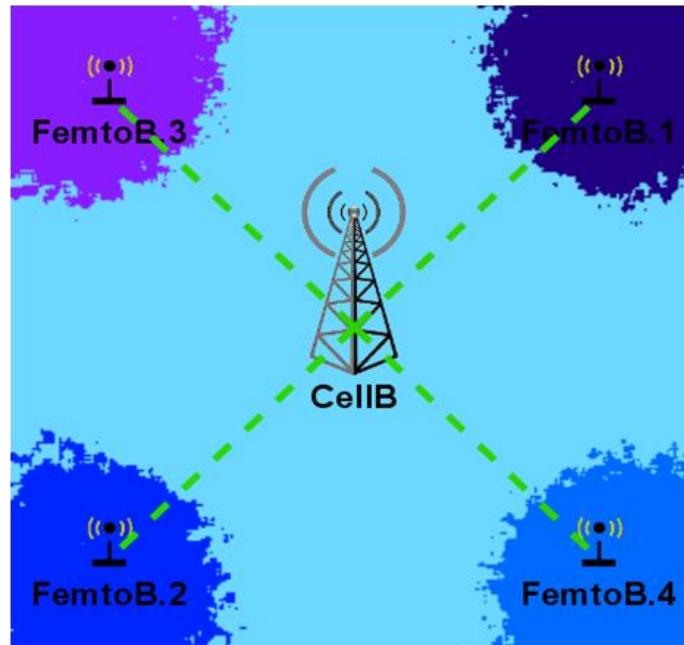
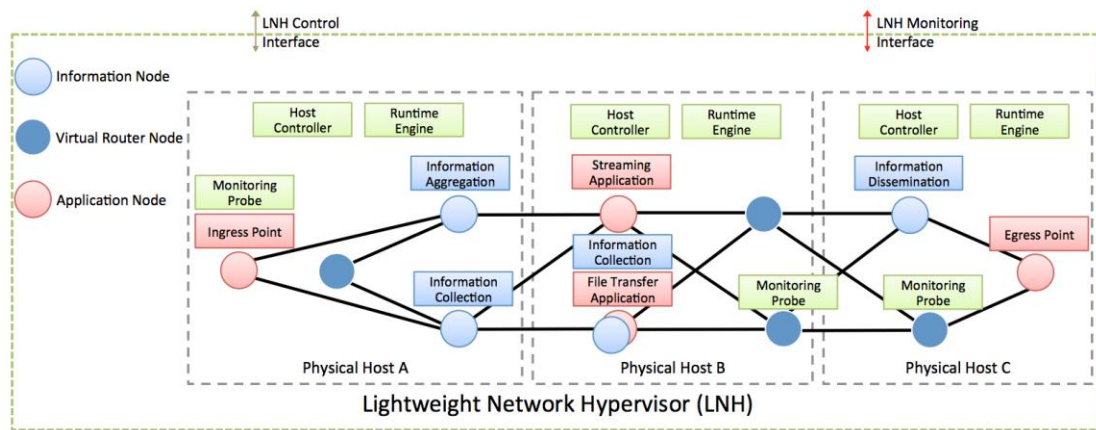


Figure 2: Simulated LTE SON used on the SON Coordination scenario

The simulator features an HTTP RESTful interface both for the monitoring and the setting of configuration parameters as well as for the monitoring of KPIs from NEMs.

### 2.1.3 Very Lightweight Software Driven Network and Services Platform (VLSP)

The Very Lightweight Software Driven Network and Services Platform (VLSP) test-bed includes a new lightweight network hypervisor, on top of which a NEM for the management and manipulation of virtual networks and the UMF core are deployed.



**Figure 3: The Lightweight Network Hypervisor**

The Lightweight Network Hypervisor includes a fully operational lightweight virtual router (VR) combined with virtual network connectivity. These elements can be combined in order to build any network topology required. The created virtual network is designed with the goal of transmitting and routing datagrams from any source to any destination. It behaves like a lightweight virtual network, but it has management facilities to start and stop virtual routers on-the-fly, together with the ability to create and tear down network connections between virtual routers dynamically. Furthermore, these lightweight routers have an application layer interface that provides the capability to start and stop Java software applications. These applications use a transport protocol API which can send and receive datagrams or packets, and thus act as the service elements within the platform.

The Host Controllers (shown in green in Figure 3) are executed on every machine that can host virtual routers. Their main job is to actually start a virtual router, stop a virtual router, start a virtual link, and stop a virtual link. Other tasks undertaken by the Host Controllers are to configure the routers once they have started, or to pass on commands to specific virtual routers, as needed.

The virtual network topology consists of virtual routers (shown as coloured circles) and virtual links (shown in black). Each virtual router is instrumented with the VLSP monitoring system in order to gather data on each of the network interfaces of each virtual router. The data includes information on traffic volumes coming in and going out of each interface. The monitoring system collects the raw data and passes it onto the Monitoring Engine function of the above layer.

The main LNH functions specification can be found in the following table.

Name	Description
Host Controllers	The host controllers execute on every physical machine and manipulate & configure virtual routers, links and virtual router applications.
Monitor Probes	The monitor probes are tiny configurable applications probing the software or hardware for monitoring data.
Runtime Engine	It is responsible for the runtime operation of the LNH, including support for event-based notifications and time scheduling.
Virtual Router Protocol Stack	The lightweight network protocol stack of the VRs.
Virtual Router Application Environment	The application environment that hosts VR applications.

Virtual Link Functionality	The functionality of the virtual links, including link weighting and other configuration options.
Virtual Machine for Virtual Router & Application Functionality	A virtual machine with the virtual router and the relevant applications functionality.

## 2.2 Deployment architecture

This section describes how the UMF core blocks and NEMs are deployed in the three testbeds. Section 2.3 will describe how the different components interact to build three scenarios. Even when the three scenarios share and use the same implementation of the UMF core, they have been designed so that each of the scenarios highlights the functioning of one of the core blocks.

### 2.2.1 UMF Deployment

This section summarizes the deployment of the three UMF core blocks: Governance (GOV), Coordination (COORD) and Knowledge (KNOW).

#### 2.2.1.1 Governance

This section presents a detail description and results of the deployment of Governance core block used in the different UniverSelf use cases and prototypes. Governance block responds to the need of the human network operators to supervise the functioning and control the behaviour of not only the underlying autonomic functionalities (NEMs), but of the management system as well (UMF core blocks).

The specifications of the UMF Governance core block [1] identify four main functions as necessary for the proper governance of future networks and services:

- Human to Network function
- Policy Derivation and Management function
- NEM Management function
- Enforcement function

The Human to Network (H2N) function allows the human network operator to manage the whole autonomic network, services and the other components of the UMF system by defining high level objectives. These objectives are expressed in a close-to-natural language and therefore need to be translated to a language that can be understood by the targeted elements. This translation process is implemented in the Policy Derivation and Management (PDM) function, which produces as output low level policies. Enforcement function transmits then the final policies to the corresponding NEM or UMF component, while NEM Management function allows for a fine grained control of the NEM life-cycle.

Figure 4 below shows a schematic representation of the Governance block and its functions. The diagram highlights the external interfaces of the Governance block, towards the NEMs and with the other UMF core blocks, namely Coordination and Knowledge.

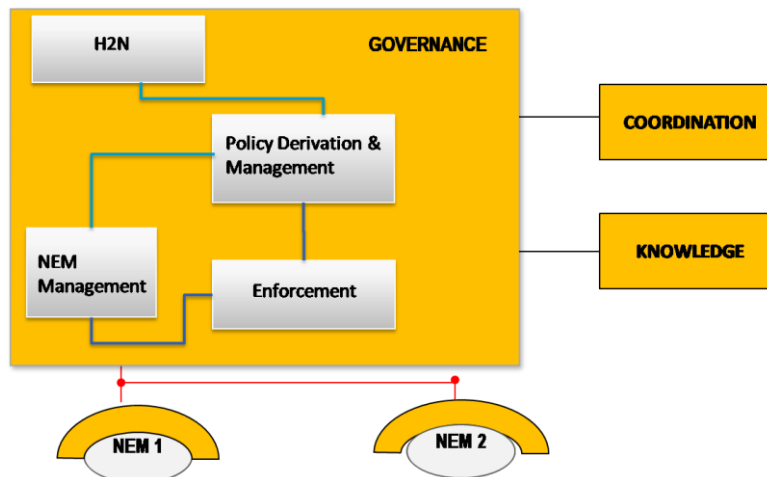


Figure 4: Schematic representation of Governance core block

All the Governance functions and their main operations have been implemented as part of the use cases work in the project. This section reports on the deployment results of the Governance functions.

### Human to Network function

The Human to Network function provides a friendly tool for the human operator to dynamically define high-level objectives and user requirements, which will be later on translated into technology-specific terms. This function alleviates the human operator from the need to deal with technical details. High-level objectives may include definition of new services, user classes (gold, silver, bronze) and their QoS levels etc. In a subsequent step, the high level objectives are forwarded to the Policy Derivation & Management (PDM) function.

At the same time, the Human to Network function provides means for the human operator to receive information about the functioning of the network, services and the UMF system.

In order to achieve these goals, the Human to Network Interface function has been decomposed into a set of operations:

- Service Definition, which allows the specification of operator's parameters: type of service, service level agreement, network technologies, user classes, available levels of availability, reliability, speed and security, processes which will run upon a considered topology, high level goals etc.
- High Level Objectives Definition, which allows the definition of objectives for a given NEM or group of NEMs, service or group of services, or even the UMF components, including the composition of objectives and their activation
- Supervision, which provides the information about the status of network elements, services, and NEMs. This includes prioritized alerts, network monitoring information, actions implemented by NEMs and the information about the available NEMs, possible conflicts between them and the degree of fulfilment of the high level objectives previously defined by the human network operator.

In UniverSelf, the Human to Network function has been implemented in Java, with a Graphical User Interface to allow the interactivity with the human operator.

### Policy Derivation and Management function

Policy Derivation and Management (PDM) function is in charge of: (i) providing storage for the policies and facilitating the management of the Policy Repository, (ii) checking whether the different policies have conflicts and resolving them, (iii) translating the policies to lower level policies, (iv) analysing if the network, in its current status, can accommodate the request, and optimizing it otherwise, and (v) evaluating if the enforcement of the network actions fulfils the defined high level objectives. PDM defines operations for the above listed functionalities.

PDM follows the Policy Continuum approach [17], composed of a set of levels. Briefly, the purpose of the Policy Continuum is to provide a semantic linkage between different types of policies that exist at different

abstraction levels. Each of the levels is optimized for a different type of user that needs and/or uses different information. For example, a business user may need SLA information, and is not aware of network mechanisms needed for delivering the defined QoS, just that the network is in fact delivering the right type of QoS to the right people. Conversely, network operators deal with the set of commands (e.g. CLI commands) that will force the network elements (e.g. routers) to deliver the QoS defined at the business level. This is a completely different representation of the same policy. UniverSelf approach considers three different levels: Business Level, Service Level and NEM Level.

UniverSelf Governance approach strongly relies in semantic techniques for the management of policies, which allow automatic reasoning procedures on a constructed ontology. Figure 5 shows a subset of the UMF ontology, representing the Policy concepts.

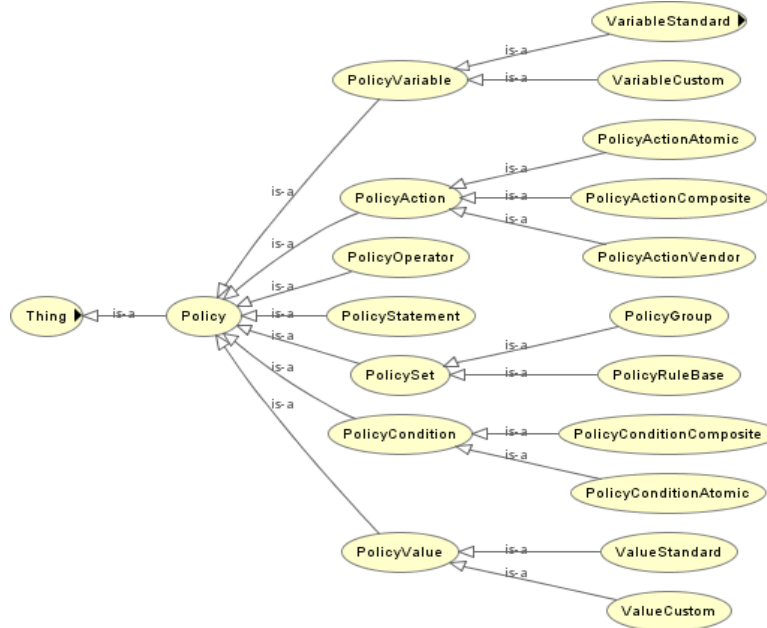


Figure 5: Ontology representing the Policy concept

The proposed policy translation methodology relies on ontology-based policy refinement approaches and it is in line with the translation methodologies studied in [18] and [19]. The policy management approach makes use of OWL/SWRL for the representation, translation and reasoning of policies.

Once the policy translation process has built a new policy in one of the levels of the policy continuum, this policy is checked for possible conflicts with the other policies in the same level of the continuum. A conflict is defined to occur when two policies can be triggered and satisfied at the same time, but their actions contradict. That implies that the event clauses and the condition clauses of two policies evaluate to TRUE at the same time, but the actions lead to contradictory states. The adopted ontology-based conflict detection and resolution mechanism [1] is a three-step process, where each step performs an analysis of the three clauses (events, conditions, actions) and filters the set of policies that could lead to a policy conflict.

Once the policy translation process reaches the low level of the continuum, the candidate NEMs to receive the new policies must be selected. This selection is made by the Check Feasibility & Optimize (CFO) [1] mechanism. CFO requires a variety of input information, related to the features of each NEM, the nature of the request that has been received, as well as the conditions in the network environment. CFO's outlines a feasible and optimized way in which the available NEMs and the corresponding network infrastructure, can handle the request. More specifically, the CFO mechanism shares the request into appropriate pieces and allocates each one of them to a selected NEM. Then the low level policies are enforced onto each of the selected NEMs through the Enforcement function of the Governance block. CFO takes into account a number of parameters as input for the decision making process, such as the supported network and service technologies, the geographical location of the network infrastructure, the supported QoS, the nature of the problem, the NEM's response time, the trust index of a NEM, the approximation to the optimal solution or energy consumption.

In summary, the deployment of the Governance core block showcases how UMF can effectively govern network and services through the governance of NEMs. Starting from high-level business objectives expressed in a close-to-natural language, GOV generates conflict-free low level policies and enforces them onto the NEMs.

#### 2.2.1.2 Coordination

The large-scale deployment of NEMs necessitates their coordinated operation and interworking, which will enable the autonomous management of complex networks and simultaneously the optimization of their performance. To this effect, proper coordination mechanisms were designated in D2.4, which achieve conflict free NEM operation, ensure stability, convergence and trustworthy NEM interworking.

The specifications of the UMF Coordination core block [1] identify three main functions:

- Orchestration
- Conflict identification
- Optimization and Conflict Avoidance

##### Orchestration

Orchestration function addresses and arranges the order of the execution sequence of NEMs and maintains the proper workflow in a way that is needed to resolve inter-NEM dependencies, in the framework of operator policies/scenario, corresponding input/output and timing relationships.

##### Conflict identification

Conflict identification function identifies potential conflicting situations that may arise due to the concurrent operation of NEMs and handles the respective NEMs accordingly, altering their default behaviour and/or constraining their actions. Specifically, three types of situations are regarded as giving rise to potential conflicts:

- parameter value conflict, where multiple NEMs have control over the same exactly network parameter,
- metric value conflict, where a metric used as input from one NEM is affected by modifications by other NEMs, and
- loops, when a NEM A uses inputs by a NEM B, but the output of NEM A “feeds” NEM B forming a loop; either explicitly/directly or implicitly through a cascade of other NEMs leading eventually back to NEM B.

Conflict identification function realizes static conflict identification, namely identification of potential conflict situations before they can actually occur (proactively), based on the NEMs behavioural specification; and dynamic conflict identification, which identifies conflict situations during runtime through the use of monitoring procedures. Static conflict identification is based on a priori knowledge of conflicting parameters, captured by the UMF system, and is triggered mostly upon configuration changes in the network or the UMF (e.g. the assignment of a mandate to a NEM). Knowledge engineering techniques, such as ontology, logic programming, and reasoning languages, are utilized in order to empower static conflict identification and minimize the required a priori given knowledge. Dynamic conflict identification regularly collects, through monitoring, information related to the derivatives of Key Performance Indicators (KPIs) as a function of the NEM control parameters.

##### Optimization and Conflict Avoidance

Optimization and Conflict Avoidance function guides the re-computation of the resource allocation to the NEMs in a way that optimizes the global system’s utility, capturing even the end-to-end optimization of different segments. Furthermore, Optimization and Conflict Avoidance function detects and avoids conflicts between NEMs, by grouping conflicts and assigning feasible mechanisms to handle them, taking into account the available optimization and conflict avoidance mechanisms and the dependencies between NEMs, as instructed by the Orchestration constraints.

Four categories of core mechanisms for managing conflicts and concurrent operations of NEMs have been defined:

- hierarchical optimization, which deals with NEMs that operate at different time scales (hierarchical optimization method, random token method, and self-orchestration through Utility Predicates method),
- separation in time strategies, which aims at having only one NEM enforcing a network parameter change at a time,
- centralized multi-objective optimization, which deal with NEMs that operate at the same time scale and tries to optimize a global/weighted utility function that combines the utility functions of all considered NEMs, and
- synchronous control theory (and its asynchronous generalization, which deal with NEMs (processes/functionalities) that operate at the same time scale and are synchronized, aims at having NEMs jointly optimize towards their individual objectives, but taking into account the influence of the other NEMs.

Furthermore, multi-priority event driven separation in time method can be used, which avoids possible conflicting behaviour of NEMs issues at design time by setting up appropriate operation requirements and validate that NEMs operation is not violating these requirements.

### 2.2.1.3 Knowledge

In this subsection, we discuss KNOW block deployment issues, including aspects related to the distributed KNOW architecture and the structure of the different KNOW nodes.

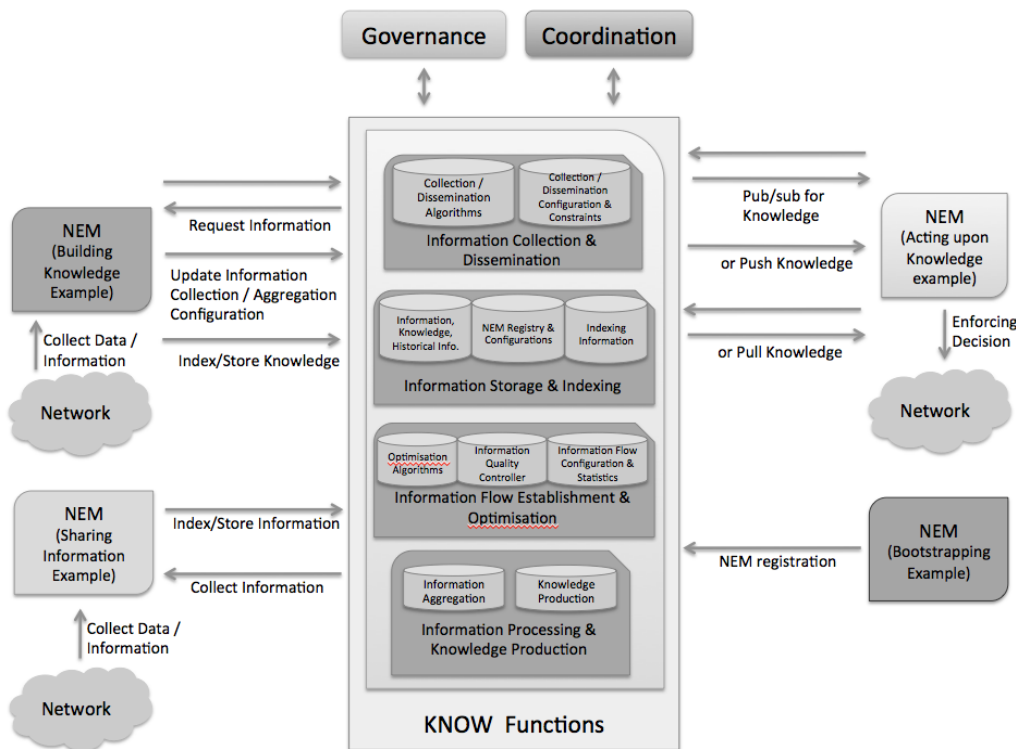


Figure 6: Overview of the Knowledge Block

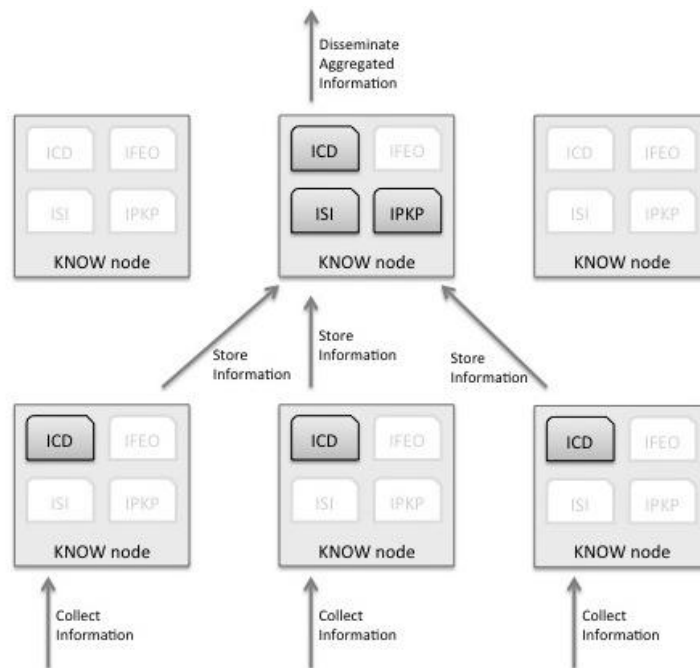
In Figure 6, we show a conceptual KNOW architecture. In practice, the different KNOW functions can be packaged at KNOW distributed nodes, in different combinations. We define as KNOW node a distributed node that hosts at least one KNOW function being able to communicate with at least one more KNOW function deployed at a different KNOW node. For example, the Information Collection & Dissemination (ICD) function can be deployed at nodes close to the information collection points (e.g., the physical infrastructure that is



monitored). The nodes' locations and combinations of KNOW functions per node should follow the global and local optimization objectives.

Since there are many instances of KNOW functions, a NEM or a core function interacting with KNOW should be able to discover the more suitable KNOW node to communicate with (e.g., the closest). Furthermore, the information/knowledge flow should cross the most appropriate KNOW nodes in order to meet the performance objectives of the particular information flow & the system.

In Figure 7, we show an example of six KNOW nodes that perform an information manipulation process (i.e., collect, store, aggregate and disseminate information). Not all available nodes are needed for this process. For the same reason, two of the KNOW nodes do not have any of their KNOW functions enabled. The bottom three nodes collect information from the monitored infrastructure. So, they have their ICD functions enabled. The collected information is being passed to a single node that stores the collected information (e.g., through the ISI function), aggregates it (e.g., through the IPKP function) and passes the aggregated information to the next node (e.g., through the ICD function).



**Figure 7: Example of modularized KNOW functions**

From Figure 7, we can identify a three tier architecture for KNOW functions. At the first tier, there are information providers interacting with KNOW nodes who at least provide one ICD module. ICD collects and maintains the information collected by the information providers, and provides it to the other exploiters on their demand. These exploiters include other KNOW modules and nodes (e.g. Information Processing & Knowledge Production functions - IPKPs), other UMF core blocks (e.g. GOV) or other NEMs. The mechanisms for collecting information from the information providers are detailed in the relevant sections of D2.4.

At the second tier, each node is recognized by a few modules and functionalities, working together to realise a certain functionality. Each module within a node can be considered as a module for realising a KNOW functionality. At this tier, each node is identified by a number of modules each providing a number of services.

At the third tier, there is an overlay network of KNOW nodes interacting with each other to provide knowledge to the other UMF core functions as well as NEMs. The structure of this distributed set of KNOW node is maintained within this tier. Proper mechanisms for maintaining this distributed structure should be supported to avoid inconsistencies, conflicts and other important issues in maintaining distributed knowledge. A few proposals as solutions for defining the architecture for second and third tier are as follows.

#### **Architecture of KNOW nodes**

The proposal for specifying a KNOW node is based on the Universal Plug and Play (UPnP). UPnP provides specification of a service discovery and provision suit. This protocol relies on web-based technologies such as HTTP and HTTPU and HTTPMU for 1) discovery, 2) obtaining descriptive information of the services, 3) control



of services by the clients, 4) notification of the clients (by the servers) of changes in status of the offered service and 5) graphically presentation of the services and their access means.

UPnP components generally consist of devices, control points and services. An *UPnP device* is a container of services and nested devices. Different categories of UPnP devices consist of different services and embedded devices. All device information, services and properties are described in an XML device description document.

*Service* is the smallest unit of control in an UPnP network. A service introduces actions and models its state with the state variable. This information is also part of an XML service description. A pointer to these service descriptions is included in the device description document. Devices may contain multiple services.

*Control Points* in an UPnP network provides discovery and control of other devices. After discovering a service, control point is capable of 1) retrieving device description and getting a related services list, 2) retrieving service descriptions for interesting services 3) invoking actions to control services and 4) subscribing to service's event source. Whenever the state of the service changes, an event server will send an event to the control point.

Based on the UPnP device specifications, the architecture of a KNOW node in UniverSelf is provided in Figure 8. Two KNOW nodes are specified as UPnP devices, and each include embedded devices such as ICDs and IPKPs. The embedded devices offer different types of services, such as Information Dissemination and Information Collection, all specified as UPnP services. To establish direct interactions between two KNOW nodes, without involvement of any other entities, each KNOW node can include a Control Point. Control points are for obtaining information from the services provided by the devices, and this can be done by search/response mechanism (PULL) or advertisement made by the devices (PUSH).

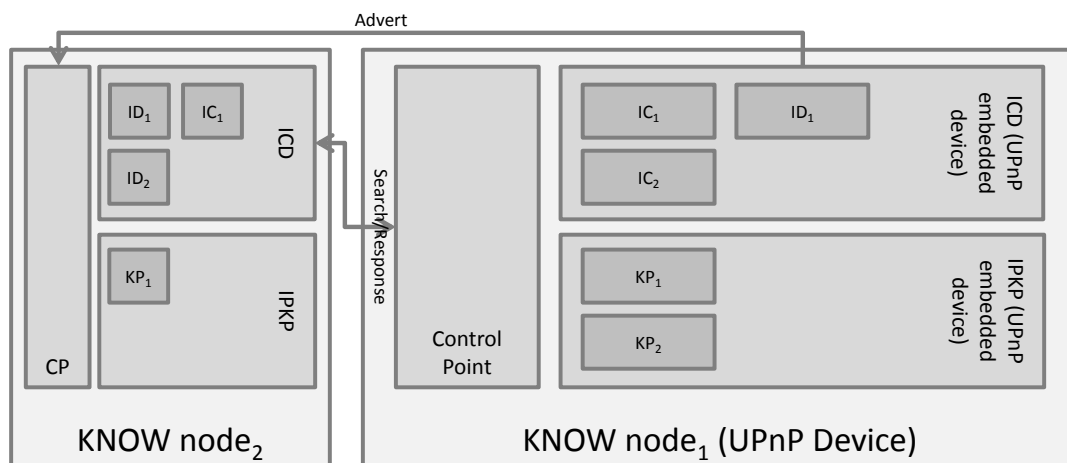


Figure 8: KNOW node architecture based on UPnP

UPnP relies on different protocols, including TCP/IP, HTTP/HTTTPU/HTTPMU, Simple Service Discovery Protocol (SSDP), Generic Event Notification Architecture (GENA) and Simple Object Access Protocol (SOAP) and XML for describing services.

Discovery stage employs SSDP for searching different services by the Control Points or advertising the services by the Devices. After getting the information about the location of the Services, Control Points get the Description of the services, which include the URL to other mechanisms, such as Controlling by SOAP, Eventing by GENA and Presentation.

**KNOW distributed architecture optimization issues (i.e., decide the number + location of KNOW nodes and combination of KNOW functions per node).**

At Tier 3, different KNOW nodes form a network within an enterprise network. Since these nodes are spread over a wide area network, there are concerns about the locality of the nodes, how nodes communicate with each other, how the hierarchy between the nodes are defined, and how the communication with tier 2 and 1 can be established. For setting up a network of KNOW nodes at Tier 3, we propose a number of solutions.

**Employing Service Location Protocol (SLP) at tier 3**

The first solution is based on employing Service Location Protocol (SLP) for setting up KNOW nodes which include Information Storage and Indexing (ISI) modules. Service Location Protocol (SLP) is an IETF recommendation; it defines a standard for service discovery and registration on the Internet. It therefore uses extensions to the Uniform Resource Locators (URLs) in order to define services and their attributes alongside of body texts transmitted via HTTP.

Active nodes in the SLP are agents. Each of the agents acts as an end partner in the process of service discovery. The defined agents in SLP are 1) Service Agents (SA), active nodes providing specifications of the services to the network, 2) User Agents (UA), active nodes making queries to find out specifications of the available services. The queries are made to either Service Agents or Directory Agents and 3) Directory Agents (DA), active nodes that register the specifications of the services and hold a copy of this information. By formation of the network, there are mechanisms that SAs register their offered services with the DAs. As mentioned before, UAs make queries to the DAs, but they still can also make them to the SAs. An important concept in SLP is scope; Scopes in SLP are parameters used for logically partitioning the SLP network. This concept helps in limiting the extent of the smaller networks, sharing the same context, which is very important in context aware service discovery.

There is a proposal for enhancing SLP with a multi-DA environment, and is called *mesh Service Location Protocol* (mSLP). This proposal introduces interactions between the DAs in a multi-DA system, which can be considered as an overlay between the DAs. The protocol between a pair of DAs called *peering*, manages the establishment, maintenance and tearing down of peer relationships. SAs communicate with only one DA, who forwards the registration of the SA to other peer DAs. This solution is applicable to clustered networks and in UniverSelf for partitioning the network bases on Context. Figure 9 depicts an example of an mSLP network. The figure depicts interactions between three *Mesh Directory Agents* (MDAs) through mSLP peering protocol, and between individual MDAs, *Mesh Service Agents* (MSAs) and *User Agents* (UA). MSA registers its service with only one MDA. In this figure, MDA is mapped to KNOW node ISI, MSA to other KNOW modules and UA to any exploiter of Knowledge.

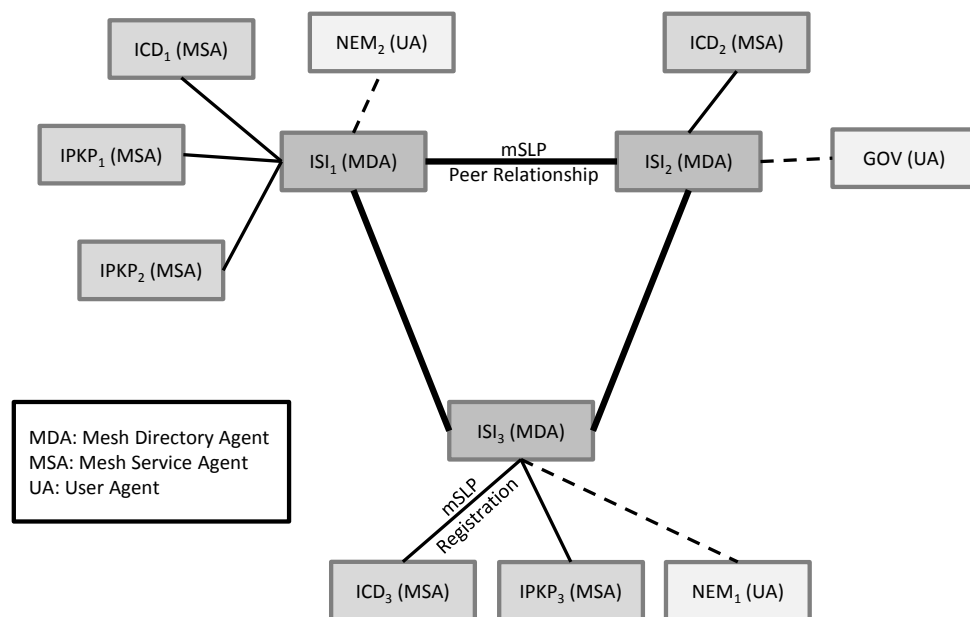
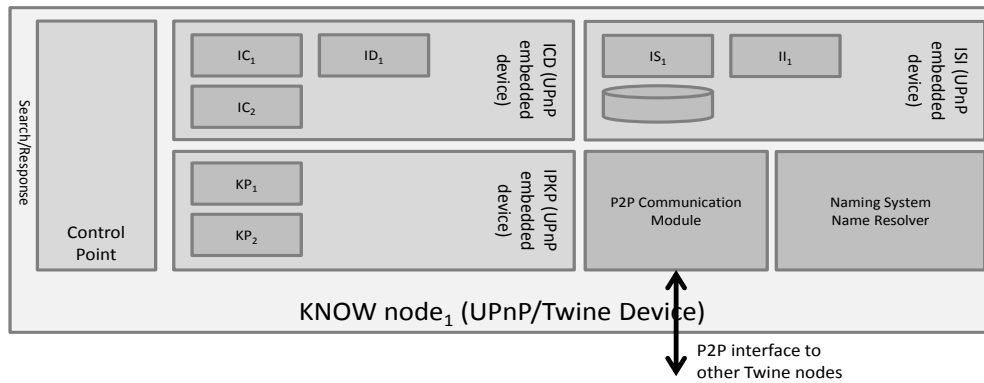


Figure 9: mSLP example network mapped to KNOW modules

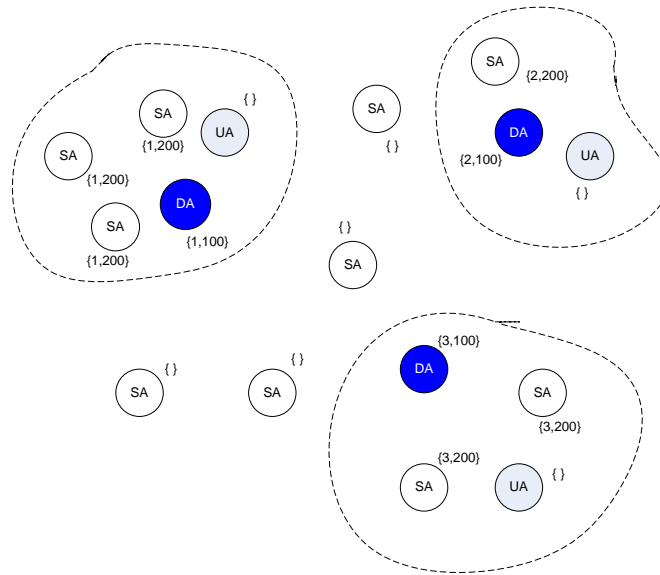
The same approach can be taken for peer-to-peer connectivity between pair of ISIs by employing other peer to peer service discovery protocols such as Jini, Jxta and INS-Twine. This solution for the combination of UPnP at tier 2 and Twine at tier 3 was previously applied in ICT-MAGNET project.

Figure 10 depicts the architecture of empowered KNOW nodes who act as cluster lead and able to communicate with other lead nodes in a peer-to-peer manner. The solution based on INS/Twine relies on a new naming system and name resolver (INS).



**Figure 10: INS/Twine solution for P2P connectivity between empowered KNOW nodes- KNOW nodes' discovery issues (i.e., how to discover the most appropriate nodes)**

The structure proposed for tier 3 relies on grouping the KNOW functionalities. Here the concept of Context Aware Discovery is applied, i.e. the nodes sharing the same or nearby context will be grouped and registered with the corresponding ISI. Figure 11 depicts the concept of clustering the network into smaller clusters based on the context. In the extreme case, the context of each KNOW node or Knowledge providers is identified by a number. The context of each node is communicated over the discovery messages. This is done in SLP based on the *scope* concept used in SLP. *Scopes* in SLP define virtual border of clusters, and are efficient in supporting context awareness in discovery.



**Figure 11: Clustered mSLP based on the context**

Efficiency of discovery protocols is being evaluated by measuring success rate in discovery and saving costs. Accuracy of discovery based on context is of very high importance. The results of location aware discovery present a high level of improvement in success rate and cost efficiency. Figure 12 depicts that success rate in discovering services in Scoped case is higher and used energy is lower than the normal case.

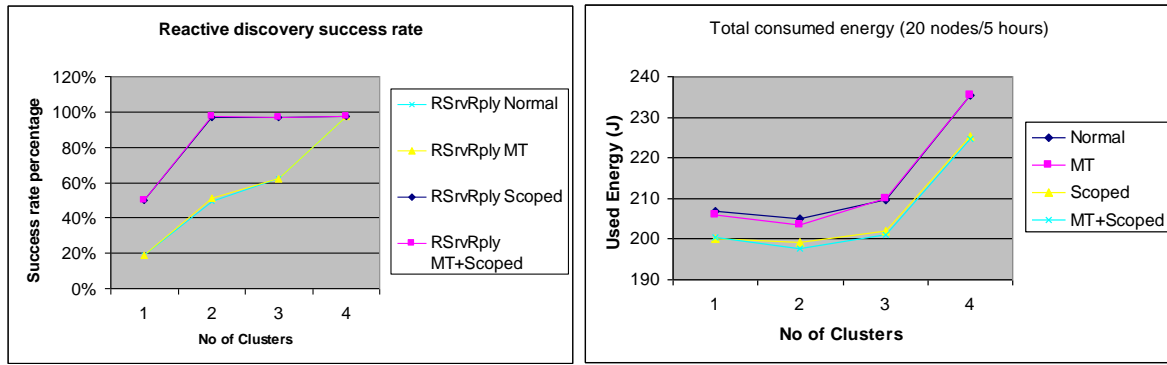


Figure 12: Scope-based cluster aware discovery success rate and energy efficiency

The current work in UniverSelf is on modelling the context based on defining the context of the nodes as different multi-dimensional vectors and use of similarity function to evaluate matching of the services (entities) to the context.

#### Internal KNOW interfaces (i.e., for communication between the KNOW nodes)

The interfaces between the KNOW nodes should be based on the definition of the parameters passed between the modules formatted into either tier 2 or tier 3 protocols (e.g. UPnP or SLP).

In defining the parameters to be passed during UPnP discovery and description, we use the following values for KNOW block components. Discovery, the first step in UPnP networking, specifies when a device is added to the network, that device advertises its services to control points on the network. Also, a control point searches for devices of interest on the network. The fundamental exchange in both cases is a discovery message containing parameters about the device or one of its services, e.g., its type, universally unique identifier, a pointer to more detailed information and optionally parameters that identify the current state of the device. The port assigned for advertisements and search messages is port 1900 or on the port specified by the optional SEARCHPORT.UPNP.ORG header field. For SLP and mSLP, port 427 on UDP and TCP are used for SLP messages. Further details on the parameters passed between different SLP nodes can be found in the next sub section.

#### Communication between the KNOW nodes (e.g., protocol issues)

Communication between KNOW nodes are either based on tier 2 protocol (i.e. UPnP) or tier 3 protocol (e.g. SLP)

To specify the UPnP discovery protocol for KNOW nodes, we should use an UPnP Basic Device as the container KNOW node. This node is the root device for other KNOW components. The search message in UPnP can be used, however the device and service fields should be defined.

## 2.2.2 NEM Deployment

This section introduces the different NEMs that are deployed on the three testbeds to build the three scenarios. All the deployed NEMs are governed, are coordinated and share knowledge through the mechanisms of the UMF core blocks.

### 2.2.2.1 SDN-based core and WLAN access testbed

We introduce 4 different NEMs to orchestrate the SDN-based Core and WLAN Access dataplane. Two NEMs are instantiated in the access segment, namely, WLAN Infrastructure Management NEM and Wireless access Load Balancing NEM. For the core segment, Virtual Infrastructure Management NEM and Core Load Balancing NEM are instantiated as well. Figure 13 visualizes UMF and NEMs deployment.

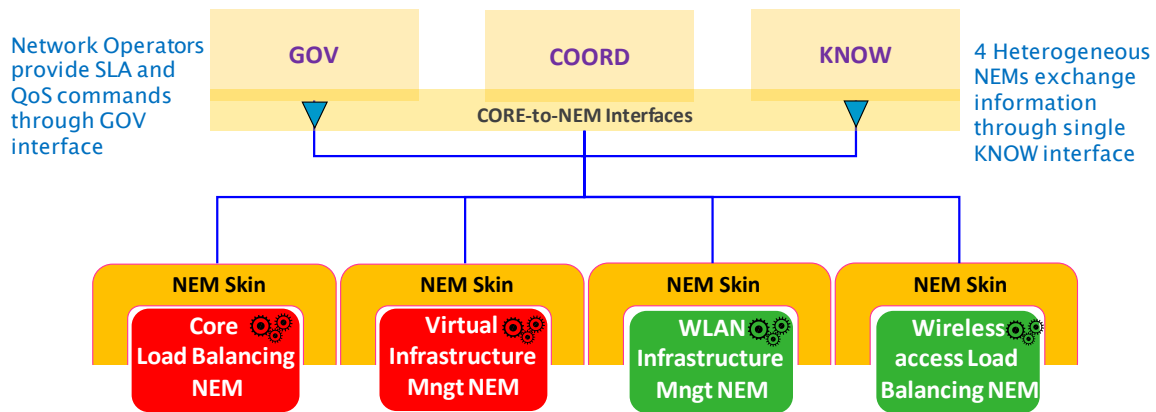


Figure 13: SDN-based Core and WLAN Access Data plane - UMF and NEMs.

In a higher technical granularity, NEMs' functionality is presented above:

1. WLAN Infrastructure Management NEM
  - Monitoring of the topology (#APs, #MTs), providing network parameters (CU, Available BW, RSS, Noise, Link Quality, Tx),
  - Network Elements' Reconfiguration dictated by Wireless access load balancing NEM through KNOW.
2. Wireless access Load Balancing NEM
  - Derivation of complex load parameters (i.e. Device Status based on # of users, CU, Available BW, and PL) for load balancing,
  - UMF KNOW update with handover decisions to be executed by WLAN Infrastructure Management NEM.
3. Virtual Infrastructure Management NEM
  - Dynamic instantiation and deployment of SDN topology,
  - Reconfiguration of virtual routers (Ingress/Egress path establishment and adaptation, forwarding, re-routing).
4. Core Load Balancing NEM
  - Monitor service flows,
  - UMF KNOW update with link activation/deactivation commands to be executed by Virtual Infrastructure Management NEM.

The data plane together with the introduced UMF capacities is presented in Figure 14.

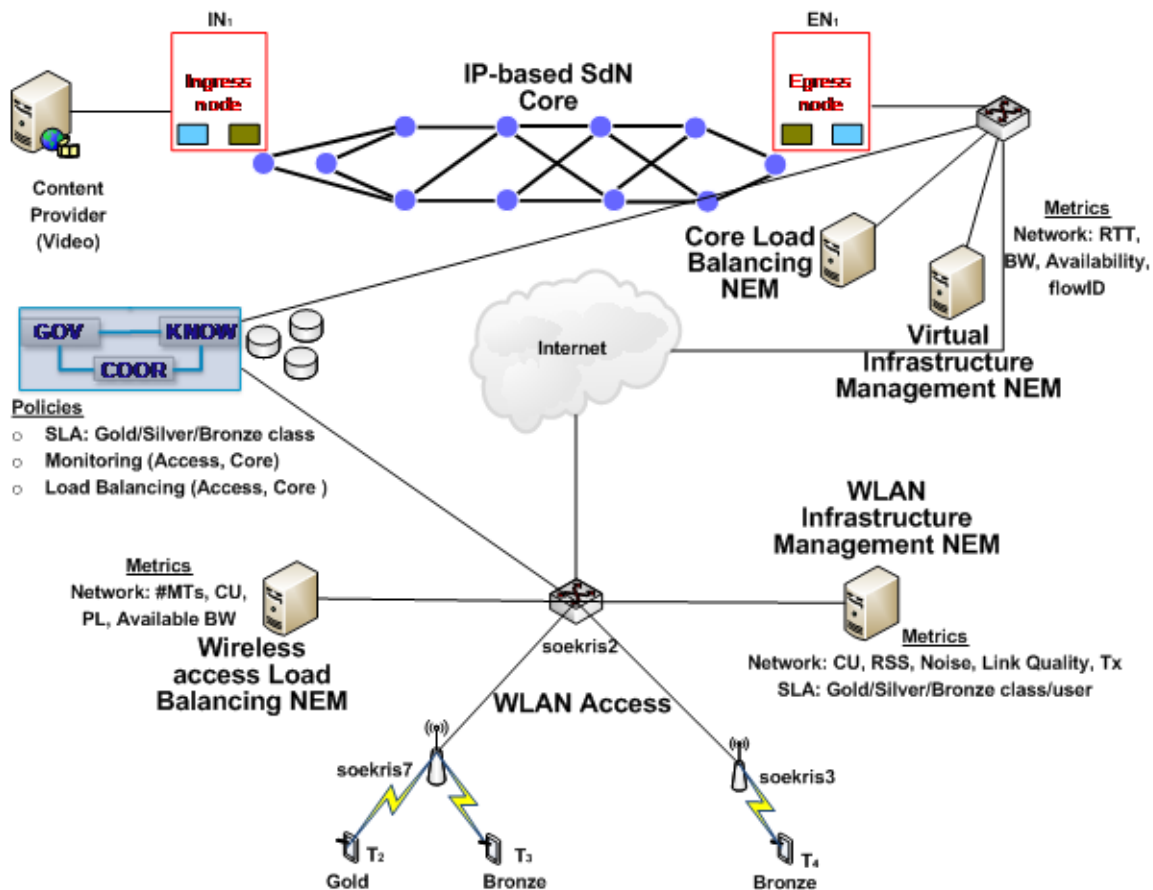


Figure 14: SDN-based Core and WLAN Access Data plane – UMF enhanced dataplane.

#### 2.2.2.2 SON Coordination LTE testbed

In the SON Coordination LTE testbed, two SON NEMs are deployed, as shown in Figure 15. Both use the REST interface provided by the simulator for the monitoring and the setting of configuration parameters as well as for the monitoring of KPIs from NEMs. The first NEM is responsible for adjusting the pilot powers of the Femtos in order to affect their coverage, while the second SON NEM optimises the backhaul ratio, i.e. the ratio of radio resources allocated to backhaul links (Femto-to-Macro) over the radio resources allocated to user links (Users-to-Macro). The same interface is used to manually change the Poisson arrival rates for the sake of demonstration, during the progress of demos (see Section 2.3.2).

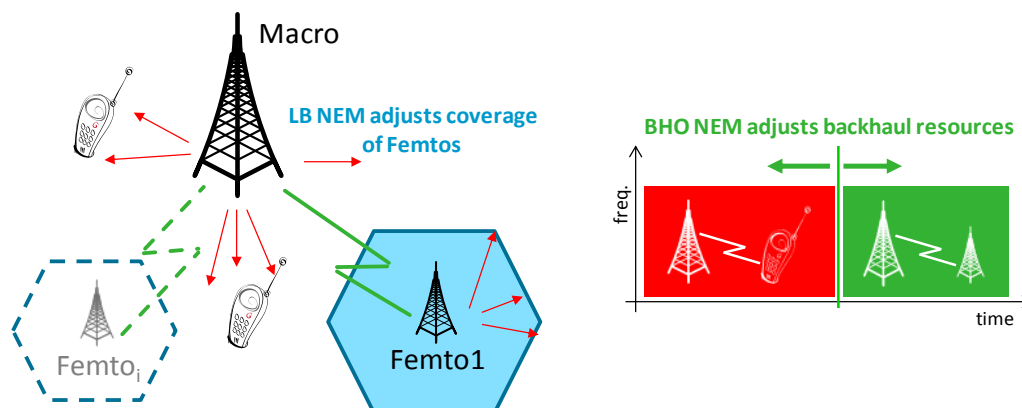


Figure 15: LTE advanced network with two SON functions

### 2.2.2.3 Very Lightweight Software Driven Networks and Services testbed

The full VLSP software stack consists then of three main layers:

- (i) the Lightweight Network Hypervisor, already described in Section 2.1.3
- (ii) the Virtual Infrastructure Management layers, and
- (iii) the Knowledge core block.

An architectural overview of the software stack is shown in Figure 16. The VLSP test-bed software consists of over 700 java classes and more than 100 k-lines of code. In our experiments with VLSP, we have executed over 100 virtual routers on each of 12 dedicated physical servers.

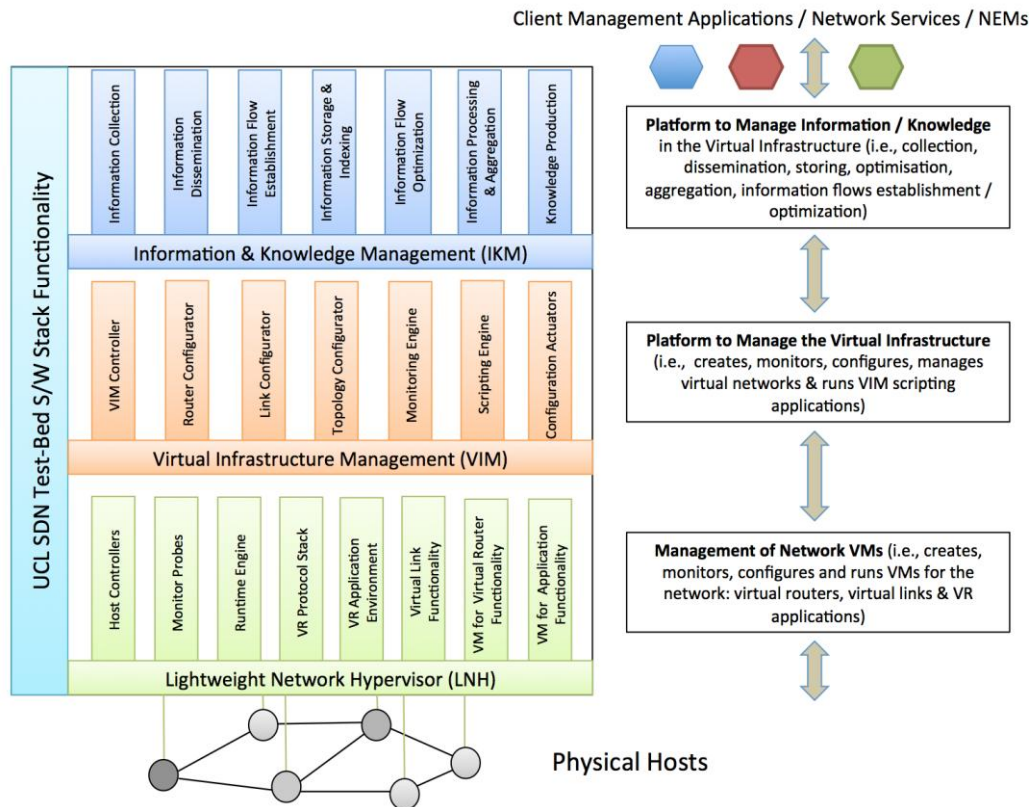


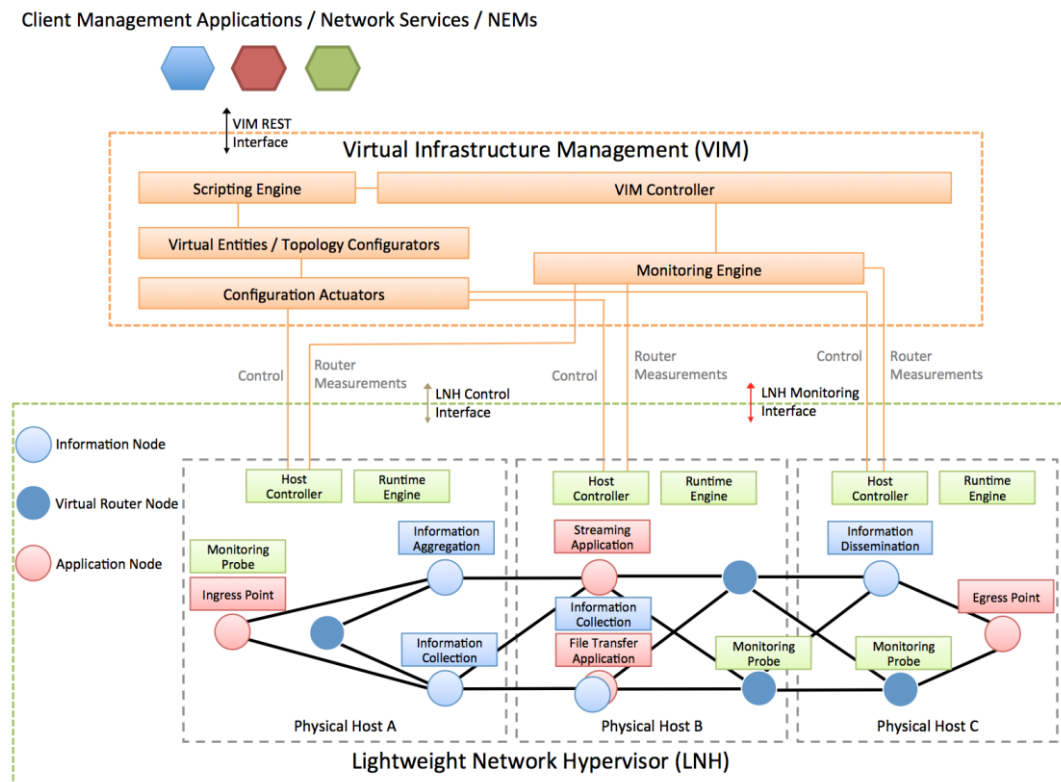
Figure 16: Overview of VLSP Test-bed Software Stack

The virtual Infrastructure Management layer consists of two NEMs: The Virtual Infrastructure Management (VIM) NEM and the Placement Optimization (PO) NEM.

#### The Virtual Infrastructure Management (VIM) NEM

In this section we describe the Virtual Infrastructure Management NEM, highlighting its purpose and its architecture. The Virtual Infrastructure Management (VIM) NEM is responsible for the management and lifecycle of the virtualized elements that will be used within a network, particularly virtual network elements. As the virtual elements are not physical themselves, but exist on top of physical elements, their lifecycle and their management needs to be approached carefully to ensure continued operation and consistency. An overview of the VIM architecture on top of the LNH is shown in Figure 17.





**Figure 17: The Virtual Infrastructure Management NEM**

The virtual network elements, which exist on top of physical networks, can be setup with arbitrary topologies and with an arbitrary number of end-points. The virtual links in a virtual topology are eventually mapped onto physical links in the underlying network. A virtual link may span multiple physical links, and cross many physical routers, or it may span a single physical network link. New virtual links can be added or can be removed from a virtual network dynamically at run-time.

The virtual networks are very flexible and adaptable, and generally have few limitations, except that a virtual link cannot support more traffic and higher-data rates than the underlying physical links. Furthermore, a whole virtual network can be shutdown as needed, if the applications that use it no longer need the network. Such a shutdown frees resources from the underlying physical network.

The full management of virtual networks on physical networks requires the matching and analysis of the flow rates on the virtual links to the flow rates of the underlying physical links. It is important to ensure that the physical links are not congested with too many virtual links. Also, the allocation and mapping of virtual links must take into account the current state of the physical network and the current virtual networks. However, if a situation arises where a re-configuration is required, the virtual network management should be capable of mapping a virtual link across different physical links at run-time, but leave the virtual topology intact.

The VIM component has a seemingly simple task, but in reality the management requires continual monitoring, analysis, and adaption of the virtual elements to the physical elements. As all of these virtual elements are distributed, the management is a complex task.

The diagram in Figure 17 shows how the VIM NEM interacts with the virtual network elements that will be present in a running virtual network. All of the elements of the VIM NEM constitute a fully distributed system, whereby an element or node can reside on any host. A full virtual network can be instantiated on a single machine, for demonstration or testing purposes, or instantiated across multiple servers, in a full deployment situation.

The VIM directly controls the lifecycle of each virtual element, by collecting knowledge on the status of physical resources in order to determine where a virtual element can be created. The virtual network element will be created, managed, and shutdown by lifecycle phase of this component.



Due to the dynamic nature of virtual elements and because they can be disassociated from the physical elements they are mapped to, it is possible to do a live adaption of a virtual element from one physical host to another physical one, at run-time.

The VIM controller acts as a control point for managing the virtual elements. This block (shown in black in Figure 3) accepts all its input via the VIM REST interface from other management applications / network services or NEMs. The monitoring engine acts a collection point for the monitoring data needed to keep the management functions running. Control commands are being sent to the VIM and they are either acted upon immediately or are passed to the corresponding Host Controllers of the LNH.

The main VIM functions specification follows.

Name	Description
VIM Controller	It is the heart of the component, providing the central control of the VIM operations.
Scripting Engine	VIM can be configured using Closure scripting.
Monitoring Engine	It is the main monitoring component of the infrastructure, i.e., collecting & manipulating measurements from the monitoring probes residing at the LNH.
Virtual Entities / Topology Configurators	These functions are responsible for the configuration of virtual routers, links and topologies, supporting different levels of abstraction.
Configuration Actuators	The Virtual Entities / Topology configurators communicate with the configuration actuators which in turn enforce the configuration changes through the LNH's host controllers.

### The Placement Optimization NEM

The Virtual Networks are characterised as highly dynamic network environments, where topologies and nodes adapt rapidly to changes in user and service demands, user location and context changes, or resource constraints. In order to manage the challenging and dynamic infrastructures of virtual networks there needs to be a monitoring system which can collect and report on the behaviour of the resources, combined with a management system that can use the monitoring information to make decisions regarding network behaviour.

The Placement Optimizer NEM (PO NEM) is responsible for optimizing the placement of special management nodes within a network. Such placement is used in various network management functions, and in addition, the nodes which are placed can have various functions. For the NEM described here there are various algorithms which take into account the current topology and current traffic volumes and produce as output the nodes which have been determined to be 'special'.

An example of this, consider the application placement of nodes include monitoring collection nodes and monitoring aggregations points. Such an architecture uses Information Collection Points and Information Aggregation Points to scalably aggregate, filter, and collect data within the virtual domain.

### The Knowledge core block

The Knowledge core block (KNOW) is a critical part of the VLSP and UMF since it plays the role of information / knowledge collection, aggregation, storage/registry, knowledge production and distribution across all the functional components, management applications, network services and NEMs in the network environment (see Figure 18). It can run on top of VIM, since it is fully integrated within the virtual network, e.g., the virtual

routers have embedded information / knowledge manipulation capabilities. Furthermore, it is used by any client management application / network services, Network Empower Mechanisms (NEMs) or UMF core service. As we have shown above, KNOW is fully integrated within the SDNs but acts as a standalone component as well. In the context of UMF, KNOW supports a wide-range of network environments beyond SDNs.

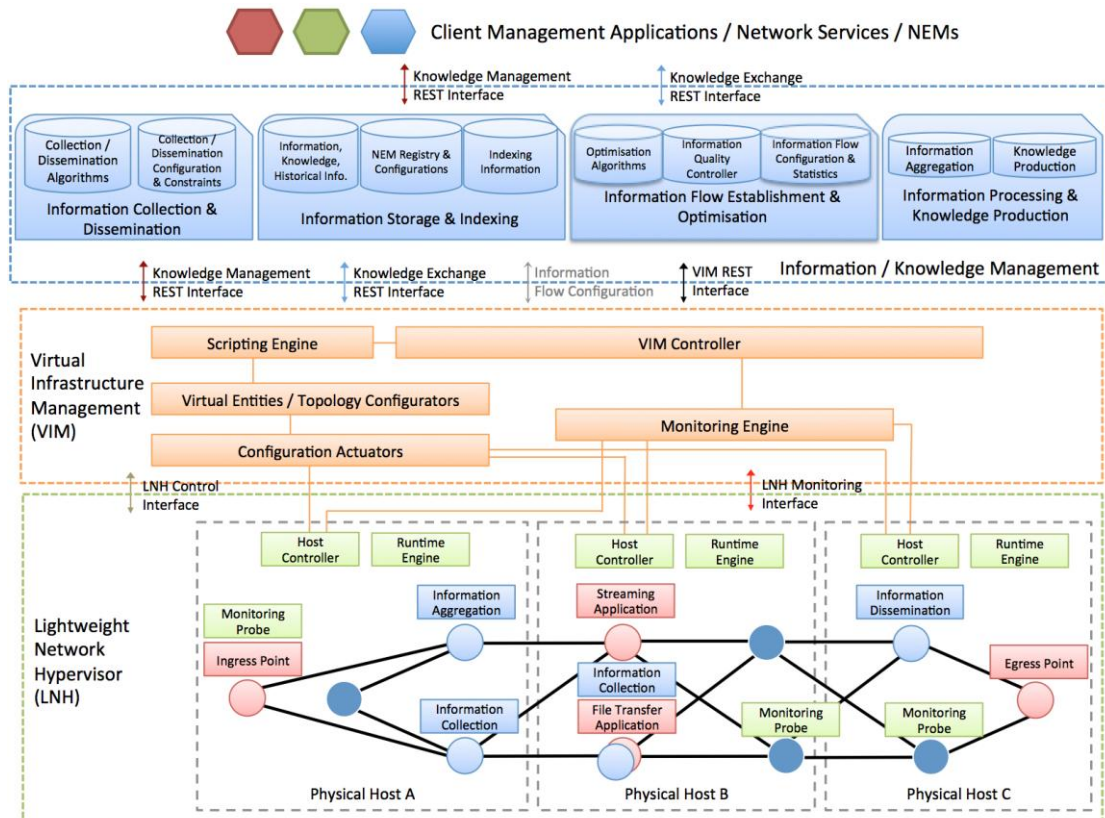


Figure 18: Knowledge core block in VLSP testbed

The main KNOW functions can be found in the following table.

Name	Description
Information Collection & Dissemination	This function is responsible of information retrieval, sharing and dissemination.
Information Storage & Indexing	This function is a logical construct representing a distributed repository for registering information-enabled entities, indexing (and optionally storing) information/knowledge.
Information Flow Establishment & Optimization	This function regulates the information flow based on the current state and the locations of the participating entities and nodes.
Information Processing & Knowledge Production	The Information Processing and Knowledge Production function is responsible for operations related to information processing (e.g., aggregation) and knowledge

	production.
--	-------------

## 2.3 Scenarios and Deployment Results

Once the testbed and the deployment aspects have been described, this section presents the three scenarios that are used as basis for the demonstration and the deployment results, in the form of storylines describing the processes, events and actions that take place in each of the testbeds.

### 2.3.1 Scenario 1- A Unified Framework for QoS and SLA-aware multi-domain self-management

In this section a more detailed storyline description will be provided as applied to the SDN-based Core and WLAN Access data plane. This scenario highlights how UMF enables the QoS management in a multi-domain autonomic network. The UMF dashboard facilitates the governance of the network, through the Ontology and Policy tabs.

The Ontology Concept Composition tab (see Figure 19) provides the Network Operator with the ability to insert its own concepts, regarding:

- Services and SLA's offered to end users
- Processes that run on top of the heterogeneous network environment
- Network Domains and Technologies that compose the heterogeneous network environment that Network Operator governs,
- Goals that operator wants to achieve through the use of UMF tool.

The screenshot shows the 'UMF Core' dashboard with the 'Ontology Concept' tab selected. The interface is organized into four columns: 'Service SLA', 'Process', 'Network Domains & Technology', and 'Goal'. Each column contains a table of existing concepts and a form to add new ones. The 'Service SLA' table has columns 'Type Service' and 'SLA'. The 'Process' table has a column 'Concept Name'. The 'Network Domains & Technology' table has columns 'Concept Name' and 'Technology'. The 'Goal' table has a column 'Goal Name'. Below each table is a form to add a new concept. At the bottom of the dashboard, there are 'Clear' and 'Insert' buttons.

Figure 19: Ontology Concept tab in the UMF dashboard

Policy Composition tab (see Figure 20) constitutes a unique interface for definition of high level business rules and objectives. These rules are composed with concepts introduced in Ontology Concept Composition. Policy Derivation and Management Module of Governance elaborates on those rules so as to generate low level, NEM specific, rules communicated through NEM skin to the appropriate NEM instances.

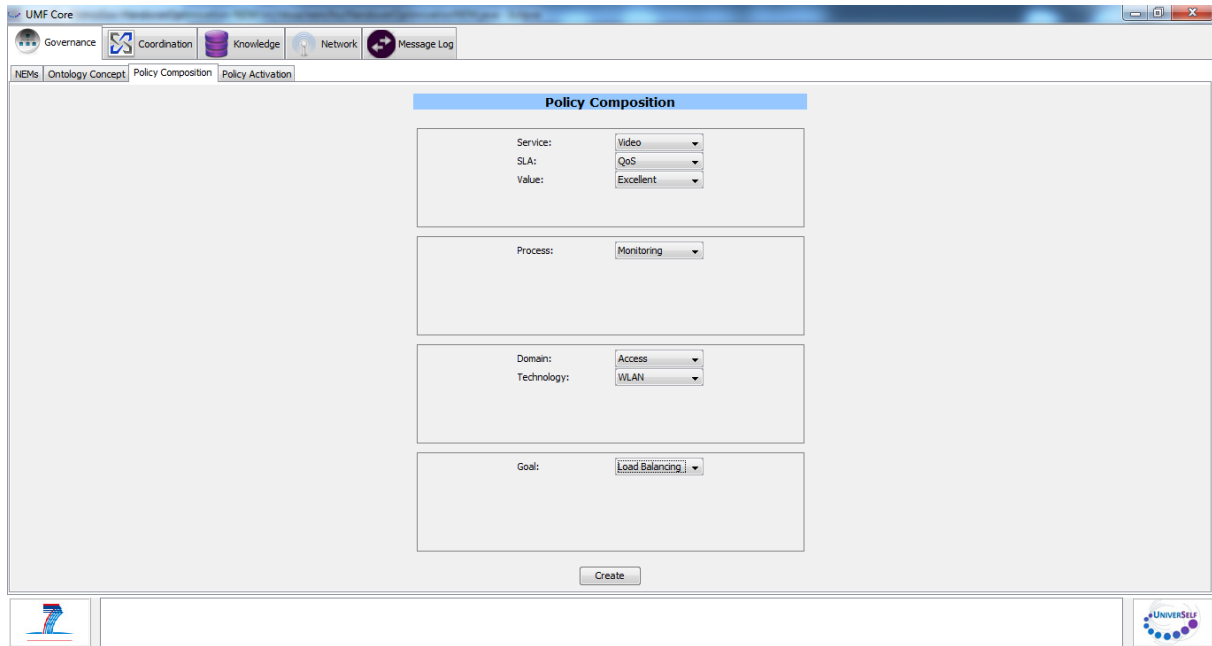


Figure 20: Policy Composition tab in the UMF dashboard

The UMF dashboard also offers a view of the topology, as shown in Figure 21:

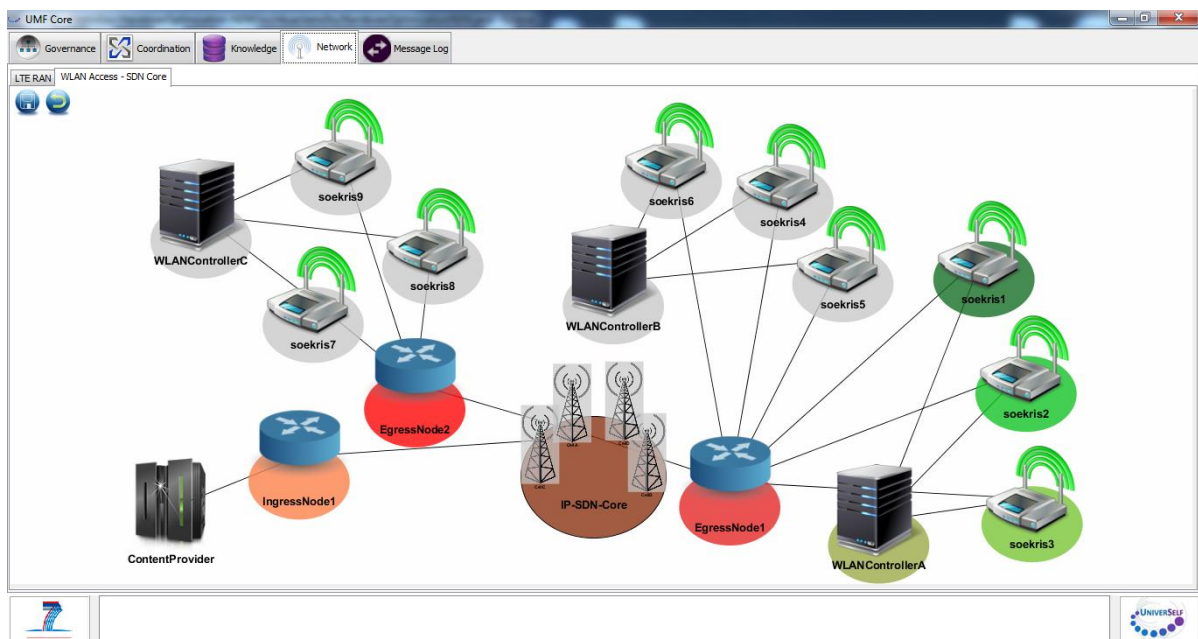




Figure 21: Topology of scenario 1 in UMF dashboard

In this scenario, initially UMF is deployed while WLAN Infrastructure Management (WIM) and Virtual Infrastructure Management (VIM) NEMs are instantiated and deployed for providing monitoring information of the access and core network respectively. No load is injected in the network, thus no need for reconfigurations is raised. Figure 22 presents access network status, as monitored by WIM NEM.

WLAN Infrastructure Monitoring

ID	CHANNEL	TX POWER	CELL UTILIZATION	ASSOCIATED TERMINALS	TIME STAMP
soekris3	1	15.0	0.0270987037037037	2	2013-07-02 10:56:37
soekris7	11	15.0	0.0285262037037037	1	2013-07-02 10:56:37

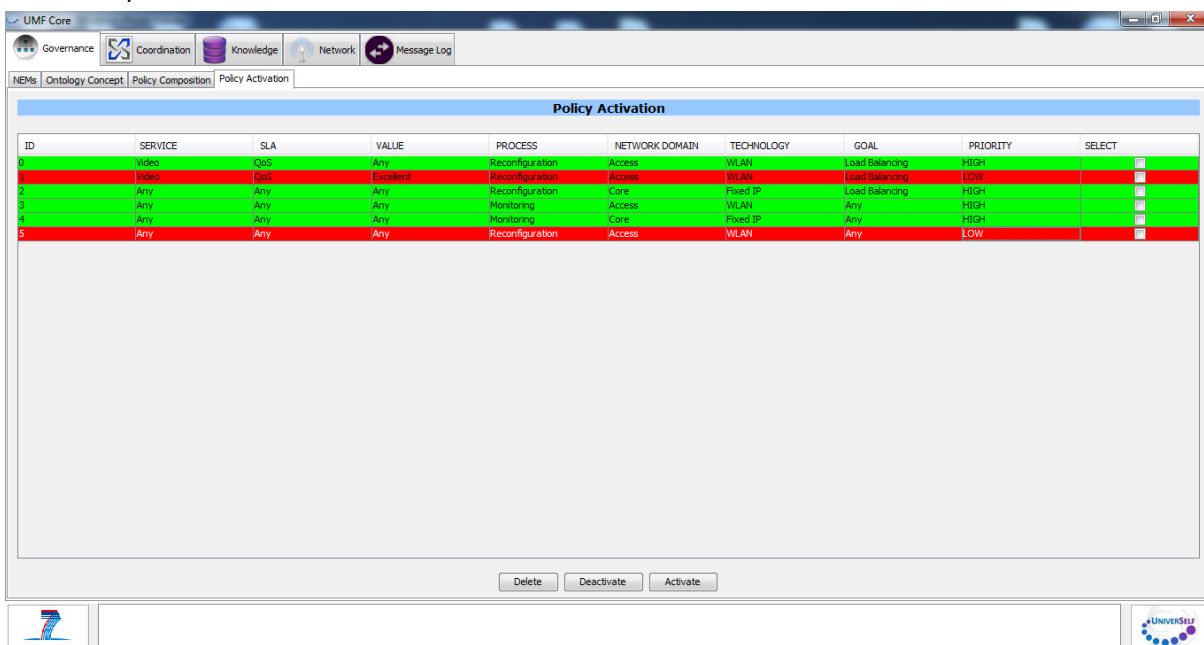
  

ID	SLA	ESSID	CELLID	NOISE	BITRATE	ADDRESS	TIME STAMP
terminal2	GOLD	soekris7	00:0c:42:61:41:a8	-256.0	590.0	10.10.1.36	2013-07-02 10:56:33
terminal4	BRONZE	soekris3	00:0c:42:61:41:a8	-256.0	1754340.0	10.10.1.67	2013-07-02 10:56:17
terminal3	BRONZE	soekris7	00:1b:b1:05:3e:35	-256.0	1643015.0	10.10.1.34	2013-07-02 10:56:26

Figure 22: Scenario 1 – Initial access network status – No Load.

As depicted in Figure 22, three users exist with different SLAs (two «GOLD» and one «BRONZE»). Terminal2 (Gold user SLA) and Terminal3 (Bronze User SLA) are associated to soekris7 and Terminal4 (Bronze user SLA) is associated to soekris3.

Injecting some load in the network, by streaming a video service to the mobile terminals, the load is increased and network operator enables policies for load balancing in the access and core network. Those policies are dictating the deployment of 2 NEMs, namely, Wireless access Load Balancing (WLB) NEM and Core Load Balancing NEM. The former NEM produces a combined metric called Device Status by gathering network measurements from UMF KNOW, as produced by WIM NEM. Once the device status exceeds a predefined level, indicated in the Governance policies, then a reconfiguration action (i.e. a mobile terminal reallocation) is executed by WIM NEM.



UMF Core

Governance | Coordination | Knowledge | Network | Message Log

NEMs | Ontology Concept | Policy Composition | Policy Activation

**Policy Activation**

ID	SERVICE	SLA	VALUE	PROCESS	NETWORK DOMAIN	TECHNOLOGY	GOAL	PRIORITY	SELECT
0	Video	QoS	Any	Reconfiguration	Access	WLAN	Load Balancing	HIGH	<input checked="" type="checkbox"/>
1	Video	QoS	Excellent	Reconfiguration	Access	WLAN	Load Balancing	LOW	<input checked="" type="checkbox"/>
2	Any	Any	Any	Reconfiguration	Core	Fixed IP	Load Balancing	HIGH	<input checked="" type="checkbox"/>
3	Any	Any	Any	Monitoring	Access	WLAN	Any	HIGH	<input checked="" type="checkbox"/>
4	Any	Any	Any	Monitoring	Core	Fixed IP	Any	HIGH	<input checked="" type="checkbox"/>
5	Any	Any	Any	Reconfiguration	Access	WLAN	Any	LOW	<input checked="" type="checkbox"/>


Delete Deactivate Activate

Figure 23: Scenario 1 – Policy activation for load balancing.



Figure 23 visualises UMF Governance H2N interface for policy activation. Currently, monitoring and reconfiguration policies are activated for both access and core segments, however no SLA constraints are being put to address video QoS requirements for each user class. This means that it is equally possible to execute a mobile terminal re-allocation that belongs to a GOLD class user (depending on load and signal level conditions




(week received signal strength for the candidate user). Indeed, based on Device Status metric, the GOLD user, who in Figure 22 was allocated to Soekris7 AP, has been re-allocated to the neighbouring AP, Soekris3, as shown in Figure 24.

UNIVERSELF

  
SEVENTH FRAMEWORK  
PROGRAMME

WLAN Infrastructure Monitoring

	ID	CHANNEL	TX POWER	CELL UTILIZATION	ASSOCIATED TERMINALS	TIME STAMP
	soekris3	1	15.0	0.0324987037037037	2	2013-07-02 10:56:37
	soekris7	11	15.0	0.030426203703703705	1	2013-07-02 10:56:37

	ID	SLA	ESSID	CELLID	NOISE	BITRATE	ADDRESS	TIME STAMP
	terminal2	GOLD	soekris3	00:0c:42:61:41:a8	-256.0	590.0	10.10.1.66	2013-07-02 10:56:33
	terminal4	BRONZE	soekris3	00:0c:42:61:41:a8	-256.0	1754340.0	10.10.1.67	2013-07-02 10:56:17
	terminal3	BRONZE	soekris7	00:1b:b1:05:3e:35	-256.0	16430.15.0	10.10.1.34	2013-07-02 10:56:26

**Figure 24: Scenario 1 – User re-allocation according to network metrics.**

This decision, taken in the access segment, affects also the core network in the sense that the service flow should be redirected to the neighbouring AP so as to preserve video and avoid service interruption. This flow/path reconfiguration is automatically executed by Core Load Balancing (CLB) NEM once WLB NEM specifies the handover decision. The egress node of the core segment updated the IP address of the mobile terminal. This is feasible since all the NEMs are able to communicate with each other through UMF Knowledge block. Figure 25 visualizes CLB reconfiguration results. As can be seen, there exist a service path per mobile terminal. Router-1, which is also the ingress one, stores service paths for every mobile terminal from the three ones. The same status also exists in Router-2. Then the traffic is balanced among the two service paths, and their consecutive Routers. Finally, Router-7 as the single egress router of the topology, stores paths for all terminals. In case of a handover event at the access part, the relative IP address of the mobile terminal is updated (e.g. AdaptEgress/4).

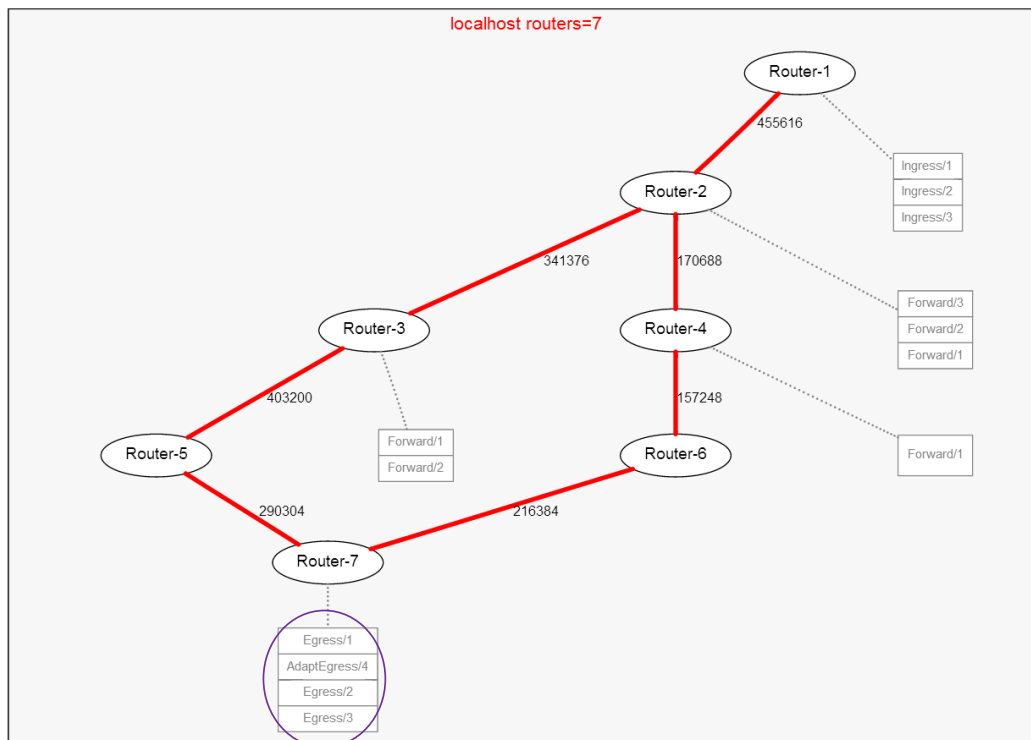


Figure 25: Scenario 1 – Egress path adaptation in the core network together with the first access network handover.

The above-taken decision is not the optimal one since within a real network environment end users are assigned with SLAs which reflect the QoS that they expect for their services from the network operator while in parallel the cost of this QoS provision. This means that SLAs should be accounted in the big figure; network operator can easily enable this option using UMF Governance H2N interface. By composing a new policy (Figure 26) which assigns user classes to end-users for a particular service, WLB NEM is able to update its decision making.

ID	SERVICE	SLA	VALUE	PROCESS	NETWORK DOMAIN	TECHNOLOGY	GOAL	PRIORITY	SELECT
1	Video	QoS	Any	Reconfiguration	Access	WLAN	Load Balancing	LOW	<input type="checkbox"/>
2	Video	QoS	Excellent	Reconfiguration	Access	WLAN	Load Balancing	HIGH	<input type="checkbox"/>
3	Any	Any	Any	Reconfiguration	Core	Fixed IP	Load Balancing	HIGH	<input type="checkbox"/>
4	Any	Any	Any	Monitoring	Access	WLAN	Any	HIGH	<input type="checkbox"/>
5	Any	Any	Any	Monitoring	Core	Fixed IP	Any	HIGH	<input type="checkbox"/>
6	Any	Any	Any	Reconfiguration	Access	WLAN	Any	LOW	<input type="checkbox"/>

Figure 26: Scenario 1 – Policy activation for SLA-aware load balancing.



In particular, upon a user re-allocation event, WLB NEM will decide about the optimal handover considering both network and SLA parameters. Therefore, in a similar high load event, the new re-allocation results are showcased in Figure 27. The updated decision performs Bronze user re-allocation respecting SLAs and improving the overall Device Status of the access network.

WLAN Infrastructure Monitoring								
		ID	CHANNEL	TX POWER	CELL UTILIZATION	ASSOCIATED TERMINALS	TIMESTAMP	
		soekris3	1	15.0	0.14031440740740742	1	2013-07-02 11:00:54	
		soekris7	11	15.0	0.15130485185185186	2	2013-07-02 11:00:54	

		ID	SLA	ESSID	CELLID	NOISE	BITRATE	ADDRESS	TIME STAMP
		terminal2	GOLD	soekris3	00:0c:42:61:41:a8	-256.0	7576978.0	10.10.1.66	2013-07-02 11:00:45
		terminal3	BRONZE	soekris7	00:1b:b1:05:3e:35	-256.0	8169498.0	10.10.1.34	2013-07-02 11:00:47
		terminal4	BRONZE	soekris7	00:1b:b1:05:3e:35	-256.0	964.0	10.10.1.36	2013-07-02 11:00:39

Figure 27: Scenario 1 – User re-allocation according to network and SLA metrics.

Besides the right decision in the access network, a respective reconfiguration takes place in the core by adapting egress service flow path as shown in Figure 28.

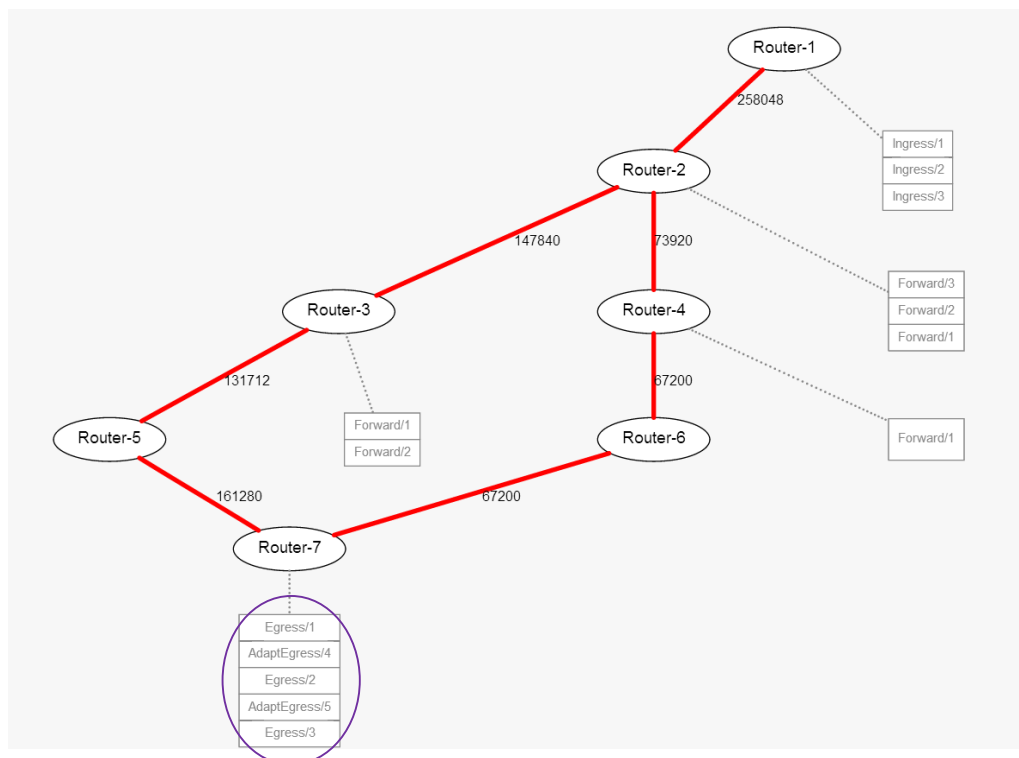


Figure 28: Scenario 1 – Egress path adaptation in the core network together with the second access network handover.

A live demonstration of the scenario was part of the UniverSelf demo booth presented in Future Network and Mobile Summit 2013 [20] in Lisbon, and reported in D4.13 [16].



### 2.3.1.1 Governance block

Governance core block plays a fundamental role in this scenario, in particular the translation process.

The Service defined through the user interface leads to a policy that can be expressed in a close-to-natural language as:

*On the Access Part of the network, with specific technology 'WLAN', process of type 'Reconfiguration' should exist, in order to achieve goal 'Load Balancing', taking also into consideration users who consume 'Video' service and have SLA agreement, which reassure 'QoS Excellent'.*

This kind of human-readable policy should be further translated into a language that can be understood by the different elements involved in the service provisioning operations. This task is accomplished by the Policy Derivation and Management function. The policies of all levels are modelled using Shared Information and Data (SID)- based ontology in Figure 5 and expressed in OWL [21]. The ontologies of different levels are linked through relationships between classes, which express the interrelation between subsequent levels. The translation then is realised by the implementation of algorithms expressed in SWRL (Semantic Web Rule Language) [22], able to transform, generate and deliver data from one level to the subsequent level.

An example of policy translation process, as it has been defined in Human-to-Network interface is shown in Table 1:

Level	Policy
Business Level	$NEM(?n) \wedge hasGoal(?n, "Load\ Balancing") \rightarrow hasNetworkDomain(?n, ?dom) \wedge hasTechnology(?n, ?tech)$
Service Level	$NEM(?n) \wedge hasNetworkDomainType(?dom, "Access") \wedge hasTechnologyType(?tech, "WLAN") \rightarrow hasProcess(?n, proc)$
NEM Level	$NEM(?n) \wedge hasProcessType(?proc, "Monitoring") \rightarrow send\_gov\_rule(1)$

**Table 1: Example of SWRL rules for translation of high level business goals into low level NEM function selection**

The scalability of this process has been examined [6] under various numbers of generated policies. In general the number of policies that the H2N function needs to generate depends on the size and the heterogeneity of the network (number of different services, technologies etc.). A set of measurements have been realised in order to study the delay of the translation process under different load of generated policies. The results of this study are illustrated in Figure 29. When the number of deployed policies is low, the delay of the translation process is insignificant (e.g. around 2.5 sec in case of 3 policies), while it highly increases with the number of generated policies (e.g. reaching 6 min for 50 policies). Given the fact that the introduction of new high-level objectives only impacts a limited number of policies, the delay is insignificant and does not affect the operation of the overall system.

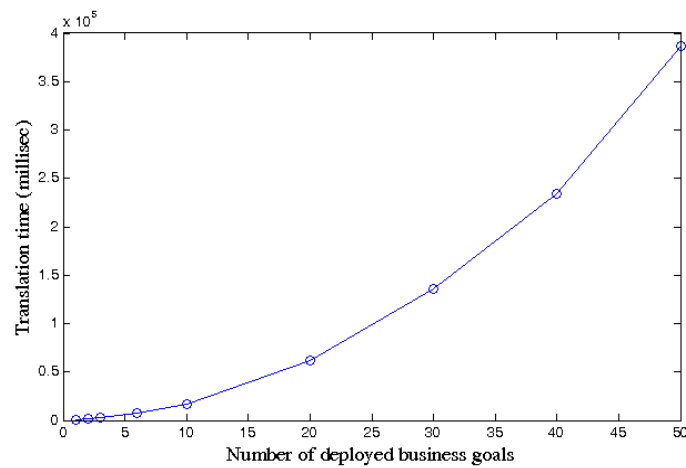


Figure 29: Policy translation delay

### 2.3.2 Scenario 2 - Conflict-free Coordination of SON NEMs

In this scenario, conflict-identification and -resolution function of the Coordination block of UMF are highlighted.



Figure 30: Typical situation of SON functions (NEMs) not being aware of other affecting functions' existence and operation

In particular, two SON functions (NEMs) implement self-optimization functions in a LTE-Advanced heterogeneous network (HetNet) using a hierarchical conflict free setup. SON 1 (Load Balancing -LB in short- NEM) adjusts relays' coverage to offload the right amount of eNodeB traffic, and SON 2 (Backhaul Optimization -BHO in short- NEM) balances radio resources between backhaul and station-to-mobile links.

One instance per NEM is created and deployed over the simulated RAN presented in paragraph 2.1.2 through the UMF dashboard (see Figure 31).

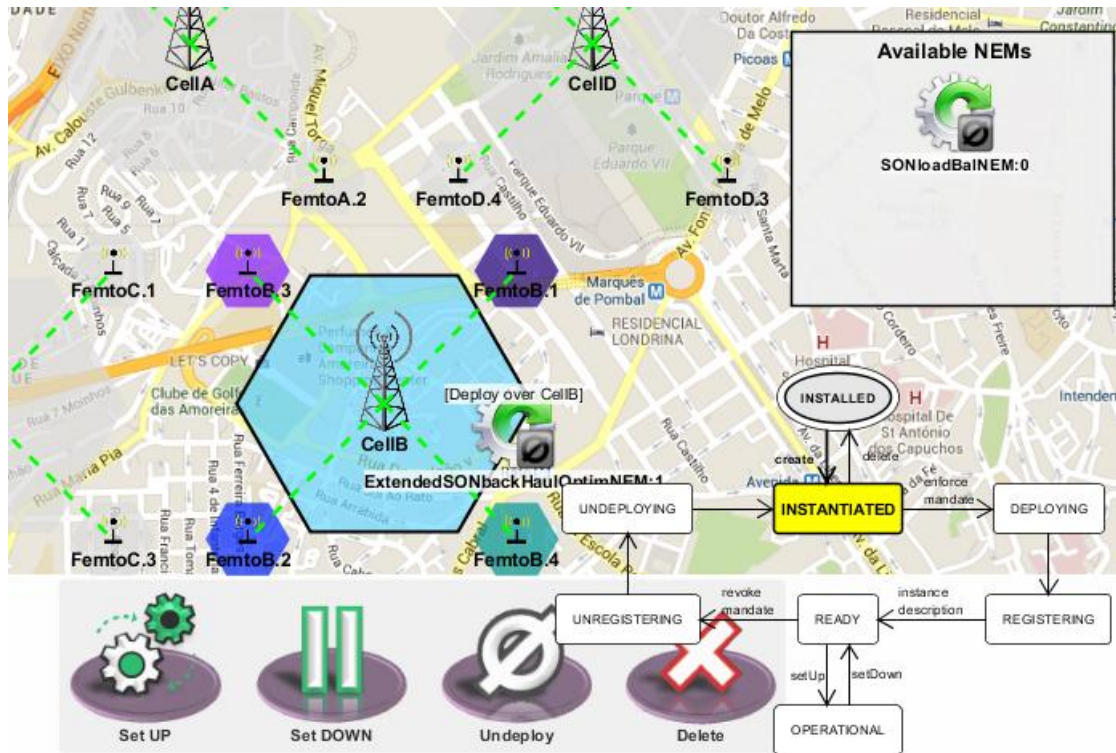
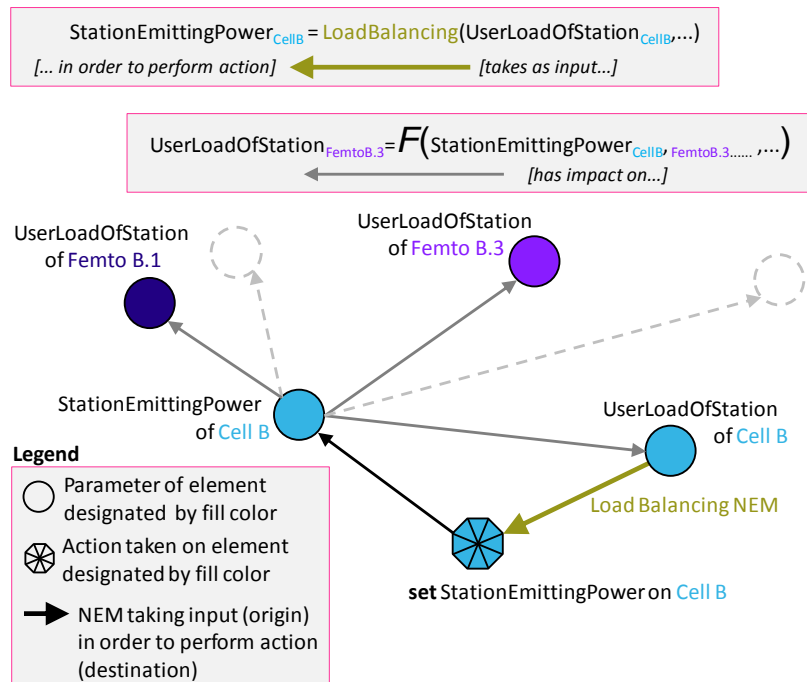
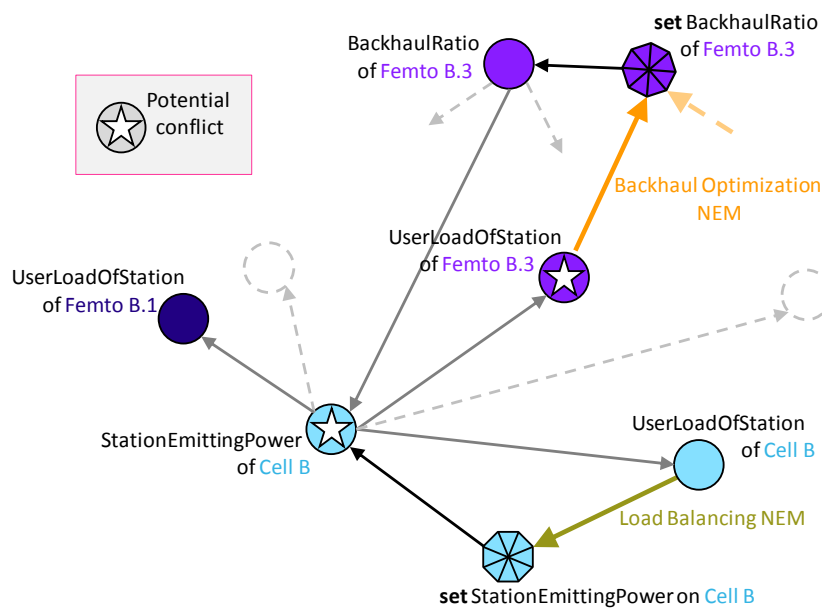


Figure 31: The LTE RAN view of the UMF dashboard just prior to the deployment of the NEMs

At that time, the users in the network are uniformly spread and there is no load balancing to be performed by the LB NEM. Prior to turning on (UP-state) these NEMs, Coordination receives their registration request with an enclosed instance description (as per the UMF specifications) for each, containing the elements to be managed (CellB in this case), any UMF information specifications to be produced and consumed as well as the UMF actions to be taken (their specifications). Conflict identification then relies on such instance descriptions to build mathematical graphs plotting all the specifications acquired by the NEMs and any inter-relations. A conflict is detected when an UMInformationSpec is part of two control loops belonging to two different NEMs. Figures below depict subsets of such a conflict graph produced by the Coordination block and visualized on the H2N interface, prior (Figure 32) and after (Figure 33) potential conflicts are identified.



**Figure 32: Conflict graph built by Coordination block during the deployment of a Load Balancing SON NEM**



**Figure 33: Conflict graph further populated by UMF during the deployment of an additional Backhaul Optimization SON NEM**

Colored arrows denote "*takes as input*" and greyscale arrows, denote "*has impact on*" associations (derived from the UMF ontology) while the conflicting metrics are denoted with a *star*. In particular, during the deployment of the BHO NEM (see Figure 33) that takes as input a metric that is being actively modified by the LB NEM, the system identifies a potential conflict/instability during their concurrent execution.

In this particular, demonstrated scenario, potential conflicts are identified as BHO NEM is taking as input the user load of stations (UserLoadOfStation) in order to decide and take action while at the same time the LB NEM's actions (SetStationEmmittingPower) are highly impacting the very same parameter (due to coverage change; knowledge derived from the UMF ontology) and driving BHO to decisions and adjustments that are

only temporarily correct, possibly contradicting or non-convergent in the long-term. Therefore, the network is driven to an unstable state unless coordination intervenes.

Therefore, UMF Coordination enforces NEM operation separated in time with the correct scales: short and long time scales for NEM 1 and NEM 2 respectively to guarantee convergence and stability. Such paces are applied through corresponding UMF Regimes instructing NEMs to operate continuously but with different time intervals between each iteration through their MAPE loops. Figure 34 and Figure 35 are snapshots of the collected KPIs for both the non-coordinated and the coordinated versions of the very same deployment scenario, respectively.

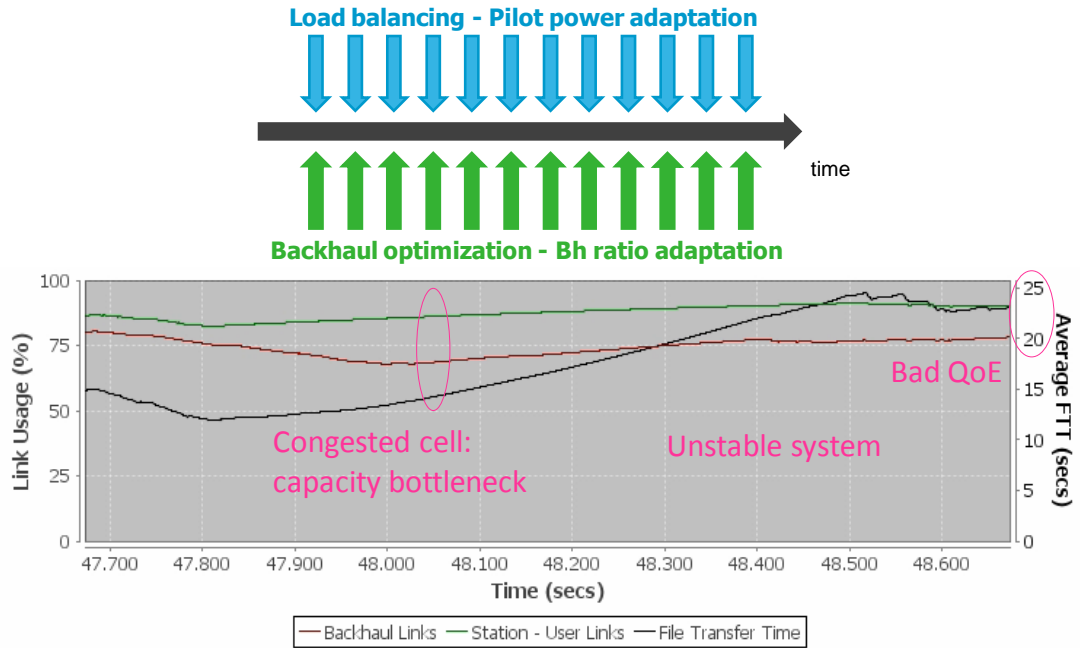


Figure 34: Snapshot of the KPIs when the 2 SON NEMs are not being coordinated (operating in the same pace)

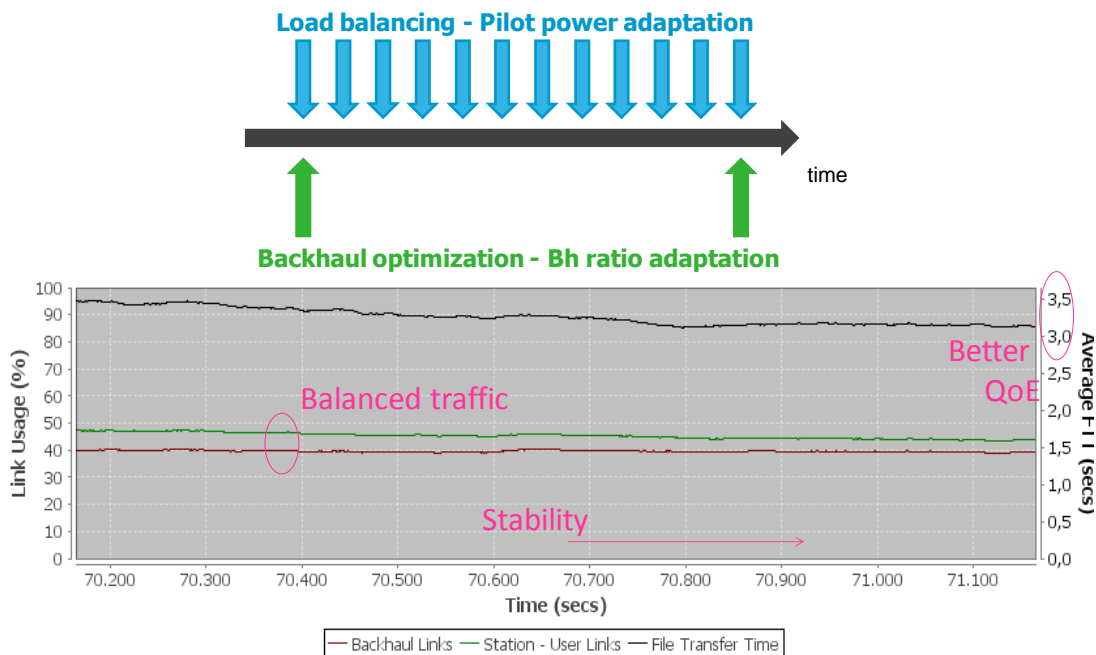


Figure 35: Snapshot of the KPIs of the very same SON NEM combination being coordinated using Time-Separation (slow/fast paces being assigned)

Finally, after the NEMs are turned on (UP-state), and coordinated, the user arrival rates are manually changed to unbalance the traffic and demonstrate the effects of both the NEMs on their own, but also of the Coordination mechanism being applied on them.

### 2.3.3 Scenario 3 - UMF Management of Software Driven Networks (SDN)

The VLSP usage and integration within the rest of UMF is presented, together with a scenario showing integration with other NEMS.

#### 2.3.3.1 The Virtual Infrastructure Management NEM Functionality

In this section we show how the Virtual Infrastructure Management NEM, is integrated with the rest of UMF, also highlighting integration with other NEMS.

In Figure 36 a scenario is illustrated which shows the VIM NEM managing a virtual network used for the transfer of data. There are a further two NEMS, called NEM A and NEM B, which are responsible for managing networks in completely separate domains. For example, NEM A may manage a CDN network, and NEM B may manage a wireless radio network.

By all using the features of UMF and by being connected by the 3 main management blocks (top part of Figure 16) allows these NEMs to interact with each other, and also to be controlled in an organised way.

In the bottom part of Figure 16, the red line depicts some of these traffic flows across multiple networks. Within UniverSelf we have built, tested, and demonstrated such an integrated scenario using video traffic from a small CDN. This video data has flowed over a software defined virtual network that has been setup and configured using some SDN concepts. After leaving the virtual network, the video data has flowed onto a wireless network and has been displayed on various display devices.

Such integration shows the use of various NEMs in the same instantiated context, e.g. the Load Level Estimation NEM, the Core Traffic Engineering NEM, and the Virtual Infrastructure NEM.

Furthermore it highlights the integration of the data plane in the same instantiated and managed context. In the demonstration of this integrated scenario, traffic flows from a video server source in one domain, across the virtual network nodes, and into a third domain with a video player.

The Software Defined Virtual Network acts as one segment of a full network over which data will flow. Each segment of the network is managed by its own NEM. There is overall UMF management using Governance, Coordination, and Knowledge.

We have demonstrated real-time traffic flowing from one network segment, into the virtual network, and out to another network segment.

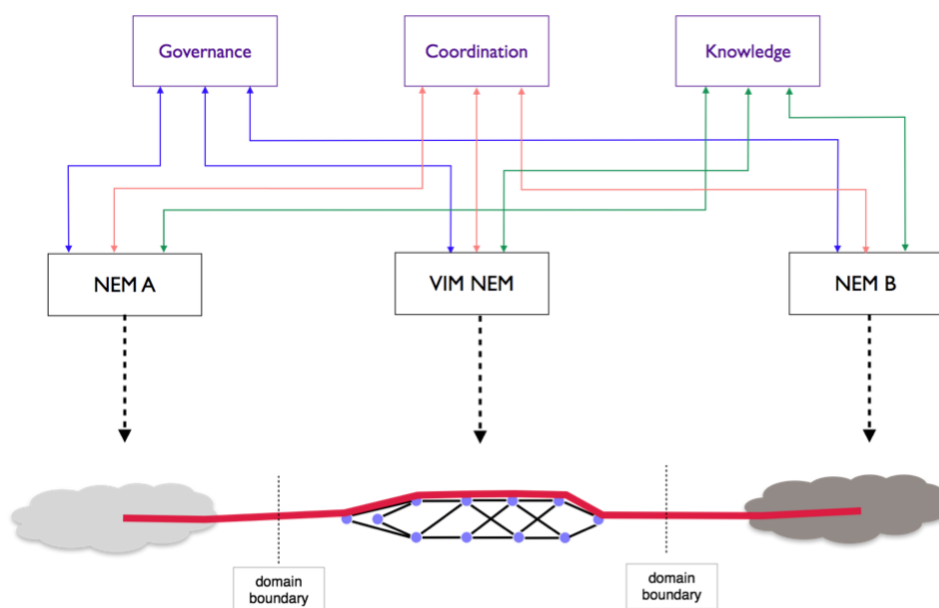
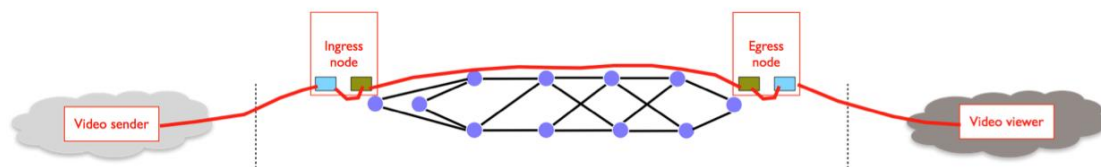


Figure 36: VIM NEM Integration with UMF

The combination of adaptations to various governance rules has updated the NEMs and then as a consequence the management of the network elements has been adjusted. In particular, we have used some SDN techniques to adapt the flow of the video to different display devices as the management and network environments have changed - all at on-the-fly at run-time.

The virtual network was setup incrementally: one node at a time and one link at a time. This was shown at FUNEMS Lisbon 2013, where we showed integration between many elements of the project. After the topology of the virtual network was created, under full management control, next an Ingress node and an Egress node were started on key nodes in the virtual network. These are special applications that are also started under management control. The Ingress and Egress applications have been specially developed to allow data plane integration between the various network segments.

The Ingress node is an application that listens on a specific port and sends any received data onto another application in the virtual network. The Egress node is an application that listens for data in the virtual network and sends that traffic onwards to a different port. A data source, such as a video server, sends its traffic to the network address of the Ingress node. The Ingress node forwards data to the Egress node, and finally the Egress node forwards the video traffic onto the video player. The full flow of traffic across the multiple domains is shown in Figure 37.



**Figure 37: Traffic Flow across Multiple Domains**

The above was shown at FUNEMS Lisbon 2013, where we demonstrated integration between many elements of the project. Compared to FUNEMS, we improved integration aspects with the UMF, i.e., being compliant with UMF v3.0. We extended the VIM API in order to accommodate better the diverse needs of other NEMs that may need to access information or facilities from the SDN infrastructure. Furthermore, we support Closure, a high-level functional language that is very convenient for representing and performing SDN configuration changes. Last but not least, we improved the distribution of the infrastructure over many physical machines.

The VIM NEM is not a standalone system, but acts within the scope of a full UMF system. One of the main integration aspects is with the Knowledge Block. Data and network interface measurements from the VIM NEM are being passed into Knowledge. This data is being used by other parts of UMF, such as other NEMs, for various management and reporting purposes.

The VIM NEM regularly updates the Knowledge Block with the current topology of the network. It sends separate information regarding the routers and the links. This allows any consumer of the data to build a full graph topology of a network, or just consider the routers or the links, if needed. The data is sent to the Knowledge via an information consumer and is formatted using the JSON annotation style.

For the routers the following fields are sent:

- list - which is a list of router IDs
- seq - which is a sequence number
- type - which has the value 'router' to indicate this is router information
- detail - which has a list of detail on each router. The detail for each router is as follows:
  - address - the address of a router
  - links - the IDs of the routers this router is connected to
  - mgmtPort - the port the router listens on for management commands
  - name - the name of the router
  - r2rPort - the port the router listens on for making router to router connections
  - routerID - the ID of the router



- time - the time the router was created, in milliseconds since Jan 1 1970

An example of such data is shown below. It contains data for 5 routers.

```
{
  "list": [6, 5, 7, 2, 4],
  "seq": 36,
  "type": "router",
  "detail": [
    {
      "address": "6",
      "links": [2, 5, 4, 7],
      "mgmtPort": 11010,
      "name": "Router-6",
      "r2rPort": 11011,
      "routerID": 6,
      "time": 1373635053851
    },
    {
      "address": "5",
      "links": [4, 2, 6],
      "mgmtPort": 11008,
      "name": "Router-5",
      "r2rPort": 11009,
      "routerID": 5,
      "time": 1373635049618
    },
    {
      "address": "7",
      "links": [6, 4, 2],
      "mgmtPort": 11012,
      "name": "Router-7",
      "r2rPort": 11013,
      "routerID": 7,
      "time": 1373635057645
    },
    {
      "address": "2",
      "links": [4, 5, 6, 7],
      "mgmtPort": 11002,
      "name": "Router-2",
      "r2rPort": 11003,
      "routerID": 2,
      "time": 1373635034475
    },
    {
      "address": "4",
      "links": [2, 5, 6, 7],
      "mgmtPort": 11006,
      "name": "Router-4",
      "r2rPort": 11007,
      "routerID": 4,
      "time": 1373635043786
    }
  ]
}
```

For the links the following fields are sent:

- list - which is a list of link IDs
- seq - which is a sequence number
- type - which has the value 'link' to indicate this is link information
- detail - which has a list of detail on each link. The detail for each link is as follows:
  - id - the ID of the link
  - name - the name of the link
  - nodes - the IDs of the routers this link is connected to
  - time - the time the link was created, in milliseconds since Jan 1 1970
  - weight - the link weight for the link

An example of such data is shown below. It contains data for 9 links.

```
{
  "list": [1048625, 1572889, 2293796, 2097201, 786457, 3145777, 1835044, 917540, 655376],
  "seq": 37,
  "type": "link",
  "detail": [
    {
      "id": 1048625,
      "name": "Router-2.Connection-4",
      "nodes": [2, 7],
      "time": 1373635058320,
      "weight": 1
    },
    {
      "id": 1572889,
      "name": "Router-4.Connection-0",
      "nodes": [4, 5],
      "time": 1373635050009,
      "weight": 1
    },
    {
      "id": 2293796,
      "name": "Router-5.Connection-0",
      "nodes": [5, 6],
      "time": 1373635054379,
      "weight": 1
    },
    {
      "id": 2097201,
      "name": "Router-4.Connection-2",
      "nodes": [4, 7],
      "time": 1373635058190,
      "weight": 1
    },
    {
      "id": 786457,
      "name": "Router-2.Connection-2",
      "nodes": [2, 5],
      "time": 1373635050158,
      "weight": 1
    },
    {
      "id": 3145777,
      "name": "Router-6.Connection-0",
      "nodes": [6, 7],
      "time": 1373635058040,
      "weight": 1
    },
    {
      "id": 1835044,
      "name": "Router-4.Connection-1",
      "nodes": [4, 6],
      "time": 1373635054530,
      "weight": 1
    },
    {
      "id": 917540,
      "name": "Router-2.Connection-3",
      "nodes": [2, 6],
      "time": 1373635054249,
      "weight": 1
    },
    {
      "id": 655376,
      "name": "Router-2.Connection-1",
      "nodes": [2, 4],
      "time": 1373635044318,
      "weight": 1
    }
  ]
}
```

Furthermore, as the VIM NEM manages virtual routers that are instrumented and are producing live monitoring data, it is able to provide live information about the volumes of traffic on each of the interfaces of each virtual router.

For each router the following data fields are available.



- type - which has the value 'link\_stats' to indicate this is link stats information
- routerID - the ID of the router
- links - the IDs of the routers this router is connected to
- seq - which is a sequence number
- link\_stats - which has a list of the stats on each link. The detail for each link stat an ordered list of values. The detail is as follows:

```

    ▶ name | InBytes | InPackets | InErrors | InDropped | InDataBytes | InDataPackets | OutBytes |
    OutPackets | OutErrors | OutDropped | OutDataBytes | OutDataPackets | InQueue | BiggestInQueue |
    OutQueue | BiggestOutQueue
{"type":"link_stats", "routerID":2, "links":[33,27,80,150,122], "seq":1399, "link_stats": [
    0 1],
    ["Router-33 Router-2.Connection-10" 189619 676 0 0 115260 565 79773 155 0 0 11016 54 0 1
    1 0 2],
    ["Router-27 Router-2.Connection-7" 159104 627 0 8 113220 555 140405 452 0 0 69156 339 0
    ["Router-80 Router-2.Connection-20" 238638 1012 0 0 193800 950 56501 69 0 0 0 0 1 0 1],
    ["Router-150 Router-2.Connection-25" 127716 548 0 0 106080 520 39043 43 0 0 0 0 1 0 1],
    ["Router-122 Router-2.Connection-24" 134112 601 0 0 118524 581 40661 45 0 0 0 0 1 0 1]]}

```

The VIM NEM continually produces such data, and the Knowledge Block stores such data.

### 2.3.3.2 The Placement Optimization NEM Functionality

The Placement Optimizer NEM (PO NEM) optimizes the deployment of special management nodes within a network. This location distribution is used in various network management functions, and the nodes which are placed can have various functions such as monitoring collection nodes and monitoring aggregations points. For this purpose there exist various algorithms which take the current topology and current traffic volumes as input and produce as output the nodes which have been determined to be 'special'.

The Placement Optimizer NEM takes information about the current routers and current links in a network and on a regular basis (currently configured to be 10 seconds) it applies an algorithm to determine the placement of the special nodes. It combines the router data and the link data to form a graph form of the network. It then applies one of HotSpot, Pressure, or PressureTime algorithms to determine the placement of the nodes.

As new routers and links are created by the VIM NEM, this data is sent to the Knowledge Block. The PO NEM interacts with Knowledge, collects the latest router and link data, and updates the network graph. This interaction is seen in Figure 38.

After the network graph has been updated the PO NEM then runs the algorithm again. If there are any differences in the output from the previous run, the PO NEM injects control update commands which are sent to the VIM NEM in order to enforce the decisions and to change the placement of the collection nodes and the aggregation point nodes.

These decisions become control commands to the VIM NEM. At 00:37 the PO NEM sets Router 2 as an aggregation point. At 00:47 it tells Router 3 to use Router 2 as an aggregation point for collected monitoring data.

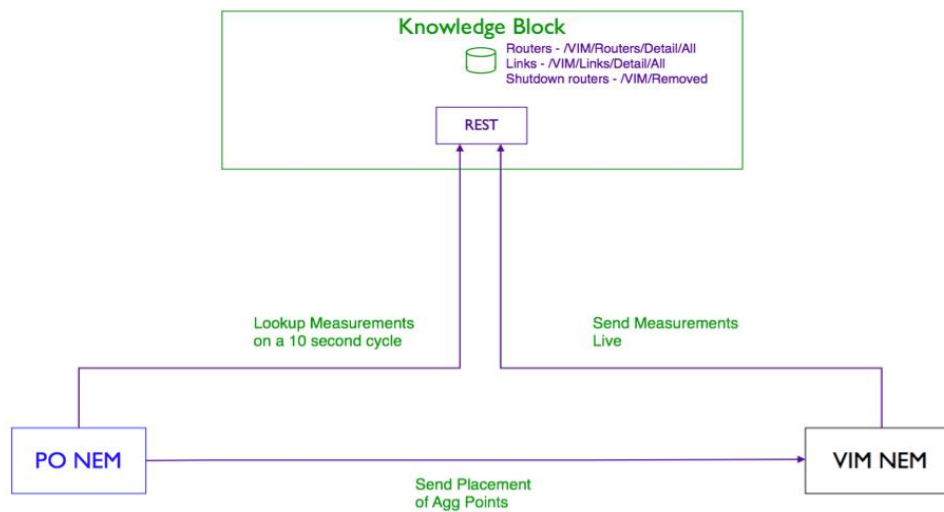


Figure 38: PO NEM Integration with VIM NEM and Knowledge

We have built and tested this for scalability in both simulation (with over 35,000 thousand nodes) and on a custom-made virtual network testbed (with over 700 virtual routers), where we show that it performs well at a variety of monitoring tasks. The system is scalable and that intelligent configuration of the monitoring system greatly improves its efficiency.

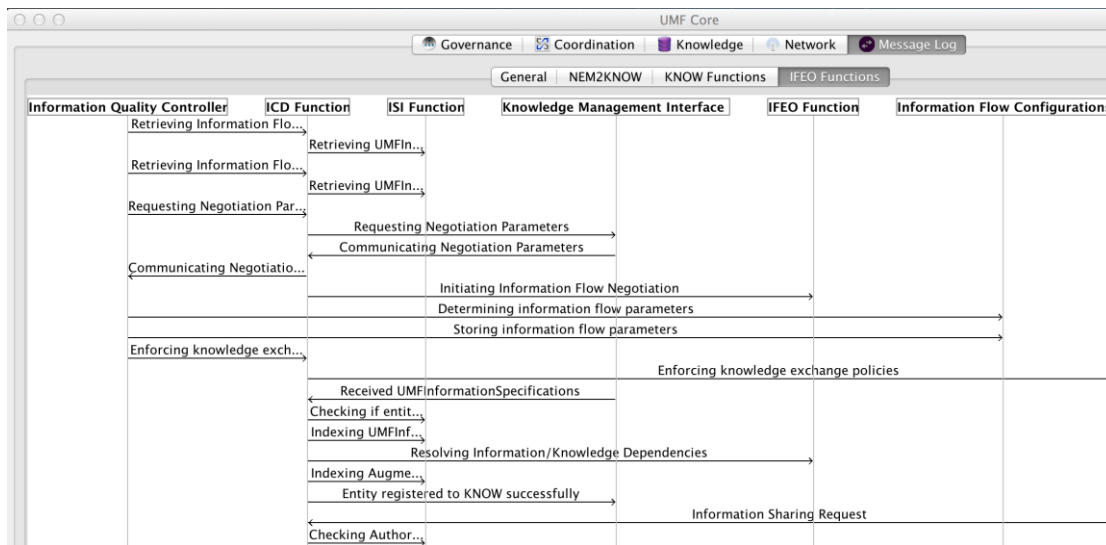
### 2.3.3.3 The Knowledge Block Functionality

The KNOW block implementation is consistent with the UMF v3.0 specifications documented in D2.4. As we show in Figure 39, its user interface is part of the unified UMF dashboard. The corresponding GUI shows an overview of the KNOW architecture at the left hand side and real-time measurements regarding its operation at the right hand side. The different measurements can be enabled or disabled through the KNOW configuration file. They include the instantaneous CPU load of KNOW related processes, the memory state required and the total numbers of active flows and numbers of flows per communication method (i.e., pub/sub, push/pull and direct communication).



Figure 39: The Knowledge Block Integrated in the UMF Demo Dashboard

By clicking the output tab, a view of the Knowledge Block console output is enabled. The details of the communication between the UMF entities (i.e., NEMs or other UMF core blocks) and the KNOW is being displayed through corresponding console messages. A message workflow graphical representation of the same communication can be shown through the Message Log tab. Different workflows show alternative viewpoints of the KNOW related communication. The IFEO view point shows the messages exchanged in the Information Flow Establishment and Optimization Function. This workflow is useful for highlighting the information/knowledge flow negotiation activity of the Knowledge block. The NEMS2KNOW show the messages exchanged between the UMF entities (i.e., either core services or NEMs) and the Knowledge block. It is actually considering KNOW as a black box. The KNOW functions workflow in Figure 40 shows the messages exchanged between the different KNOW functions. So, the KNOW activities are totally transparent to the administrator.



**Figure 40: The Knowledge Block Related Workflow Diagrams**

The KNOW implementation can be configured using the know.cfg configuration file. The available configuration options include: (i) KNOW GUI interface related options (e.g., whether it is embedded in the dashboard or not, visualization window size etc), (ii) the default global optimization goal used which is important for the information/knowledge flow negotiation functionality, (iii) credential information for the redis key-value database, which is the heart of the KNOW storage, and (iv) which real-time measurements to enable and show in the KNOW GUI.

## 2.4 Conclusions

This chapter presented the final UniverSelf integrated prototype, aiming to showcase how the project solutions can enrich the network operation and facilitate the network management. Three different testbeds were used, in each of which different NEMs were deployed:

- The first scenario highlighted the QoS management in a multidomain network, built with a SDN core and WLAN access network.
- The second scenario showed how coordination of two competing SON functions can be effectively achieved with UMF
- The third scenario showed a new and developed Very Lightweight Software Driven Network and Service Platform (VLSP) testbed and its use in the UMF management of Software Driven Networks (SDN).

We have shown the integration of the elements of UMF into a fully working, multiple domain demonstration, showing the integration of multiple NEMs in the same context together with physical and virtual network acting as part of a full data plane.

### 3 OPEX Impact Analysis

This section proposes a model for the calculation of the impact of autonomics in a Network Operator's OPEX. Although a number of market research reports present potential OPEX savings of autonomics, literature does not discuss methods to calculate the OPEX benefits of the implementation of autonomic networking. The model presented here combines the simulation performance results of the NEMs with an expert input resulting from an *a posteriori* analysis in order to get an accurate as possible OPEX-gain indication. This methodology and relevant inputs are presented in section 3.1. The model is then applied to an extended version of Scenario 2 (the so-called Use case 4 *SON and SON collaboration according to operator's policies* [23]), introduced in Section 2.1.2. Section 3.2 presents these results, while section 3.3 discusses the results as well as the methodology. Finally, in Section 3.4, the general conclusions are presented.

#### 3.1 Toy model methodology

##### 3.1.1 Overview

Quantifying OPEX gains for a non-existent 'typical' network operator is not a straightforward task, so an established base of simplified models and assumptions (like the eTOM framework and OPEX models from scientific literature) will be used; for this reason it is referred to as a 'toy' model<sup>1</sup>. The goal of the Toy model methodology is threefold: 1) to provide a pragmatic, simple and explicit tool to compute the attainable OPEX reductions based on NEM numerical performance evaluations; 2) to reason about the aggregate findings for scenarios; and, 3) to deduce trends and assertions about the potential of UniverSelf technologies to impact OPEX.

It is not obvious to identify on the basis of the scenario and the UMF perspectives how OPEX would be impacted. There are several dimensions to consider:

1. The NEMs and methods that are designed and developed focus on solving operational problems by providing self-x features. The direct impact related to these operations is easily identified. However, impact on other facets of OPEX has to be identified and the link between performance of these features and OPEX also needs to be identified.
2. The impact of the adoption and use of UMF is larger than the operations that are directly impacted by self-x features. In that respect, the integration and management tool life cycle are also concerned.
3. OPEX is a complex notion with many aspects. When referring to OPEX many cost categories, going from e.g., personnel to energy consumption, need to be identified.

In order to capture all these dimensions, we have proposed to identify the OPEX impact based on the processes of the TMF Business Process Framework (eTOM) [24]. An exhaustive analysis of the processes and their definition helps to identify where NEM, or UMF adoption have some impact.

The proposed methodology is depicted in Figure 41. It uses NEM metric results provided by the NEM owners as a starting point, which are mapped onto eTOM *process flows*. Since expert input is inevitable in this kind of mapping, it would be relevant to also involve the structure expert mapping that was already available in the project: a QFD analysis (see e.g., [25][26]) that constituted an *a priori* analysis. This happens in branch B. Finally, in Branch C, the two earlier inputs converge in a mapping on OPEX categories, which on its turn can provide the OPEX impact per NEM.

---

<sup>1</sup> The term 'toy model' has its origins in physics; e.g., see <[http://en.wikipedia.org/wiki/Toy\\_model](http://en.wikipedia.org/wiki/Toy_model)>

A more detailed level of the new model is presented in Figure 42. In this figure, the blue, red and purple elements still correspond to branches A, B and C respectively. Apart from the branch, also the type of element is indicated. Ovals refer to simulation results; angled boxes refer to inputs from experts or literature; rounded boxes refer to results calculated using the two previous types.

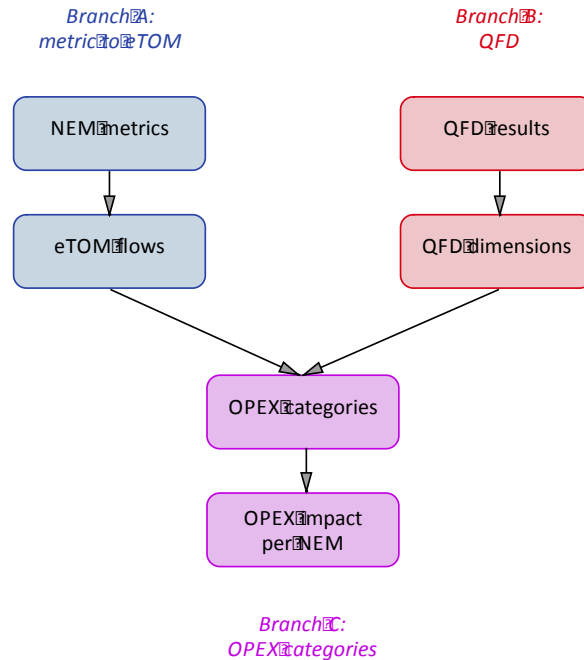


Figure 41: A high-level overview of the Toy Model

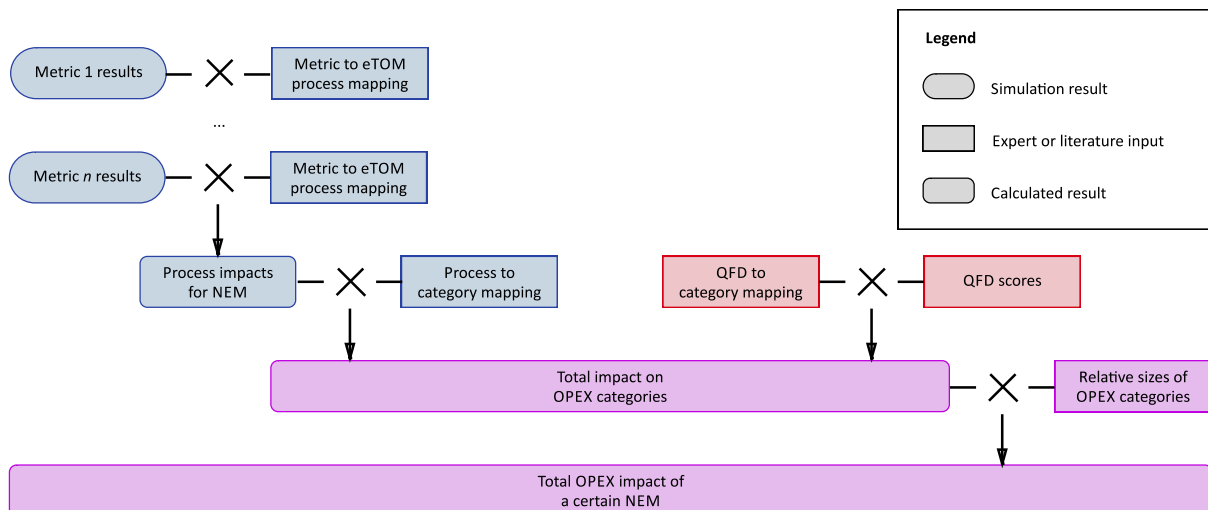


Figure 42: The Toy Model in detail

The metric results that form the starting point for our analysis are presented in Subsection 3.1.2. The QFD scores that had been provided in an *a priori* analysis are used here. For the other three types of expert inputs (metric to eTOM process mapping, eTOM process to category mapping, QFD to category mapping), these results are presented in Subsection 3.1.3. For the relative sizes of OPEX categories, a reference from literature has been used, which is explained in detail in Subsection 3.1.3 as well. Finally, in Section 3.2, the results are presented.

### 3.1.2 NEM metric inputs

The metrics used as input in this exercise have been identified by the NEM owners along with simulation results. An overview of the NEMs together with their metrics is presented in Table 2.

The results obtained by the deployment of the NEM 3 – *Inter-Cell Interference Coordination* (best case) are compared with the results obtained by the deployment of NEM 4 – *Coverage and Capacity Optimization* (base case). In the same way, referring to NEM 4, the results obtained by the deployment of NEM 4 (best case) are compared with the results obtained by the deployment of NEM 3 (base case). NEM 3 presents gains on inter-cell Interference and SINR while there are losses on cell throughput and user throughput. The opposite remarks are valid while considering NEM 4 instead.

NEM 28 – *Prediction-based Load Balancing* metrics refer to outage derived from the number of dropped mobile devices and user satisfaction as the delta between the actual and predicted user satisfaction. An algorithmic framework is used for the proactive load balancing of user decision-making requests, assuming reconfigurable mobile devices (base case) and autonomous mobile devices (best case). The obtained results show the gains of the presented framework in terms of the number of dropped user requests for autonomous mobile devices.

For NEM 32 – *Self-healing mechanism for Cell Outage Management*, a fuzzy inference system is considered and the values presented refer to worst performance results (base case) and the best performance results. These indicative results have been simulated for a 3x3 grid (9 Access Points (APs)) and for a varying number of Mobile Terminals. The algorithm uses a fuzzy inference system with the following inputs: a) the compensating AP's load and b) a metric of the overlapping area that each compensating AP creates with the neighbouring APs.

NEM 34 – *LTE Load Balancing* and NEM 35 – *LTE Tilt Optimization* adapt the cell and/or antenna parameter in such a way that cell resources and coverage area are best adapted to serve the user traffic. Thereby the achieved performance gains do strongly depend on the amount and distribution of traffic. Very large performance gains can in particular be achieved with diverse traffic loads, varying between neighbouring cells and at an overall traffic load level which is in the medium-high-load to high-load range. In contrast the NEM gains are very low for very uniform traffic distributions and at traffic load levels of a either a basically empty system or of a complete overload everywhere. For realistic traffic modelling and diversity, the typical NEM gains are in the order of around 10%. These NEM simulation results of macro-cellular studies without traffic hot-spots, are presented below.

As for NEM 46 – *GP Coverage*, a static coverage is used as a base case for comparison, where femtocells are configured with a fixed pilot power of -29 dBm. The results provided below represent the performance under overloaded conditions. The load supported by the femtocells with the coverage algorithm is approximately 20% higher than the load supported by the fixed coverage deployment. The number of macrocell user mobility requests with the coverage algorithm is significantly lower compared with the fixed coverage deployment. The number of mobility events between macrocells and femtocell is slightly higher due to the femtocells creating temporary coverage holes when changing coverage when load balancing, but this is not significant.

		Best Case	Base Case	Gains <sup>2</sup>
NEM 3: Inter-Cell Interference Coordination (ICC)	Inter-cell interference (dBm)	-47,3	-46,2	2%
	SINR (dB)	12,8052	12,3399	4%
	Cell throughput (Kbps)	7720	12440	-38%
	User throughput (Kbps)	643	1040	-38%
NEM 4: Coverage and Capacity Optimization (CCO)	SINR (dB)	12,3399	12,8052	-4%
	Cell throughput (Kbps)	12440	7720	61%
	User throughput (Kbps)	1040	643	62%

<sup>2</sup> Gain = 100 \* (best case – base case)/base case



NEM 28: Prediction-based Load Balancing	Outage	9,3%	11,1%	16%
	User satisfaction	0	0,06	100%
NEM 32: Self-healing mechanism for Cell Outage Management (SH)	Interference indicator	0,06	1,72	97%
	Load fairness index	0,581	0,709	18%
	Ratio of recovered terminals	6/6	1/1	0%
NEM 34: LTE Load Balancing	Cell throughput (Mbps/cell)	10,6	10,3	3%
NEM 35: LTE Tilt Optimization	Cell throughput (Mbps/cell)	20,9	19	10%
	Cell edge throughput (Mbps/cell)	1,6	1,2	33%
NEM 46: GP Coverage	Average load supported by femtocells (Erlangs)	80,91	64,89	25%
	Average macrocell user mobility requests per pass	0,1835	3,9998	95%
	Average femto ↔ macro mobility events experienced by a femtocell user per hour	0,0938	0,0627	-50%

Table 2: NEM metric inputs based on simulation results

### 3.1.3 Conversion tables

This subsection contains the three mapping tables used in the Toy Model generated by expert inputs (metric to eTOM process mapping, eTOM process to category mapping, QFD to category mapping) as well as the cost category mapping that has been taken from literature. As explained previously, the data in the first three tables are based on an expert survey. Five experts have been identified amongst the project members to complete the survey, of which two decided to work together to complete one survey. The team responsible for the method and the analysis completed one additional survey together. The results have been merged into three mapping tables.

The ‘Metrics to eTOM flows’ tables map the impact of the different NEM metrics onto the eTOM flows of the two most relevant processes: Customer Centric E2E Business Process and Network Centric E2E Business Process. The results are presented in Table 3 and Table 4 respectively.

Metric to eTOM flow Customer Centric E2E Business Process		Request to Answer	Order to Payment	Usage to Payment	Request to Change	Termination to Confirmation	Problem to Solution	Complaint to Solution
NEM 3: Inter-Cell Interference Coordination (ICC)	Inter-cell interference	0%	0%	0%	1%	0%	5%	2%
	SINR	0%	0%	0%	1%	0%	6%	2%
	Cell throughput	0%	0%	0%	1%	0%	5%	1%
	User throughput	0%	0%	0%	1%	0%	5%	1%
NEM 4: Coverage and Capacity Optimization (CCO)	SINR	0%	0%	1%	2%	0%	6%	2%
	Cell throughput	0%	0%	1%	2%	0%	6%	2%
	User throughput	0%	0%	1%	2%	0%	7%	3%
NEM 28: Predication-based Load Balancing	Outage	0%	0%	0%	1%	0%	12%	2%
	User satisfaction	0%	0%	0%	1%	0%	6%	1%
NEM 32: Self-healing mechanism for Cell Outage Management (SH)	Interference	0%	0%	0%	1%	0%	8%	3%
	Load fairness index	0%	0%	0%	1%	0%	11%	1%
	Percentage of reconnected MTs	0%	0%	0%	1%	0%	19%	9%

NEM 34: LTE Load Balancing	Cell throughput	0%	0%	0%	1%	0%	4%	1%
NEM 35: LTE Tilt Optimization	Cell throughput	0%	0%	0%	1%	0%	3%	0%
	Cell edge throughput	0%	0%	0%	1%	0%	4%	0%
NEM 46: GP Coverage	Average load supported by femtocells	0%	0%	0%	1%	0%	7%	6%
	Average macrocell user mobility requests per pass	0%	0%	0%	1%	0%	2%	0%
	Average femto ↔ macro mobility events experienced by a femtocell user per hour	0%	0%	0%	1%	0%	4%	2%

Table 3: Metric to eTOM flow Customer Centric E2E Business Process

Metric to eTOM flow Network Centric E2E Business Process		Production Order to Acceptance	Trouble Ticket to Solution	Usage to Usage Data	Capacity Management	Service Lifecycle Management	Resource Lifecycle Management
NEM 3: Inter-Cell Interference Coordination (ICC)	Inter-cell interference	2%	6%	0%	14%	2%	8%
	SINR	2%	6%	0%	13%	3%	8%
	Cell throughput	2%	7%	0%	14%	2%	8%
	User throughput	2%	7%	0%	14%	2%	8%
NEM 4: Coverage and Capacity Optimization (CCO)	SINR	1%	6%	0%	17%	3%	10%
	Cell throughput	2%	6%	0%	18%	3%	10%
	User throughput	2%	7%	0%	18%	3%	10%
NEM 28: Predication-based Load Balancing	Outage	1%	9%	0%	8%	3%	11%
	User satisfaction	1%	4%	0%	5%	2%	9%
NEM 32: Self-healing mechanism for Cell Outage Management (SH)	Interference	1%	3%	0%	4%	2%	9%
	Load fairness index	1%	7%	0%	3%	3%	13%
	Percentage of reconnected MTs	1%	14%	0%	8%	3%	13%
NEM 34: LTE Load Balancing	Cell throughput	1%	4%	0%	12%	2%	10%
NEM 35: LTE Tilt Optimization	Cell throughput	1%	4%	0%	19%	2%	9%
	Cell edge throughput	1%	4%	0%	19%	2%	9%
NEM 46: GP Coverage	Average load supported by femtocells	2%	2%	0%	14%	2%	10%
	Average macrocell user mobility requests per pass	2%	1%	0%	6%	2%	6%
	Average femto ↔ macro mobility events to experienced by a femtocell user per hour	2%	3%	0%	6%	2%	6%

Table 4: Metric to eTOM flow Network Centric E2E Business Process

The mapping of the eTOM process flows to the OPEX categories is presented in Table 5.

	Non-process costs	Marketing	IT (excl. Billing)	Finance and mgmt.	HR, benefits, others	Sales	Customer service	Billing (Order to cash)	Network and maintenance
Request to Answer	0%	3%	1%	1%	5%	14%	17%	0%	1%
Order to Payment	0%	0%	5%	1%	1%	0%	5%	11%	1%
Usage to Payment	0%	0%	5%	2%	2%	0%	5%	21%	1%
Request to Change	0%	0%	5%	1%	2%	1%	6%	0%	3%
Termination to Confirmation	0%	0%	3%	2%	2%	0%	7%	2%	2%
Problem to Solution	0%	2%	3%	1%	8%	2%	17%	0%	9%
Complaint to Solution	0%	2%	3%	1%	9%	2%	17%	1%	2%
Production Order to Acceptance	0%	1%	4%	1%	4%	2%	0%	0%	8%
Trouble Ticket to Solution	0%	0%	3%	0%	5%	2%	5%	2%	14%
Usage to Usage Data	1%	0%	2%	0%	2%	0%	2%	10%	6%
Capacity Management	1%	0%	0%	1%	3%	0%	10%	0%	25%
Service Lifecycle Management	0%	3%	2%	1%	3%	3%	2%	2%	10%
Resource Lifecycle Management	0%	3%	1%	2%	5%	3%	2%	2%	19%

**Table 5: Mapping of eTOM process flows to OPEX categories**

Finally, the mapping of QFD dimensions is displayed in Table 6.

	Non-process costs	Marketing	IT (excl. Billing)	Finance and mgmt.	HR, benefits, others	Sales	Customer service	Billing (Order to cash)	Network and maintenance
OD1 New services and revenues	0%	11%	1%	0%	5%	11%	5%	1%	5%
OD2 Cost of adoption	0%	0%	6%	0%	3%	4%	4%	1%	13%
OD3 Energy	3%	0%	1%	0%	4%	1%	2%	0%	9%
OD4 Operation and maintenance personnel	1%	0%	5%	0%	12%	3%	8%	0%	33%
OD5-14 General autonomies impact	0%	0%	5%	0%	9%	2%	13%	0%	28%
OD15 Visits / transportation	1%	4%	0%	1%	8%	8%	6%	0%	26%

**Table 6: Mapping of QFD dimensions to OPEX categories**

Since the results from Table 3 to Table 6 are considered intermediary results, used for calculating the subsequent tables but without much value on their own, we will not extensively discuss the results of these tables.

Finally, we need a table that specifies the different components of the OPEX of a typical network operator, as well as their relative sizes. Several overviews of cost categories for OPEX for telecom operators exist, for example by Verbrugge *et al.* [27], which is extended by Cid *et al.* [28]. While the cost categories proposed are very suitable for modelling, there is no information of the relative sizes of these categories for a typical network operator, be it fixed or mobile.

Several sources provide alternative OPEX categories while also providing information on the relative sizes of the categories, e.g., [29][30][31][32][33][34]. In this work, the categories of a 2011 white paper of Deloitte [35] will be used, since it is sufficiently fine-grained, it focuses on wireless carriers only, and it is the most recent of the studies. Where comparisons are possible, the results of this study seem coherent with the results of the others. Moreover, for most categories the study provides ranges rather than possibly inaccurate fixed values. Its cost categories, with the corresponding relative sizes, are presented in Table 7.

OPEX group	OPEX category	Minimum	Maximum
Non-process costs	e.g., interconnection fees, taxes, CPE, and uncollectible items.	35%	35%
Support-process costs	Marketing	10%	12%
	IT (excluding billing)	4%	6%
	Finance and management	3%	5%
	HR, benefits and others	3%	5%
Operational-process costs	Sales	18%	20%
	Customer service	10%	12%
	Billing (Order to cash)	2%	4%
	Network and maintenance costs	15%	17%
<i>Sum</i>		<i>100%</i>	<i>116%</i>

**Table 7: Relative sizes of OPEX categories**

When summing up the minimum values from the range, one arrives at a total OPEX of 100%. This is not workable, since it means that for one category to be larger than the minimum, one other needs to drop below the minimum. Therefore, we have taken the middle of the range (108%) and normalized over that. That is presented in Table 8.

OPEX group	OPEX category	Minimum	Maximum
Non-process costs	e.g., interconnection fees, taxes, CPE, and uncollectible items.	32%	32%
Support-process costs	Marketing	9%	11%
	IT (excluding billing)	4%	6%
	Finance and management	3%	5%
	HR, benefits and others	3%	5%
Operational-process costs	Sales	17%	19%
	Customer service	9%	11%
	Billing (Order to cash)	2%	4%
	Network and maintenance costs	14%	16%
<i>Sum</i>		<i>93%</i>	<i>107%</i>

**Table 8: Corrected relative sizes for OPEX categories**

Since we do not expect any impact from NEMs on the categories non-process costs, finance and management and billing, only 56% to 66% of the total OPEX costs can be impacted.

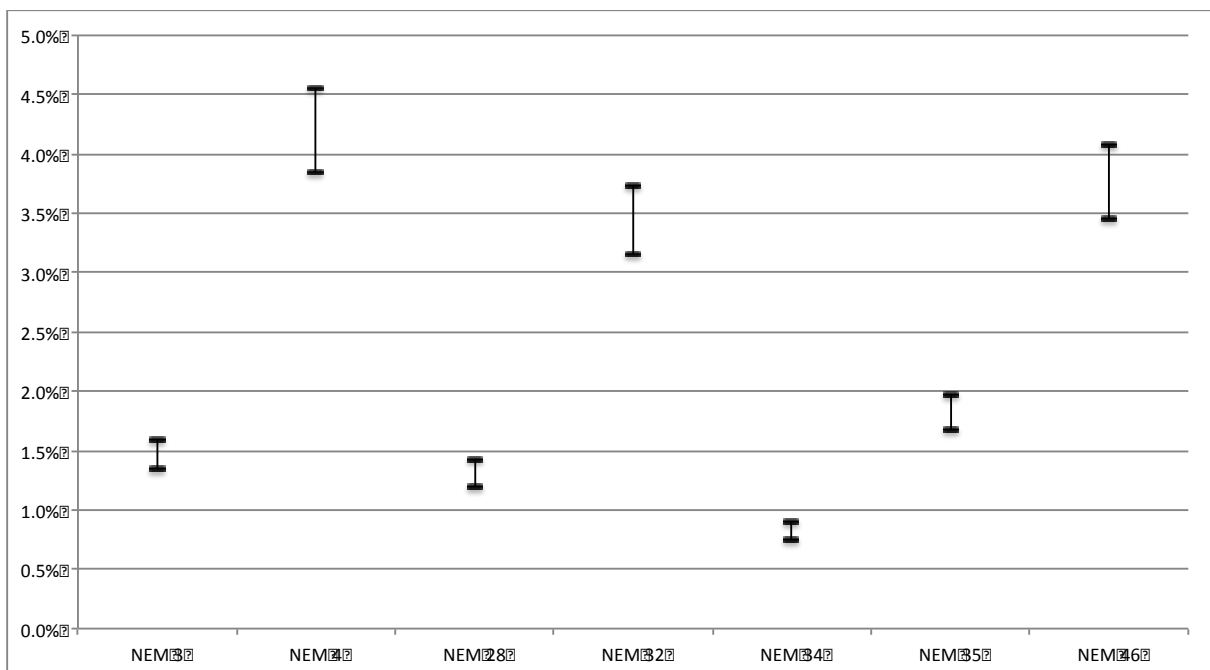
## 3.2 Model results

This section presents the results of the Toy method applied to the selected scenario. In Section 3.2.1, the OPEX impacts for the individual NEMs are presented, while in Section 3.2.2 the individual results are aggregated into a general OPEX impact for the scenario.

### 3.2.1 Individual NEM results

When calculating the OPEX savings according to the proposed methodology and using the tables provided in the previous subsections, one arrives at the results presented in Table 9 and Figure 43.

	Min	Max
NEM 3: Inter-Cell Interference Coordination (ICC)	1,3%	1,6%
NEM 4: Coverage and Capacity Optimization (CCO)	3,8%	4,6%
NEM 28: Predication-based Load Balancing	1,2%	1,4%
NEM 32: Self-healing mechanism for Cell Outage Management (SH)	3,2%	3,7%
NEM 34: LTE Load Balancing	0,8%	0,9%
NEM 35: LTE Tilt Optimization	1,7%	2,0%
NEM 46: GP Coverage	3,5%	4,1%

**Table 9: Results of the Toy Model analysis for predicted OPEX savings by single NEMs****Figure 43: Results of the Toy Model analysis for predicted OPEX savings by single NEMs**

Some NEMs have a low impact of less than one per cent (which can still be significant for an operator), while others go as far as almost five per cent. Normally, an operator will not implement the UMF framework just for executing one NEM instance, so most likely multiple NEM instances will be active at the same time,

orchestrated by the UMF coordination. However, the OPEX impact when running multiple NEMs is not simply a summation of the individual results, in the same way that employing a cleaning staff of two does not result in a twice as clean building with a staff of one. This is because some of the functionalities and benefits might overlap with each other, and there is a maximum to the total gain that can be achieved. The next section provides a potential solution to NEM aggregation.

### 3.2.2 NEM impact aggregation

As explained above, since the exact degradation of benefits is not strictly defined, the following NEM order is assumed: NEMs 1 to n are ordered from the largest maximum gain to the smallest maximum gain, and apply for both the best and worst case the following simple formula:

$$\text{Total OPEX savings} = NEM_1 + NEM_2 \times \frac{1}{2} + \dots + NEM_n \times \frac{1}{n}$$

The ordering is justified by the assumption that an operator will start instantiating the NEMs with the highest expected benefits, and from there one will start looking for incremental improvements.

To clarify the impact of this formula, a simple example is given in Table 10.

	1 NEM	2 NEMs	3 NEMs	4 NEMs	5 NEMs	10 NEMs
Impact of NEM 1	1	1	1	1	1	1
Impact of NEM 2		0.5	0.67	0.75	0.8	0.9
Impact of NEM 3			0.33	0.5	0.6	0.8
Impact of NEM 4				0.25	0.4	0.7
Impact of NEM 5					0.2	0.6
Impact of NEM 6						0.5
Impact of NEM 7						0.4
Impact of NEM 8						0.3
Impact of NEM 9						0.2
Impact of NEM 10						0.1

**Table 10: Examples of impact weights using the proposed formula**

When applying this to the results for this use case, we have seven NEMs in the following order: (1) NEM 4, (2) NEM 46, (3) NEM 32, (4) NEM 35, (5) NEM 3, (6) NEM 28 and (7) NEM 34. The formula for the best case is given as follows:

$$\begin{aligned} \text{Best case OPEX savings} \\ = 4,6 \times \frac{7}{7} + 4,1 \times \frac{6}{7} + 3,7 \times \frac{5}{7} + 2,0 \times \frac{4}{7} + 1,6 \times \frac{3}{7} + 1,4 \times \frac{2}{7} + 0,9 \times \frac{1}{7} \end{aligned}$$

whereas the worst case is analysed below:

$$\begin{aligned} \text{Worst case OPEX savings} \\ = 3,8 \times \frac{7}{7} + 3,5 \times \frac{6}{7} + 3,2 \times \frac{5}{7} + 1,7 \times \frac{4}{7} + 1,3 \times \frac{3}{7} + 1,2 \times \frac{2}{7} + 0,8 \times \frac{1}{7} \end{aligned}$$

resulting in the conclusion that the OPEX saving of use case 4 for a typical operator is expected to fall between 11 and 13 per cent.

The development of these ranges is graphically presented in Figure 44. In this graph, the black lines represent the value calculated by the formulas above. For reference, the grey lines are added, representing the 'simple aggregation' where each additional NEM is considered to have its full autonomous OPEX impact. The solid lines represent the best case, while the dashed lines represent the worst case. Adding additional NEMs will not have a large impact, unless the NEM has a large autonomous gain that will place it at the beginning of the ranking.

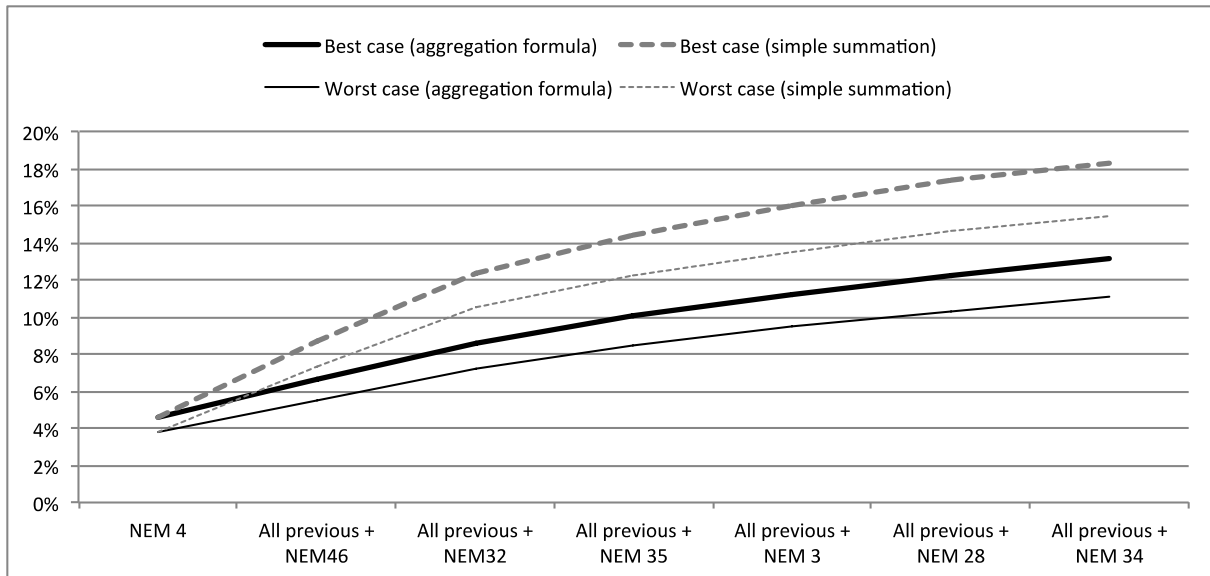


Figure 44: Progression of the accumulated OPEX impact when more NEMs are added

### 3.3 Discussion of results

The Toy Model, as described in the previous subsections, allowed us to arrive at some quantified results on the OPEX impact of autonomics (UMF and NEMs) in scenario 2. However, some positive and negative considerations need to be taken into account regarding the method and its application.

Although a number of market research reports present potential OPEX savings of autonomics, literature does not discuss methods to calculate the OPEX benefits of the implementation of autonomic networking for a typical network operator. This environment is both applied and relying on an unsure future, and is as such more complex than the environments studied by for instance Verbrugge *et al.* [27] and Cid *et al.* [28], that provide static analyses for green-field deployment rather than showing gains. The work proposed in this chapter ventures in new methodological territories for this reason. However, a methodology has been developed that takes into account both expert opinions as well as simulated metrics, uses established frameworks from literature, and that makes conversion steps between those frameworks that are justified.

The model can also yield critical remarks. First of all, our analysis applies to a 'typical' operator, which is a concept that can be discussed. It is unclear whether the typical operator described in the eTOM processes is the same typical operator from Deloitte's OPEX categories. Second, since a typical operator will most likely not exist in the real world, one can expect the UniverSelf framework to be mostly applied by atypical operators, which consequently might alter the experienced OPEX benefits compared to the model.

Second to that, one can have considerations regarding some aspects of the model, e.g., the conceptual nature of some of the requested expert inputs. When experts were asked for mapping tables, it took some effort to get them to understand what was actually asked from them – a similar experience as the exercise in which the QFD inputs were gathered. Especially the mapping from metrics to OPEX process flows was considered 'difficult'.



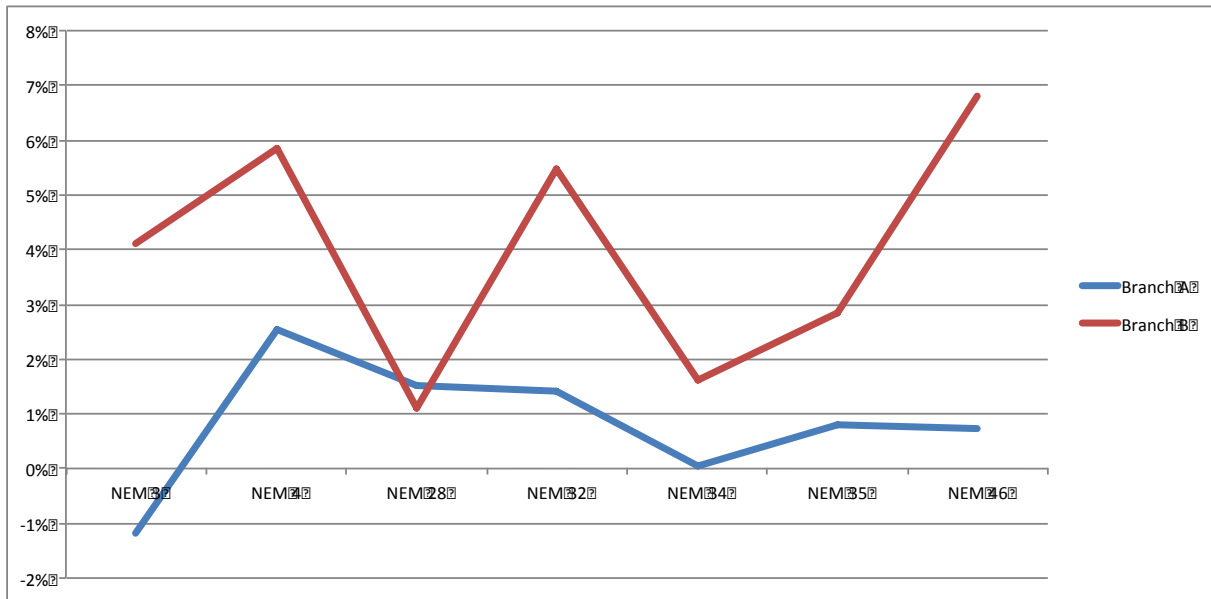


Figure 45: Impacts of the branches A and B on the outcome of the Toy Model

Finally, we can make slight reservations when analysing the output of the model. In Figure 45, the impacts of the branches A and B on the final branch C are shown. It is striking that for NEM 3 (*Inter-Cell Interference Coordination*) the expected OPEX gains according to the Toy Model turn out negative, which means an expected increase of OPEX. This is due to two significant negative metrics that could not be compensated by two modestly positive metrics. The QFD however does pull this NEM to the positive side of the OPEX benefits.

One conclusion from this graph is that both lines seem to follow a similar pattern, with an exception for NEM 28, *Prediction-based Load Balancing*. This indicates that both branches have a shared vision on which NEMs will yield large OPEX gains and which will not. The figure also holds another conclusion: with the exception of, again, NEM 28, the impact of branch A is significantly smaller than that of branch B. This can be criticised, since the purpose of the Toy Model was to assess the impact of the NEMs and their metrics in another way than the *a priori* QFD analysis; however, in this exercise, the QFD seems to still have had a large voice in the outcome.

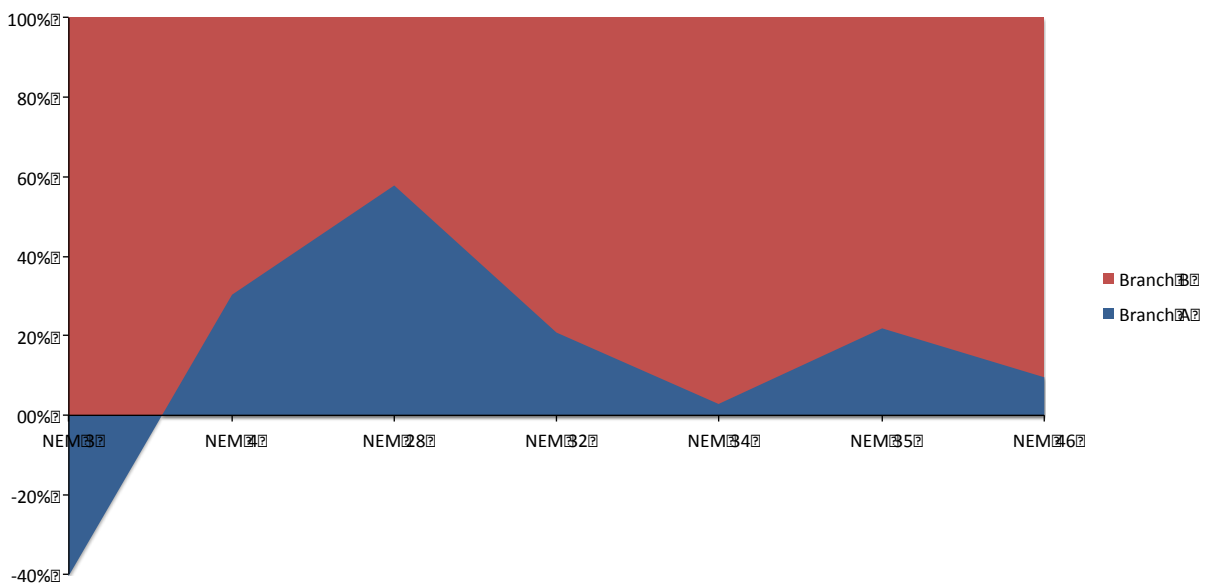


Figure 46: Relative impacts of branch A and B on the outcome of the Toy Model

This point is further explored in Figure 46, which shows the relative impacts between the two branches. Especially NEM 34, *LTE Load Balancing*, comes out very low on the side of the metric branch. This can be explained by the fact that this NEM only provided one metric – one on which it on paper did not even excel greatly. This raises issues with the subjectivity of the NEM metrics and their implementations by the NEM designers. The comparisons between the NEMs performance and the base cases are not always uniform, and moreover, there seems to be an indication that more metrics provided by a NEM designer will favour that particular NEM in terms of expected OPEX benefits.

### 3.4 Conclusions

The Toy Model performs an OPEX impact expectation analysis using simulation results as well as expert inputs. Based on the metric results provided by NEM developers, it assesses the impact on eTOM processes. Together with the expert evaluations from a QFD exercise, this allows for an assessment of the impact on the different OPEX categories, and subsequently on the total OPEX for a typical network operator. The results are presented as ranges instead of single figures. In the case of Scenario 2, the NEM with the highest results is expected, in isolation and with the UMF active, to deliver between 3.8 and 4.6 per cent OPEX savings typically. In addition, a formula is proposed to assess the total impact of all NEMs of the Scenario together with the UMF. This formula starts from the most promising NEM, and adds new ones in decreasing order of expected impact. With all NEMs in Scenario 2 active, the predicted OPEX saving for a typical network operator will fall between 11 and 13 per cent.

## 4 UMF deployment in existing network infrastructures

Three scenarios of UMF deployment in existing network infrastructures are presented in this section. The presentation consists of a concise introduction of the selected infrastructures and analysis of deployment issues. Furthermore, it includes principles/factors that can be taken into consideration during the stages of the deployment process and challenges for future work.

### 4.1 Deployment Study of UMF in Metro Ethernet Service

This section studies the deployment of UMF in metro Ethernet service. To begin with, we introduce a target metro Ethernet service in brief, after which we discuss the current network management system and the arising problems. When trying to deploy a new technology to a commercial system, we have to take care of more factors such as cost and system stability than just considering the technical issues. In this section, some basic principles of deploying new technologies to carriers' commercial systems are listed as an example, which to some extent decide the UMF's deployment scenario in metro Ethernet service. Finally, we conclude this section with some future work.

#### 4.1.1 Introduction to Target Metro Ethernet Service

The target metro Ethernet service utilizes Next Generation Network (NGN) technology to deliver much higher quality, higher reliability than ordinary broadband Ethernet services. A service image is illustrated in Figure 47. With the service, a broadband network that spans multiple domains can be provided with a single one-stop service that covers all steps from application to development, maintenance and invoicing. Moreover, connecting to networks of participating carriers means a seamless network can be developed that spans the entire country.

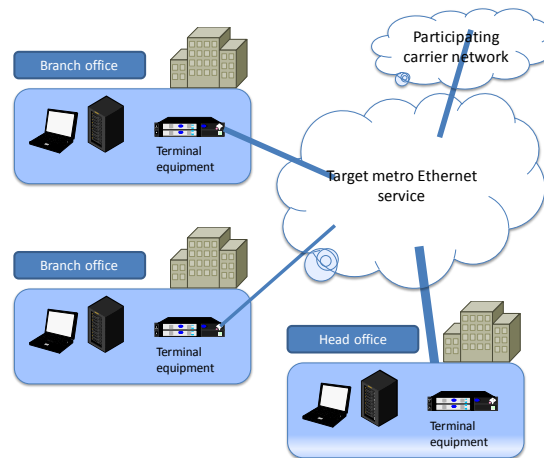


Figure 47: A service image of the target metro Ethernet service

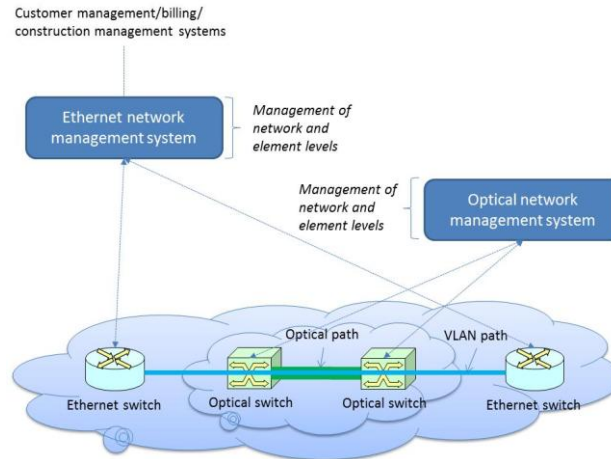
In fact, the target metro Ethernet service is a closed network equivalent to a dedicated service. Virtual LAN (VLAN) technologies are employed to provide ample security for each customer, so there is no need to worry about unauthorized access. In addition, the service utilizes Ethernet Operation Administration and Maintenance (OAM) technology (for checking communication with monitoring frames) to constantly monitor each line. In the event that there is a malfunction, the maintenance centre quickly sends a notification of the malfunction, resulting in reliability that is so much higher than ordinary Ethernet services.

This type of service is usually offered to enterprises, and therefore it is delivered with a high level availability SLA, in the areas such as failure recovery time, delay time and uptime.

#### 4.1.2 Current Network Management System

In order to offer the Ethernet service with high quality and high reliability, the network infrastructure and the management system can be extremely complex as illustrated in Figure 48. In fact, the network spans multiple layers including the optical layer and the Ethernet layer. In order to manage all the network devices, at least

one network management system is necessary in each layer. As illustrated in the figure, the management of network level and element level is closely integrated as a whole in both layers in the current design. In addition, a large amount of Operations Support Systems (OSSs) such as customer management system, billing system, and construction management system are actually deployed.



**Figure 48: Infrastructure and management system of the target metro Ethernet service**

We choose the Ethernet network management system and optical network management system as the deployment objectives in this study. However, we believe that it is just the beginning and UMF has the potential to be deployed in all the other management systems.

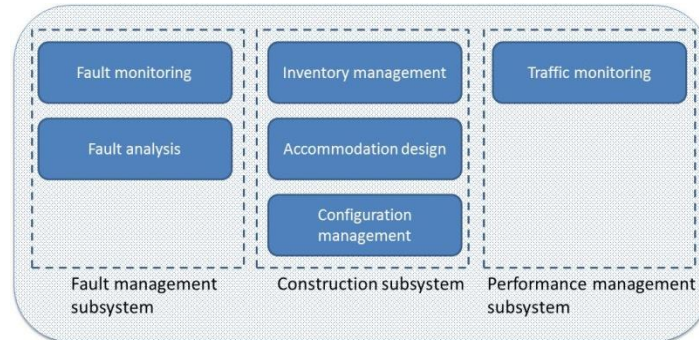
#### 4.1.2.1 Design of Ethernet Network Management System

The whole structure of the current Ethernet network management system is illustrated in Figure 49, which consists of three subsystems: fault management subsystem, construction subsystem and performance management subsystem.

The fault management subsystem is composed of function blocks such as fault monitoring and fault analysis. In the function block of fault monitoring, the network operators can monitor the real-time status of an interface, a component of a device such as a CPU core and memory through obtaining the syslog and the Management Information Base (MIB) information. Besides, through setting Simple Network Management Protocol (SNMP) traps, alarm information will be sent to the network operators immediately. In the meantime, the OAM functions such as ping and traceroute are used in order to realize the end-to-end path monitoring. When a failure is detected, the VLAN path will automatically switch to the backup path so that there will be no impact on the running service. Since the optical paths in the network are currently static, the restoration is limited in only Ethernet layer. The fault analysis is designed to be performed automatically to some extent in the current system in which when a failure occurs in a device, the identification of the impacted user or service is realized. However, on the other hand, when a service down is detected, the detailed failure point should be searched manually, and in the case of some special silent failures, the identification can be extremely difficult and time-consuming, greatly influencing the quality of the service.

The construction subsystem can be further divided into function blocks such as inventory management, accommodation design and configuration management. The inventory management function block allows a network operator to maintain up-to-date records about the number, type and status of all devices on the network. Currently, all the records are registered manually by network operators in advance and all the changes to these records are conducted manually as well. In the accommodation design function block, a network operator designs how a newly required VLAN path is accommodated to the current physical links from the prospective of resource optimization. Here, the physical links are actually optical paths in the lower layer, which are established by the optical network management system. In fact, the accommodation design is also made by hand and the network operators make the decision only based on the knowledge of Ethernet layer. Finally, through the configuration management function block, the network operators input commands to all the related devices in order to setup the necessary VLAN paths. In addition, the input of OAM commands is also the scope of this function block.

The performance management subsystem mainly performs traffic monitoring. As the name suggests, the traffic monitoring function block detects the real-time traffic volume in each link and reports the results to the inventory management function block so that the accommodation design can be performed based on the latest information.



**Figure 49: Whole structure of the current Ethernet network management system**

Table 11 summarizes the current deployment status of autonomic functions.

Possible autonomic functions	Deployment status	Related function blocks	Remark
Auto-inventory	×	Inventory management	
Auto-provisioning	Δ	Accommodation design	Automation deployed only in the configuration management
		Configuration management	
Auto-fault-analysis	Δ	Fault analysis	Limited functions within the Ethernet layer

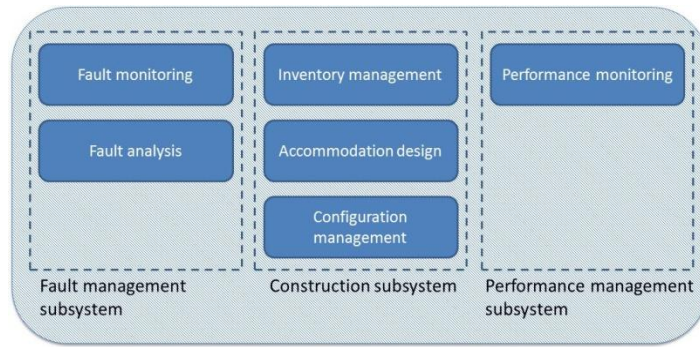
**Table 11: Deployment status of autonomic functions in the current Ethernet network management system**

×The cross mark for the deployment status means that the autonomic functions are not deployed at all in the current system, while the triangle mark represents that autonomic functions are deployed with limited functions.

#### 4.1.2.1 Design of Optical Network Management System

The whole structure of the current optical network management system is illustrated in Figure 50, which shares the similar design with the Ethernet management system. In fact, the managed objects change only from VLAN paths to optical paths, so we only focus on the difference.

First, the layer-1-specific MIB and OAM are used instead and moreover, different fault analysis scheme is utilized in the fault management subsystem. Due to the layer-1-specific restrictions, in the construction subsystem, more factors such as lambda transformation and power level should be considered when the network operators perform the accommodation design. In the same way, the accommodation design is currently manual and limited in the optical layer. Finally, as to the performance monitoring, performance-related parameters such as power level, noise are monitored instead of the traffic volume monitored in the Ethernet network management system.



**Figure 50: Whole structure of the current optical network management system**

Similarly, we summarize the current deployment status of autonomic functions in the optical network management system in Table 12.

Possible autonomic functions	Deployment status	Related function blocks	Remark
Auto-inventory	×	Inventory management	
Auto-provisioning	×	Accommodation design	
		Configuration management	
Auto-fault-analysis	△	Fault analysis	Limited functions within the optical layer

**Table 12: Deployment situation of autonomic functions in the current optical network management system**

The cross mark for the deployment status means that the autonomic functions are not deployed at all in the current system, while the triangle mark represents that autonomic functions are deployed with limited functions.

#### 4.1.2.2 Discussion

Based on the description above, we can conclude the current problems in the target systems as below.

- 1) The autonomic functions are only limitedly deployed (ex. in fault analysis).
- 2) The management of the Ethernet network and the optical network is totally separated.

Thanks to UMF, we positively believe that autonomic functions have great potential to be deployed in the target system and moreover, it promotes the integration of management of both the Ethernet network and the optical network, which is expected to accelerate resource optimization and system efficiency.

#### 4.1.3 Basic Principles of Deploying New Technologies

Nowadays carriers have already had large, expensive and mature network management systems, so in fact it is difficult to deploy new technologies to the current systems. Therefore, when we consider the deployment of UMF, the following two key principles should be well considered before making any decisions.

*Principle I: Do not make any great modifications to the existing systems.*

*Principle II: It is better to wait until the next system alternation including network equipment.*

One of the promising candidate solutions for the next system alternation is Software defined Networking (SdN), so we believe that it should be a good chance for UMF to be deployed if it can be well aligned to the SdN architecture. Since the detail of the alignment of UMF and SdN will be discussed later in Section 5.1, in this



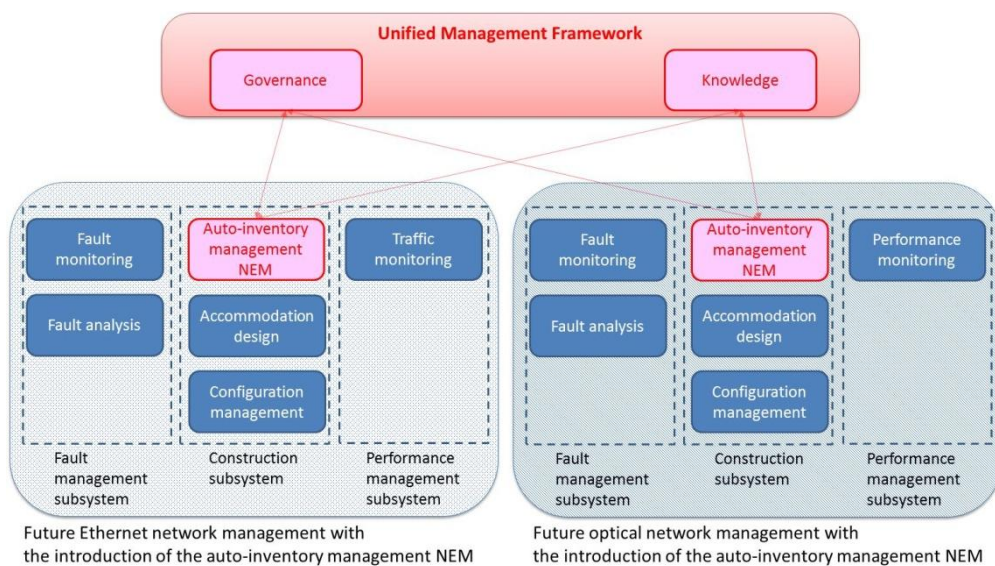
section, we focus on Principle I. Here we propose an iterative deployment scenario as discussed in detail in the following.

#### 4.1.4 UMF's Deployment Scenario

Based on the discussion above, we study UMF's deployment scenario. We think that UMF and NEMs can be deployed in the target system step by step as illustrated below.

##### Step I: Deployment of Auto-Inventory Function

Because the inventory management is regarded as one of the most costly and unintelligent parts in the current network management systems, occupying a large amount of an operator's working time, we plan to deploy the auto-inventory function to both the Ethernet network management system and the optical network management system in the first place as presented in Figure 51. In the figure, the newly added functions are illustrated in red.



**Figure 51: Deployment of auto-inventory function**

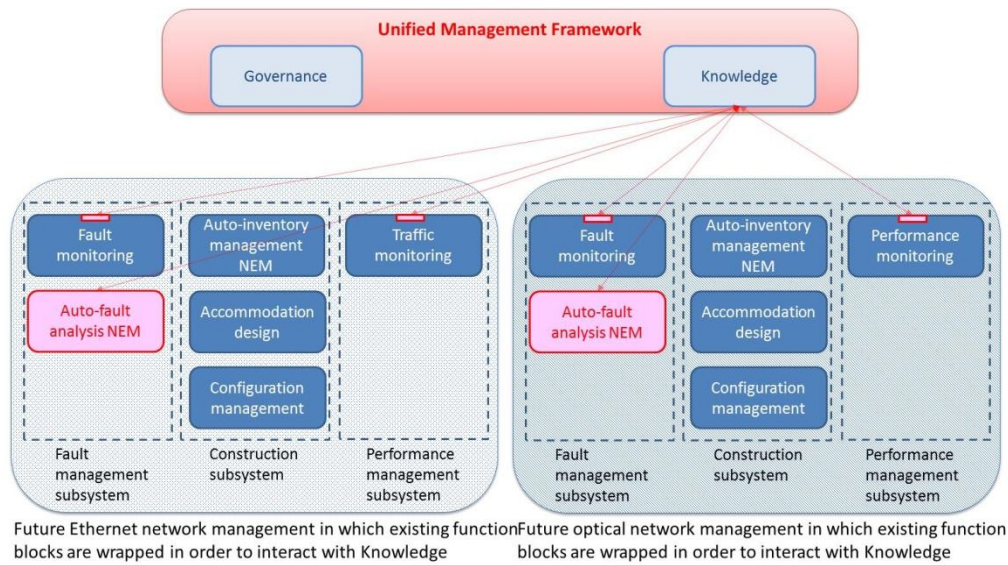
The detailed deployment method is shown below:

- 1) Replace the existing manual inventory management function block with a suitable auto-inventory management NEM in both the Ethernet network management system and the optical network management system.
- 2) The Governance function block is deployed along with the deployment of the above to allow their control.
- 3) We further deploy the Knowledge function block in UMF in order to realize the inter-layer information sharing, especially the topology information. In this way, for example, any modifications in the allocation of optical paths will be fed back to the Ethernet layer automatically.

##### Step II: Deployment of Auto-Cross-Layer Fault Analysis

In the current system, the fault analysis is limited within the single layer and it is not totally automatic, which complicates the identification of the failure point. In Step II, we plan to add new fault analysis NEMs and fully utilize the Knowledge function block to exchange the information across both the Ethernet layer and the optical layer; this is shown in Figure 52.





**Figure 52: Deployment of auto-cross-layer fault analysis function**

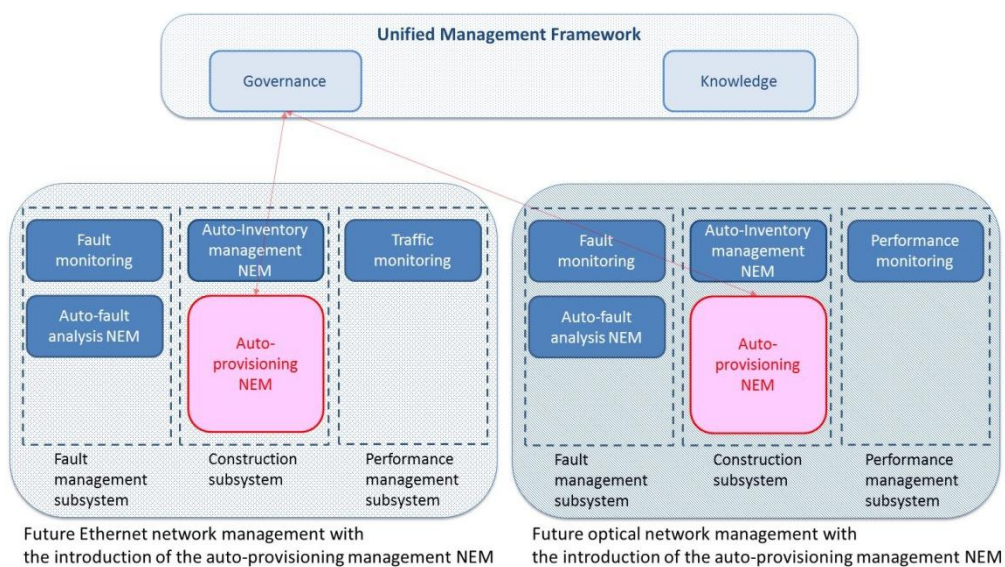
The detailed deployment method is shown below.

- 1) Replace the existing fault analysis function block with a suitable auto-fault analysis NEM in both the Ethernet network management system and the optical network management system.
- 2) We wrap the existing function blocks such as the fault monitoring, traffic monitoring and performance monitoring with new interfaces so that these existing function blocks can exchange information through the Knowledge function block.

In this way, functions such as auto-cross-layer fault analysis can be realized.

### Step III: Deployment of Auto-Provisioning Function

Since currently the accommodation design is mainly performed manually, we can deploy the suitable auto-provisioning NEMs to both the Ethernet network management system and the optical network management system and these NEMs are controlled by the Governance function block as showed in Figure 53.



**Figure 53: Deployment of auto-provisioning function.**

The detailed deployment method is showed in the following.

- 1) *Replace the existing accommodation design and configuration management function block with a suitable auto-provisioning management NEM in both the Ethernet network management system and the optical network management system.*

#### Step IV: Deployment of Cross-Layer Auto-Provisioning Function

As discussed in the previous part, until Step III, the provisioning of the Ethernet network and the optical network are totally separated. In other words, in the case of link failure or sudden traffic increasing for instance, the re-allocation of VLAN paths and the optical paths are performed separately. Due to the lack of inter-layer interaction, obviously it is not an ideal solution to reach the resource optimization [36].

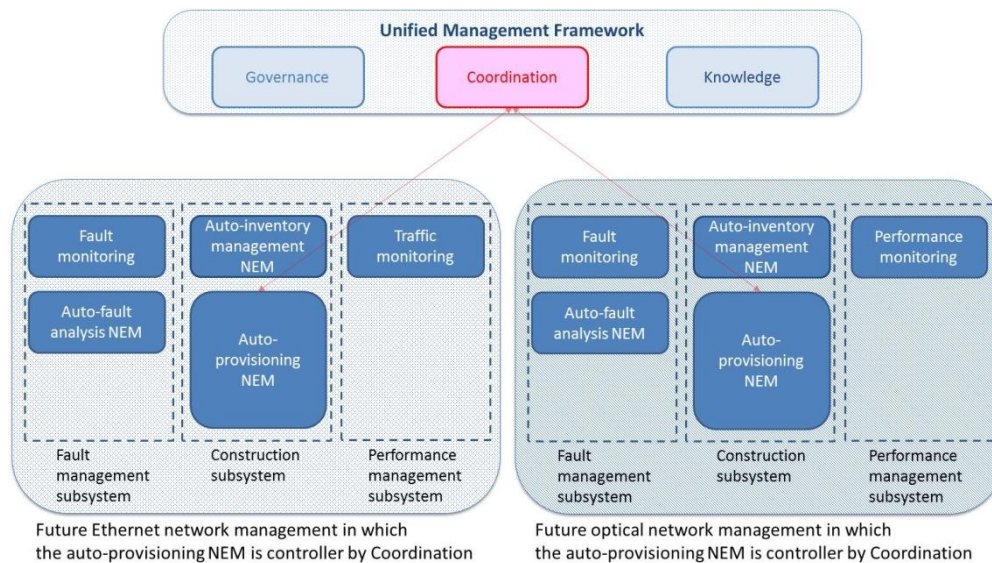
Therefore, in Step IV, we newly add the Coordination function block in UMF in order to deploy the cross-layer auto-provisioning function as shown in Figure 54.

The detailed deployment method is as below.

- 1) *We further deploy the Coordination function block in UMF in order to realize the inter-layer interaction. For example, when a new service order (SO) request is received, the provisioning function in both the Ethernet layer and the optical layer will interact in a coordinated way.*

As result, that only a VLAN path should be established or that the establishment of the VLAN path should be accompanied by the establishment of a new optical path is decided from the prospective of the total resource optimization.

In the meantime, with the successfully deployment of cross-layer auto-provisioning function, functions such as auto-recovery and auto-resource reallocation can be realized as well.



**Figure 54: Deployment of cross-layer auto-provisioning function.**

#### Completely Deployed Image

Finally, fully utilizing the features of UMF and NEMs, the completely deployed image is illustrated in Figure 55, in which the auto-inventory, auto-provisioning (including auto-recovery and auto-resource allocation) and auto-fault analysis are expected to be realized.



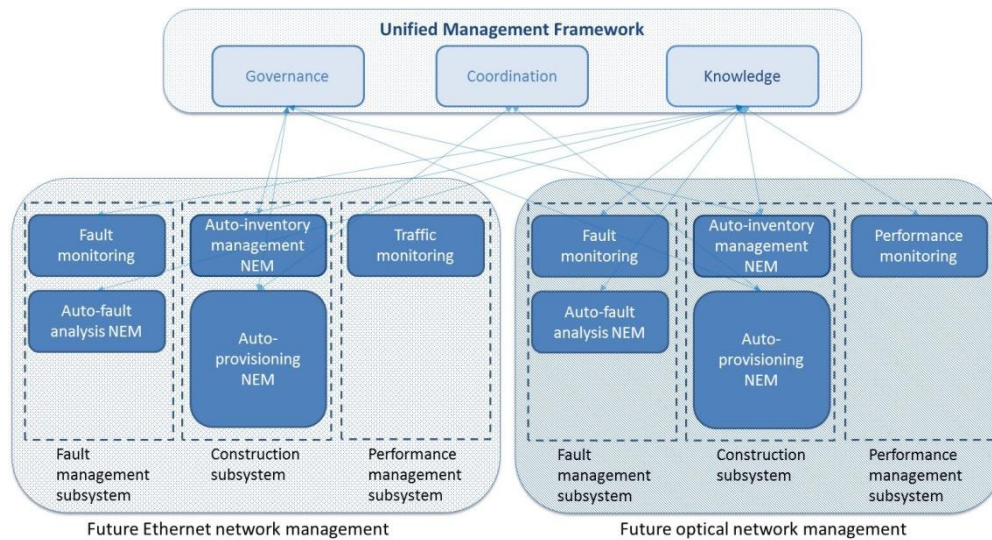


Figure 55: Completely deployed image.

## 4.2 Deployment of UMF in FTTH network

The combination of the ever growing bandwidth demand, the residential and business demand for a quick network access, the need to support high-reliability services and the increasing pressure from competitors create significant challenges for telecom network operators. To keep up with the changing trends in this highly competitive environment, telecom operators are evolving into full-service communication providers delivering high-speed connectivity and value-added services. At the same time, telecom operators have to trigger cost cutting and efficiency gains if they are to maintain a competitive edge.

An emerging technology for helping to reduce the cost and amount of infrastructure required for mass market broadband deployment is GPON, or Gigabit Passive Optical Network, a point-to-multipoint technology for bringing fibre to the premises. The big challenge for this new technology to be used on a large scale basis, is to maintain or reduce the cost of service delivery and maintenance, and to be able to maximize the customer's quality of experience; therefore it is imperative for operators to consider automated provisioning solutions, ensuring that the service is provisioned in an effective and efficient manner, while reducing the error rate attributed to manual installation processes.

This section studies the deployment of UMF in a typical FTTH network. First, the GPON architecture is introduced, and then the typical management systems and processes are described. Finally, an analysis of how UMF could be deployed on an FTTH network is presented.

### 4.2.1 GPON Architecture

Optical access networks can be initially classified into two different categories depending on whether they are based on Point-to-Point (PtP) or Point-to-Multipoint (PtMP) links. In the PtP scheme one fibre is used to connect an Optical Network Termination/Unit (ONT/ONU) to an Access Node (AN) located at the operator's premises, but in the PtMP mode a single fibre is used to connect several ONTs to the Optical Line Termination (OLT), which is generally situated within a Central Office. Usually, PtMP architectures that do not use any active equipment in the Outside Plant (OSP) are called Passive Optical Networks (PONs), while PtP schemes are known as Active Optical Networks (AONs).

Beyond having more than one ONT connected through one fibre to an OLT port, there are also different PON architectures depending on how far from customer the fibre termination is. Thus, the acronym FTTx (Fiber-to-the-x) intends to group all the different possibilities that can be found (FTTx, x = H for home, B for building, C for curb and N for node) on current deployments. Each PON scheme is therefore suitable for a concrete scenario, and except for the FTTH case will require a complementary technology (usually Ethernet or xDSL) to reach customer's premises.

Independently of what kind of FTTx architecture is used, PtMP-PON deployments are based on a general architecture that includes one or more Remote nodes (RN) in the OSP. Those nodes are the starting point of

different optical paths which at their ends are connected to the ONTs. Hence, it is said that PON networks present a tree topology where the OLT is the root of the trunk and RNs are the origin of the different branches. Usually RNs are power splitters, which provide the same optical signal at their outputs but dividing the available power, but in the last years also Array Waveguide Gratings (AWGs), which are able to multiplex and demultiplex different wavelengths, have been deployed.

It is important to remark that in fibre PtMP topologies a single fibre is shared among different ONTs, therefore a mechanism to share the physical medium and to identify which ONT is assigned to each customer is needed. The GPON standard is issued by the ITU under ITU G.984 [37] with a line rate of 2.5 Gbps and 1.25 Gbps in the downstream and upstream directions, respectively, using the 1490 and 1310 nm wavelengths. The GPON data packets are sent time-division multiplexed (TDM) onto the network and the total available bandwidth is shared following a dynamic bandwidth assignment (DBA) mechanism.

All transmissions in a GPON system are performed between the OLT and the ONTs via passive power splitters, connecting the optical access network to the metro network while the ONT provides the interface between the customer's home network and the PON. Figure 56 shows a scheme of a typical FTTH access network.

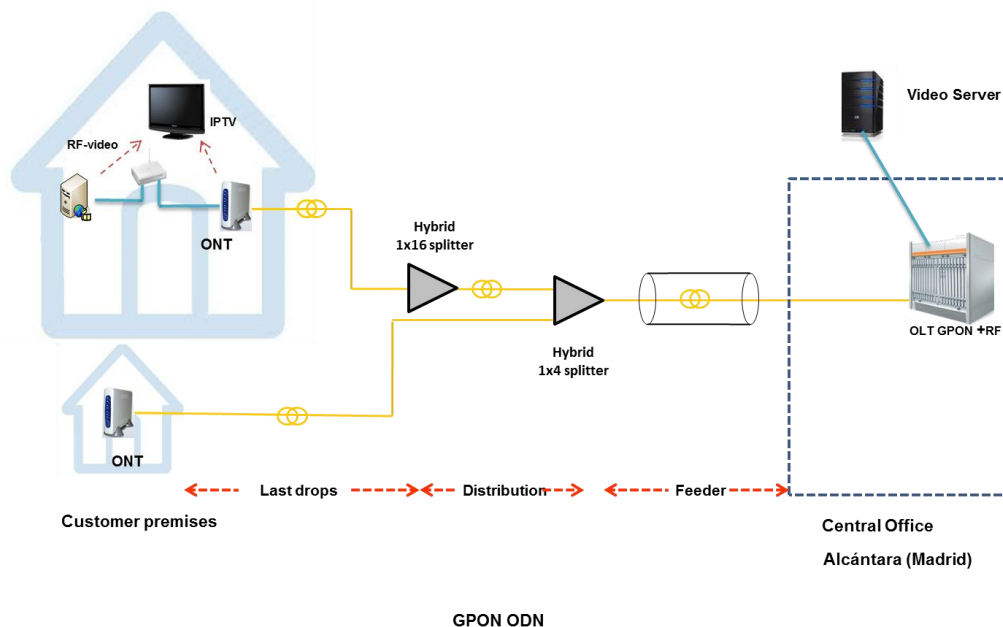
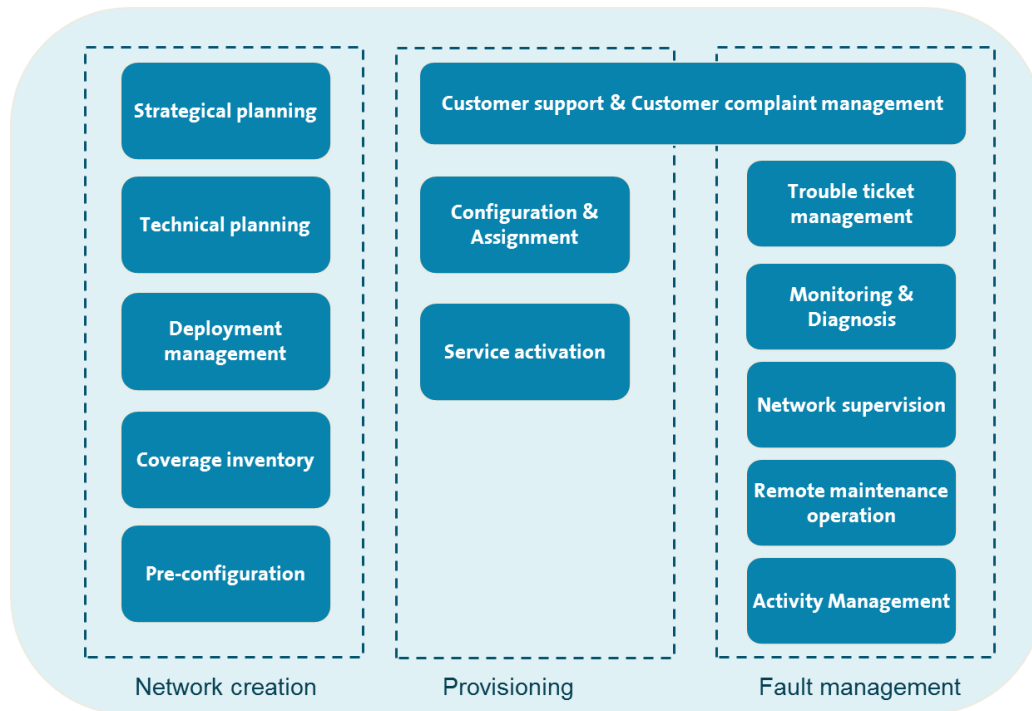


Figure 56: Network architecture for FTTH access networks

Different OSP topologies can be used in GPON networks to accommodate different deployment scenarios, for example different splitting factors 1:32 (17.5 dB) or 1:64 (21 dB) with different splitting stages (e.g. 1:8 + 1:8, 1:4 + 1:16, 1:16 + 1:4) can be used for “Low-raise” or “High-raise” buildings such that the network is deployed more efficiently and is more robust to bandwidth growth. The OSP topology is transparent (or almost transparent) for the GPON protocols used to communicate both OLT and ONT ends where the main restriction is the amount of optical losses that can be tolerated in the GPON network.

#### 4.2.2 FTTH-related Support Systems

This section covers the network creation, provisioning and maintenance aspects of a FTTH network infrastructure. Whilst each FTTH network design will differ and operate in different environments and conditions, the planning, operation and maintenance remains a common requirement to all. Figure 57 presents a simplified view of the Operation Support Systems involved in the FTTH lifecycle.



**Figure 57: Simplified view of the OSS involved in FTTH lifecycle**

FTTH deployment involves the process of building the passive network from the Central Office towards a cabinet in the proximity of subscriber homes. This is also referred as the horizontal deployment, which results in the coverage, via optical fibre, of a certain number of homes passed. Given the important investments needed to build a new network, one of the most important parts of the process is the selection of the strategy to follow in the selection of areas where FTTH will be deployed first. The geographical distribution of subscribers, the potential demand of bandwidth-hungry services, construction costs, possible environmental limitations, the availability of expert personnel and the presence of competitors in the area are some of the factors that the strategic planning phase takes into account.

Once the decision has been taken of deploying FTTH in a given neighbourhood, the operator needs to make a detailed technical plan for that particular area. The ultimate goal is to adapt the cable deployment to the infrastructures in the town, ensuring little or no disturbance to the general public and surrounding environment. This phase should not be underestimated, since the civil work often involves underground duct systems or installations on sideways or poles. Liaison with local authorities is required and should be arranged prior to the physical work.

The bulk of the work and costs in the deployment process itself is the digging, connectorization, splicing and characterization of the fibre in the access segment, followed by passive infrastructure costs (fibres and splitters). As part of the network construction, the inventory systems are manually updated. Given the lack of a unified inventory system, this operation usually implies the update of different databases in different Operation and Business Support Systems (OSS and BSS, respectively). Finally, the equipment at the Central Office (namely OLT, probes and fibre cables) are installed and pre-configured. The pre-configuration phase includes the definition of service profiles in the OLT, corresponding to the different services to be offered to customers. Once deployment is finished, everything is in place for subscriber activation.

FTTH service provisioning phase is triggered when the customer contracts the new service through the commercial channels, e.g. customer call center or web page. Then a dedicated port in the OLT is assigned to the customer and configured with the profile of the contracted service. The physical work involves installing a fibre that connects the horizontal deployment to the subscriber home (this is referred as vertical deployment), and then setting up the internal home devices and connectivity to activate the service. Again, the bulk of the costs in this phase are the physical deployment and verification work, and the cost of the active devices in the home. Configuration of home devices (router, ONT and set-top box) may also require manual technician work. Once the service is activated, the customer can start using it.

FTTH supervision and maintenance is the set of operation processes required to guarantee the availability and quality of the FTTH service once the connection has been activated. As in any other network segment and technology, it involves monitoring the service status, responding to customer complaints, troubleshooting problems and restoring the service when needed:

- Customer complaint management: the customer reports a service problem, typically by phone, smartphone app or web page. The context and symptoms of the problem are registered in a trouble ticketing system. Information may be provided to the customer about possible solutions or estimated downtime.
- Trouble ticket management: any problem derived from a customer complaint triggers the creation of a Trouble Ticket (TT). The TT will be updated with the information related to the problem: human operators that have supervised the problem, tests performed to diagnose the problem and their results, actions triggered to repair the issue, and closing information.
- Monitoring and diagnosis: a set of tests are performed to diagnose the cause of the problem. These processes might be automated for simple problems, or require the involvement of a NOC (Network Operation Center) technician. In the case of FTTH access networks, the tests are mainly manually triggered, and the diagnosis is based on the expertise of the human operator.
- Network Supervision: the Network Operation Center periodically monitors the status of the network elements, with the goal of detecting issues before customers detect and report a problem. This is done by collecting event notifications from the network and periodic sampling and testing. Advanced mechanisms involve aggregating, filtering, correlating, summarizing and presenting alarms received from the network. The root cause of service affecting alarms is identified as part of the process, either automatically or with manual tests.
- Remote maintenance operations: Once the root cause of failure has been identified, corrective actions need to be executed in order to restore the service. In some cases, these actions may be triggered remotely (e.g. issuing a reconfiguration command on a network element). These maintenance operations include those performed periodically trying to anticipate problems in network elements.
- Activity management: when the remote operation is not possible, the maintenance operations need the involvement of a technician for physically repair the problem either at the Central Office, at the customer premises or on the outside plant

There are multiple sources of operating expenses associated to FTTH maintenance, from customer care to problem diagnosis and on-the-field technician works. The diversity of problems that may impact the FTTH service, and the fact that some of them are not under the direct control of the operator (for instance, problems in the home network), makes supervision and maintenance a complex and costly process.

### 4.2.3 UMF deployment on FTTH Networks

This section elaborates on how the Unified Management Framework can be deployed on FTTH access networks. It does not aim to sketch a migration process, but to provide a final picture of the possible deployment of UMF core blocks and NEMs on FTTH networks, the functionalities each of them assumes and the advantages with respect to current network operation. Figure 58 presents this final view of UMF deployment on FTTH access networks.

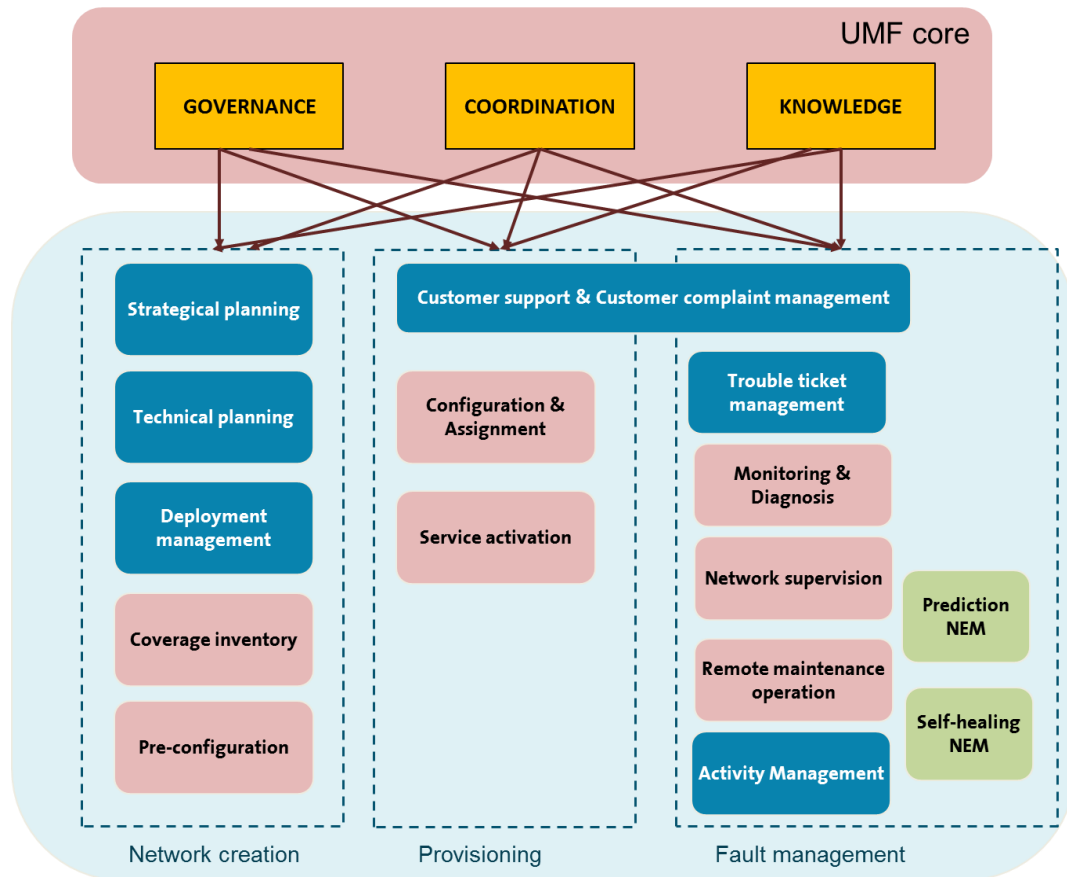


Figure 58: UMF deployment on FTTH access networks

The deployment of NEMs for the management of the different network elements in the FTTH access networks allows the evolution of the systems and the emerging of new possibilities. In the figure above, boxes in pink represent those functionalities that can be embedded into NEMs and UMF core, and boxes in green represent new functionalities that the deployment of the appropriate NEMs can add to the network operation. The proposal is to deploy the UMF core block and a set of NEMs implementing specific functions: inventory updating, configuration, monitoring, diagnosis, prediction and self-healing.

In the network creation phase, the deployment of an inventory NEM can keep an automatically up-to-date inventory of the existing network elements, avoiding thus manual intervention. This NEM is managed by the GOV core block, and the information it produces is fed into the KNOW block, making it available to the other NEMs and UMF blocks. Even when it was not explicitly depicted in Figure 57, each of the current OSS keeps its own inventory, not always synchronized with the inventories in the other systems. The deployment of UMF presents the advantage of keeping a unique inventory for the whole network, always up-to-date thanks to the NEMs deployed onto the network elements. The pre-configuration of the equipment at the Central Office can also be accomplished defining the appropriate high-level objectives in the GOV block, which will be automatically translated to NEM policies, and enforced onto the corresponding NEMs. The same approach describes the automation of the configuration and activation during the provisioning phase.

During the fault management phase, the deployment of monitoring and diagnosis NEMs allows the gathering of monitoring data during the runtime of the service, and its storage in the KNOW block. A diagnosis NEM, as the one implemented and deployed in Use Case 7 [38], finds the root cause of failure in case of malfunctioning of the service. The fact that all the information is stored in the KNOW block, together with the Supervision operation of the GOV block, presents to the human operator advanced network and service supervision capabilities. The remote operation is now achieved through the definition of high level objectives in GOV that are translated and enforced on the appropriate NEMs, which act on their managed elements.

Finally, the deployment of UMF enhances the network operation with functionalities that are not in general implemented in the current OSS: prediction and self-healing capabilities. The former throws a forecast of the future behaviour of network and services, providing for instance preventive diagnosis of possible future



malfunctioning. The latter automatically repairs the network elements in case of deficient behaviour, avoiding thus manual intervention.

## 4.3 Deployment of UMF in Heterogeneous Mobile Network infrastructure

### 4.3.1 Context – heterogeneous network with small cells

Heterogeneous mobile network (or HetNets in 3GPP nomenclature) is a particular active area in 4G networks, and more specifically, in LTE-Advanced. HetNets encompass macro, micro, pico (or small cells), femto cells (or Home Base Stations – HNB), and relay stations. HetNets are used as a solution to enhance both capacity and coverage. The growing interest in HetNets with both small cells and femto cells is related to the constant and significant growth in data traffic demand. Femto cells provide capacity and coverage solutions for the growing portion of indoor traffic, while small cells are seen as one of the most effective capacity solutions to the increasing traffic demand. Figure 59 shows the user plane architecture of a small cell network.

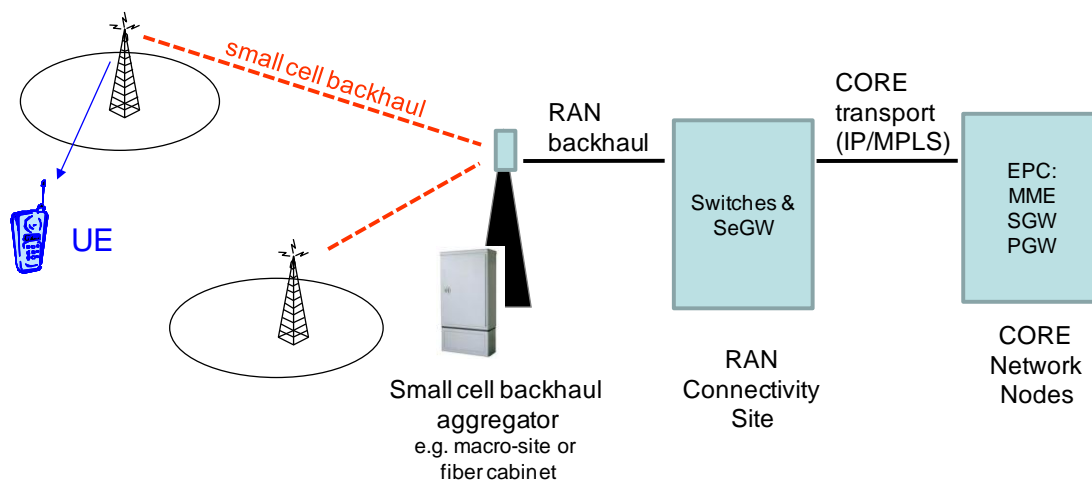


Figure 59: Small cell network

Large scale deployment of small cells implies the configuration, optimization and troubleshooting of thousands to tens of thousands small cells in a large city. Hence the management of such a network should be fully autonomous. As an example, self-optimization of a small cell will require to allocate the right amount of traffic to the small cell via load balancing and coverage capacity optimization. The interference generated by the small cells on the macro-cell, and vice versa, will require interference management. The connectivity of users to the network nodes will need to be managed according to the mobility profile of the users (e.g. high speed mobiles should be connected to the macrocell etc.). When traffic is low, the small cells could be switched off. These functions should be handled by SON functionalities (as specified by the LTE-Advanced standard) such as:

- CCO – Coverage and Capacity Optimization
- MLB – Mobility Load Balancing, taking into account both radio access and backhaul (BH), covering both intra- and inter-cell cases
- eICIC – enhanced Inter-Cell Interference Coordination
- MRO – Mobility Robustness Optimization
- ES – Energy Saving
- others

The SON functionalities will need to exchange information and knowledge, to be coordinated to avoid conflicts and to enforce stable operation, and to contribute to a high level operation / business objective defined by the operator via a H2N governance tool.

While the contribution of UMF CORE blocks: GOV, KNOW and COORD looks evident, these functional blocks are only partially reflected in standardization of 3GPP systems (were SON mechanisms have been introduced) in general and heterogeneous networks in particular (see Section 5.1.1 for more details).

#### 4.3.2 UMF deployment in heterogeneous cell networks

This section highlights how UMF can enhance management capabilities and performance of present and future heterogeneous networks. The extension to UMF deployment with other heterogeneous nodes (e.g. femto cells) is similar. Figure 60 below shows the information exchange between UMF and different entities of the heterogeneous network.

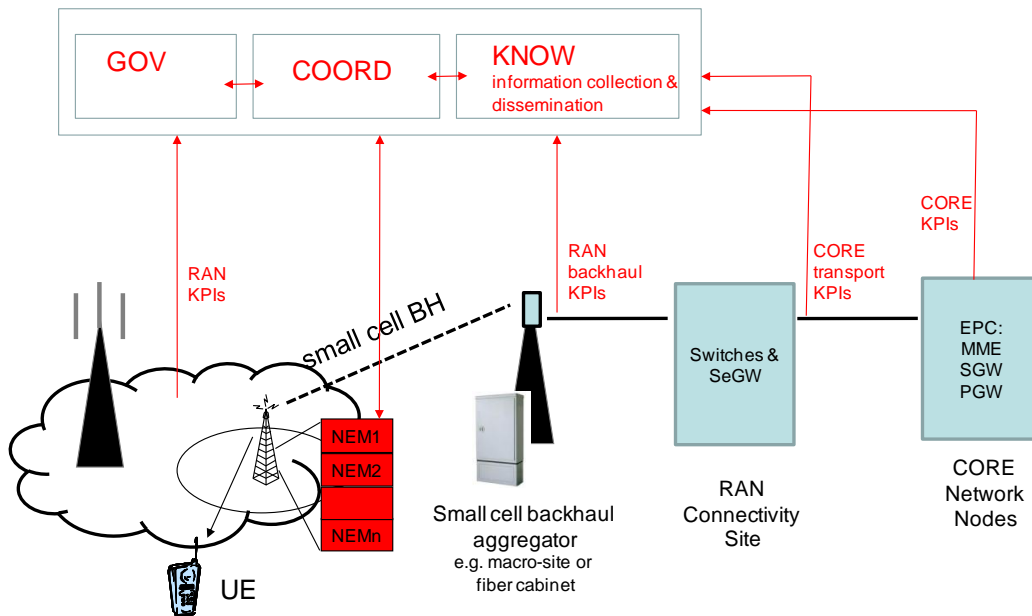


Figure 60: UMF deployment in heterogeneous networks with macro-and small cells

##### Governance

In present 3GPP networks, SON operation is defined by the SON policy control functions, which are logical functions [39]. A policy is defined for the operation of a specific SON. The natural evolution that UMF offers is the possibility to introduce high level goals that, by means of policy translation can be mapped and propagate till specific policies of each NEM. Policies can be translated to NEMs' operation, but also to specific input to COORD that manages the joint optimization process. Governance of heterogeneous networks can remain within the OMC (EMS), but if a general end-to-end view of management is sought while exploiting information from different network segments (e.g. access and transport segments), Governance functions should be deployed within the NMS.

##### Knowledge

Knowledge functions, namely the collection, processing and dissemination for NEMs can become critical in small cell network. It is recalled that today SON functionalities utilize performance metrics from the access network; however, bottlenecks can originate from the transport network (backhaul) or even the core network. Hence the collection and distribution of information is essential. In small cell network, the knowledge block can be deployed within a base station (eNodeB in LTE) in a distributed (control plane) implementation, or within the NMS when centralized implementation is chosen.

## Coordination

The management of coordination function is very recent in standardization, and is mainly addressed in Release 12 of 3GPP [39] (See Section 4.3): The coordination solution in LTE-Advanced networks is generic, and thus encompasses also heterogeneous networks. Coordination is addressed per couple or three specific SON functionalities. Possible conflicts are described, and a guideline for a solution is proposed in the form of weights or priorities assigned by the policies.

UMF proposes different solutions for NEM coordination, in the form of conflict avoidance – used proactively, before activating the NEM, or in the form of joint optimization solutions. Clearly, the coordination approach is more general, and not a “case-by-case” approach. Depending on the time scale sought, a distributed, centralized or hybrid implementation can be envisaged:

*Distributed implementation:* the coordination entity is implemented in the stations of the heterogeneous network, namely small cells and macro-cells. Distributed solution is important when high reactivity is essential.

*Hybrid implementation:* Certain solutions use a learning / pre-processing phase that need to be performed in a centralized manner. In this case the Coordination block will be deployed in both NMS and the cells (macro / small cells).

*Centralized implementation:* Centralized solutions will be implemented in the EMS (namely the Operation and Maintenance Centre – OMC) or the NMS. NMS implementation has two advantages: (i) enable multi-segment management, encompassing access, transport, core and other network segments, and (ii) opening the arena of autonomic management / SON technology to new actors such as software IT companies, and not just to the traditional OMC vendors.

## NEMs

The NEMs are a natural evolution of SON functions, which provide solutions for their deployment, and management, and allow benefitting from other functionalities provided by UMF core blocks. Interestingly, the way NEMs have been specified and implemented in the project show similarities with certain SdN concepts, which simplify and provide further flexibility in (large scale) deploying and managing the NEMs. NEMs can be deployed at different radio access nodes (base stations, small cells), in the CORE network (possibly in the MME), transport network, OMC, and in a dedicated server in the NMS.

## 4.4 Migration of UMF and NEMs from Legacy Networks

Migration of the UMF and NEM solutions into real world commercial networks is an important successive step after the UMF and NEMs have been developed within research activities. While theoretical aspects of migration strategies have already been discussed earlier [8], the present chapter focuses on some of the practical aspects of UMF and NEM migrations.

The UniverSelf UMF concept and vision do also aim to address real needs and visions of commercial network operators. The ongoing telecommunication trends result in increasing complexity as well as in requiring more and more self-organization solutions (NEMs) with their need to manage and coordinate these (e.g. via UMF). Therefore, the UniverSelf solutions and value propositions can provide a large benefit for the network operators, which should therefore be interested in migrating the beneficial solutions.

Currently, the legacy networks do often have already a few independent autonomous functions (SON) for selected use cases, and their simultaneous coordination –if any– may often be done via their Operations & Maintenance (O&M) centre. Within these existing networks, there is already indirectly a considerable amount of UniverSelf support and prerequisites implemented, such as collection of measurements, reporting of parameters, as well as the technical possibilities that detailed system parameters can be configured. Depending on the particular NEM and on the level of UMF functionality, these existing means may not be complete or perfect, but at least they provide a good basis which can be used to add those still missing UniverSelf aspects.

For actual migration the UniverSelf approaches, UMF and/or particular NEMs into commercial networks, there are some migration aspects to take into account, issues which have the potential to delay or hinder migration. These include:

- **Commercial product maturity:**

After the NEM and/or UMF have been invented, developed and tested on a research level, as the next step, this NEM will then have to be integrated into the commercial product which may require cooperating with the product business division. This leads to that the NEM must have reached a mature level and must be completely reliable as well as that any expected possible interactions must be controllable. These processes may take a certain amount of time.

- **Risk avoidance and trust:**

Network operators usually tend to be extremely careful to alter their running system as any malfunction would immediately cause a huge financial loss and would negatively impact their reputation. For deploying UMF functionalities in existing systems, the proof of reliability and trust are major aspects about which the network operators will need to be convinced about.

- **Small incremental migration steps:**

In order to minimize the risk and to build up trust, the network operators may tend to successively introduce new features and observe their interactions. This favours migrating one NEM after the other and successively adding UMF functionality as needed.

Furthermore, some network operators may tend to hesitate giving up control about what is happening. Observing small steps of incremental migration activities allows them to build up trust in the autonomic UniverSelf and UMF activities.

- **Aspects of open standards:**

The UniverSelf UMF approach of open standards is very positive, but for the UMF use cases involving complex equipment, it is so far not really in line with the current practice that –despite of standardisations– the network infrastructure in one small local area is typically taken from one single vendor without inter-vendor equipment mixing there. Furthermore equipment manufactures do not really provide open external interfaces, unless required. Thus a certain change needs to be achieved, the UMF and NEM relevant interfaces need to be open and documented and it needs to be ensured that a particular NEM works safely and trustworthy together with the existing network infrastructure. Looking e.g. at 3GPP, achieving standardization agreements may involve a considerable amount of effort and could involve longer times scales. In an earlier, short term, migration step, a network vendor itself could guarantee the problem-free operation of a particular NEM within his own network. This then requires involving the particular network vendor in the certification process of a particular NEM.

- **Benefits and priorities:**

The UMF+NEM customer, such as a network operator, is most interested in his own benefit, in his value proposition, but this may be independent of the technical solution behind. Thus for migrating to the UMF approach, the network operator needs to be convinced that the UniverSelf-advantages are larger than their effort and potential risks. Operator's priorities can also influence when and whether to migrate certain to UMF and NEM functions.

For a particular migration task, the concrete situation needs to be addressed individually for their specific application. Some NEMs and their UMF support can be migrated easily, while some other NEMs, such as strongly interacting ones and/or ones who impact the inner core of the communication system, could require more migration efforts. In a later step, after the UniverSelf approaches will have become widespread, the UMF concepts can simply be activated, but within the short term first migration steps, more detailed observation and adaptation to the particular situation could be desired or required.

## 4.5 Conclusions

In this chapter, we studied the deployment of UMF in three existing network infrastructures: a target metro Ethernet, FTTH access network and heterogeneous mobile network. Although it is the fact that it is difficult to deploy new technologies to the current mature systems, it seems that it is possible to realize the deployment if it can be achieved in a step-by-step manner. Inventory, provisioning and fault management are the three areas that will benefit the most with the introduction of NEMs and the UMF core blocks, in particular Governance and Knowledge, while in the case of mobile networks, the Coordination block will play a significant role.

The actual migration from current network approaches to UMF-empowered networks should be carefully designed, taking into account the commercial maturity of the solutions, their standardization, and a risk and benefits analysis.

## 5 UMF deployment in emerging network infrastructures

Software defined Networks (SdN) and Network Functions Virtualisation (NFV) enable fully exploitation of service-aware networks, devoid of technology-specific attributes, through provision and management of dynamically network functions and services. UMF core mechanisms and NEMs can form the basis for software-driven networks architecture development. In this context, the UMF deployment to SdN and NFV architectures is presented in this section, indicating that UMF and SdN/NFV can largely benefit from each of other.

### 5.1 UMF deployment in Software Defined Networks

Software defined Networks and Network Functions Virtualisation (NFV) are currently under 2<sup>nd</sup> round of developments. The first round developments were in the form of programmable networks technology [40]. Various solutions are currently proposed by vendors and standardization is still underway. As such the proposed UMF deployment in SdN / NFV context is to be understood as a proposal and not a final view. In the following sections, we highlight the main SdN concepts that are relevant for discussing UMF deployment and we present three deployment scenarios.

#### 5.1.1 SdN main concepts

In this section, we present some of the main concepts that are discussed around Software defined Networks (SdN). The list is not exhaustive as discussions are still underway in several bodies or forums. We can highlight the following initiatives:

- ITU-T SG13: Future networks. An SdN framework is defined and under development.
- Open Networking Foundation (ONF).
- IETF ForCES and IETF SDNRG.

An SdN related standardization activity is ETSI NFV which mainly addresses the visualisation of networking functions. It will be discussed in section 5.1.1.

Figure 61 and Figure 62 are highlighting the main discussed SdN concepts:

- A decoupling of data plane and control plane with the introduction of a programmable control plane.
- Programmable control plane Southbound Interface (aka Control Data Plane Interface in ONF) towards Data plane: e.g. Openflow, ForCES to control data plane related network devices.
- Northbound Interface relying on APIs between applications and the SdN control plane. APIs between the SdN control and applications layers, enable applications to operate on an abstraction of the network, leveraging network services and capabilities without being tied to the details of their implementation. SdN makes the network not so much “application-aware” as “application-customized” and applications not so much “network-aware” as “network-capability-aware”.
- Southern Interfaces: The OpenFlow / ForCES protocols are deployable between network infrastructure SdN-enabled devices and the SdN control software. OpenFlow uses the concept of flows to identify network traffic based on pre-defined match rules that can be statically or dynamically programmed by the SdN control software. It also allows configuration on how traffic should flow through network devices based on parameters such as usage patterns, applications, and cloud resources. ForCES protocols are used for communications between Control Elements (CEs) and Forwarding Elements (FEs) in a ForCES enabled Network Element (ForCES NE) - RFC3654
- East-West Interface between Control plane/network domains or controllers.
- System Management concepts are not yet explicitly and uniformly integrated in SdN/NFV, however further developments on this subject are expected in the next period.

Some differences exist between the ITU-T, IETF, ETSI and ONF views. For example, it is not clear how “ONF Network Services” that are part of the ONF SdN Control Software is related to ITU-T Network OS, Programmable Control Plane or SdN services/applications.

The frontier between control and management planes is also becoming blurred. Management plane is not explicitly mentioned and is globally missing in the description. It is important for UMF which is dedicated to

self-management and self-x functions that are typically targeting management plane but are also blurring the frontier with control plane.

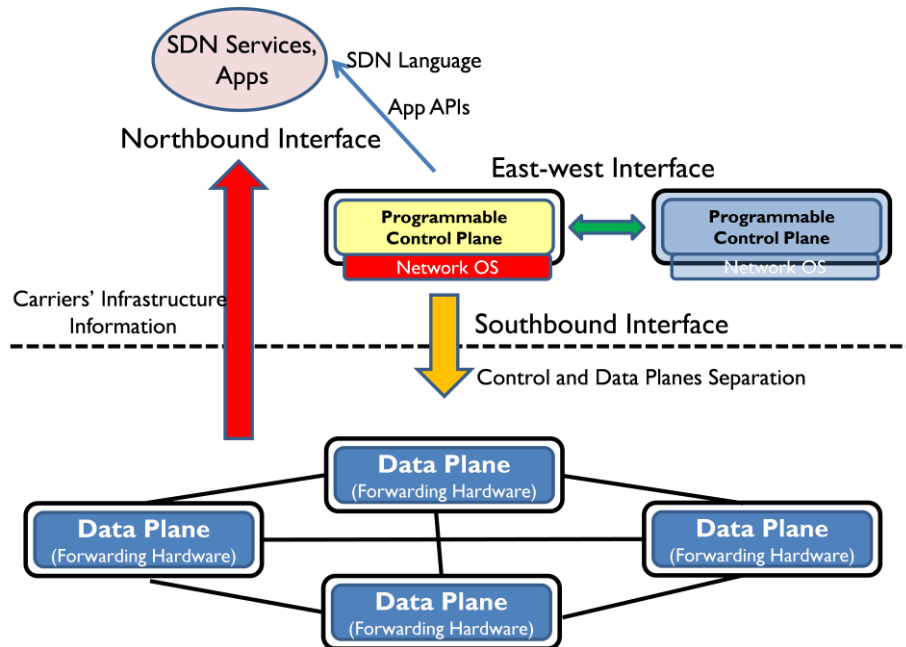


Figure 61: SdN Framework as discussed in ITU-T SG13

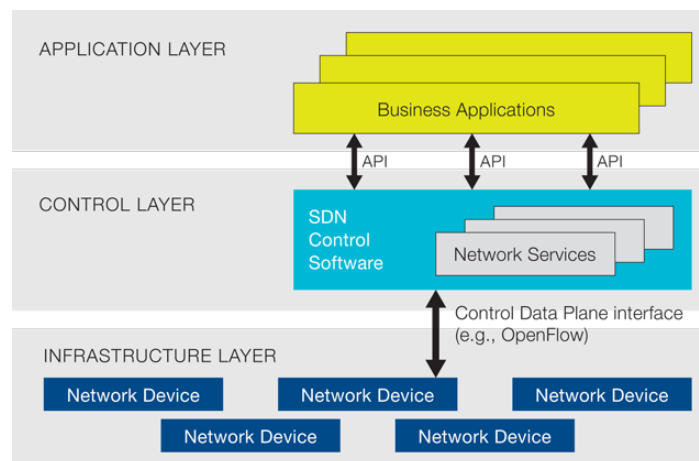


Figure 62: SdN framework as discussed in ONF

### 5.1.2 Connecting UMF to SdN enabled infrastructures

Figure 63 presents the UMF framework with the three UMF core blocks and the NEMs and related interfaces.



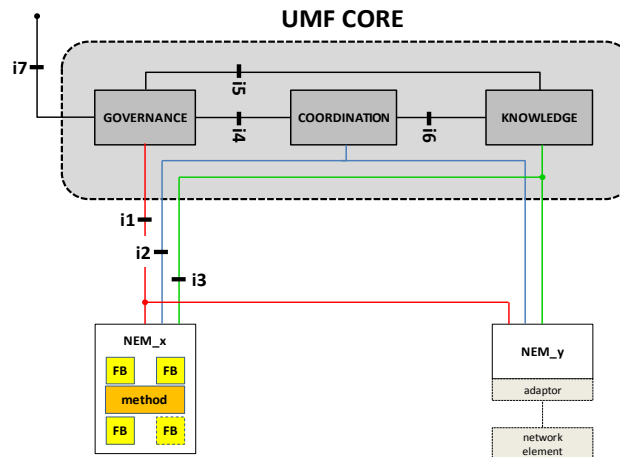


Figure 63: UMF framework

We need to distinguish the UMF Core blocks from the NEMs for this mapping.

UMF core blocks are in charge of the needed functions to manage NEMs and ensure self-management driven by operator goals. This is related to management plane. SdN applications could benefit from UMF core blocks in the case of self-management of SdN applications. In this case, SdN applications would be similar to NEMs.

Moreover NEMs are containing the algorithms/methods part of the self-x functions and as such they contain different and specific protocols for resource configuration of network elements. UMF specifications don't address the NEM adaptation or gatewaying to SdN enabled network element and in particular to the OpenFlow-enabled network device from any vendor, including switches, routers, and virtual switches. This needed adaptation could benefit from the SdN frameworks and could be facilitated by the programmable control plane.

We can also discuss the possible NEM inclusion into the programmable control plane. SdN control plane includes only the functions that can control only SdN-enabled network device from any vendor, including switches, routers, and virtual switches. SdN controllers provide visibility and control over the network, they can ensure that access control, traffic engineering, quality of service, security, and other policies are enforced consistently across the network infrastructures, including branch offices, campuses, and data centers. As stated above, NEMs are management applications which include self-x methods and at least one closed control loop enabling self- functionality of a network element. It acts on the resources which are managed by the NEMs. These two definitions are mainly disjunctive and as such without a revision of the scope of the SdN control plane, the NEMs which are applicable to SdN enabled domains and the NEMs which are applicable to other networking domains cannot be easily fitted in the control plane functionality.

As such we have 3 options:

- Option 1: to not consider as appropriate the inclusion of the NEMs in the current SdN control plane.
- Option 2: to extend the definition of the control plane to include management functionality, including self-management. In this case it makes sense to consider as appropriate the NEMs inclusion in a revised SdN control plane.
- Option 3: to extend the SdN framework to include the management functionality explicitly (i.e. in the current version system management concepts are not yet explicitly and uniformly integrated in SdN/NFV) with control & management plane separation.

In the following sections, we consider only option 3 as a target.

We present three scenarios: the first scenario considers the NEMs as SdN applications. The second scenario considers UMF as a management integration framework for SdN with Non SdN Enabled Infrastructures. The third scenario considers the UMF core positioning in a revised SdN architecture.

#### 5.1.2.1 Scenario 1: NEMs as a SdN app

In this scenario, we consider that a NEM<sub>for-SdN</sub> performs a self-x function on an SdN enabled infrastructure. Such a NEM is the Virtual Infrastructure Management (VIM) (see section 2.3.3). To support this scenario, we

mention that ONF is discussing the introduction of some components at the level of a network controller that is related to network services. From the ITU-T perspective, it could be related to the Network OS role.

Figure 64 is illustrating the NEM<sub>for-SdN</sub> positioning. First, the related to NEM<sub>for-SdN</sub> needs to support the APIs that are provided to SdN applications. NEM<sub>for-SdN</sub> needs also to support the UMF interfaces with the three UMF core blocks. In this context, it is difficult to foresee how SdN applications and UMF Core blocks could be connected as they are driven by the same goals. Second, in this context it may be a good idea to bring the UMF interfaces closer to the SdN APIs. Third, the NEM adaptor is relying on the programmable control plane functionality provided by Network OS. As such the developments of the Network OS would need to include the right level of abstraction usable by the NEM<sub>for-SdN</sub>.

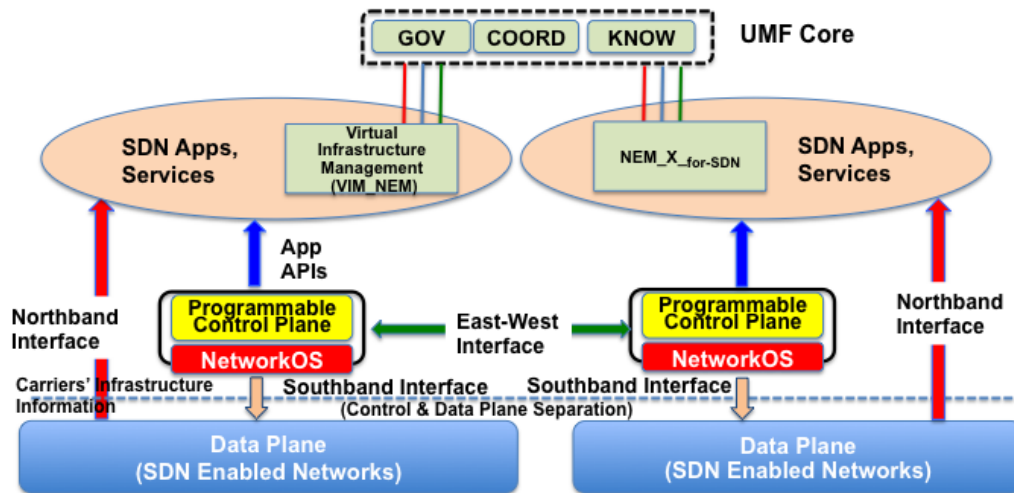


Figure 64: NEMs in SdN enabled Networks as an SdN app

#### 5.1.2.2 Scenario 2: UMF as management integration framework for the management of SdN with Non SdN Enabled Infrastructures

In this scenario we consider that a NEM<sub>x</sub> applicable to a non SdN enabled infrastructure which is not a part of the SdN applications domain as illustrated in Figure 65. In this context, the positioning of the UMF core appears to be more obvious than for the previous scenario: UMF interfaces are used to manage NEMs or SdN applications.

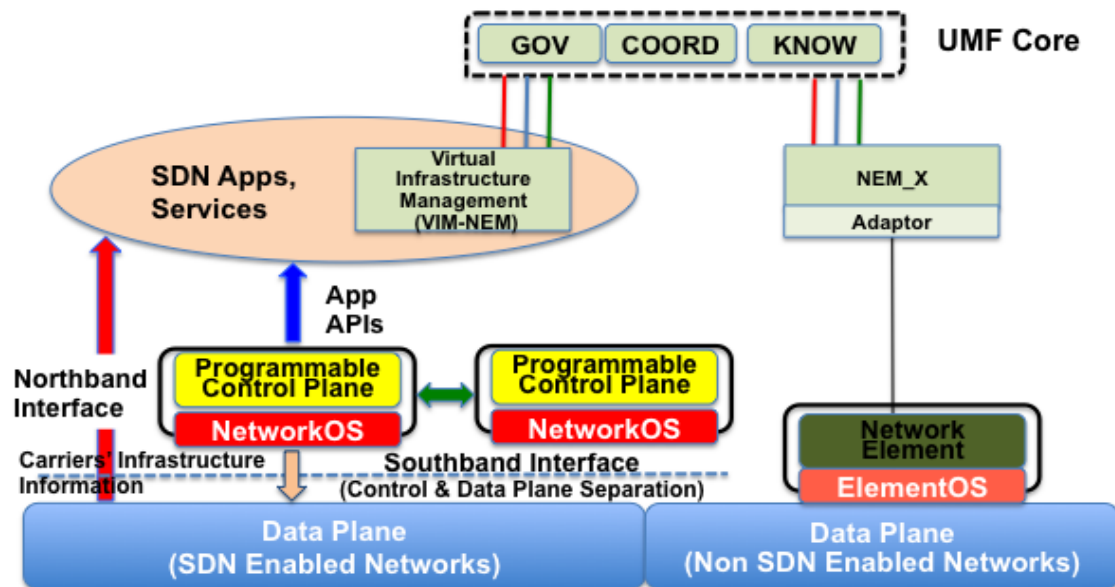


Figure 65: NEM as an SdN app

As such UMF could be established as a management integration framework for the non SdN and SdN enabled infrastructures.

#### 5.1.2.3 Scenario 3: UMF core positioning in a revised SdN architecture

As presented in Section 5.1.1 System Management concepts are not yet explicitly and uniformly integrated in current established SdN standardisation fora (IETF FORCES & SdNRG, ONF, ITU-T SG13: Future networks/ SdN, ETSI NFV), however further developments on this subject are expected in the next period.

Due to the extreme dynamic nature of the SdN enabled infrastructure it would be appropriate to propose a separation of the control and management planes that could be developed and maintained separately. The control & management planes separation enables the positioning of the full UMF core in the SdN revised architecture which is presented in the next figure.

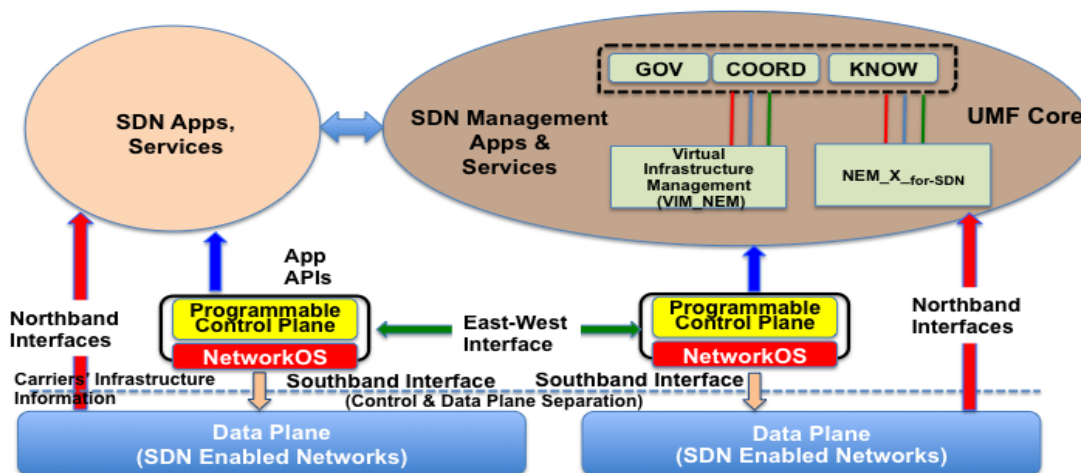


Figure 66: UMF position in a revised SdN framework

One development for a revised SdN framework [41] as presented in the previous figure considers the control and management separation and it also proposes the transition from the current SdN as “application-customized” towards “application-aware” infrastructures and for applications to be “network-aware”. It proposes a service-aware and management-aware network control infrastructure for heterogeneous networks (i.e., wired and wireless) that uses software driven features for the elaboration, development, and validation of

networking concepts. It aims also at optimal integration of the connectivity and management layers. It operates across multiple network environments and on top of private and public network clouds utilising fixed and mobile virtual resources, OpenFlow enabled network devices like switches and routers, and networks of Smart Objects. Figure 67 sketches such SdN revised framework with the introduction of UMF Core blocks and NEMs.

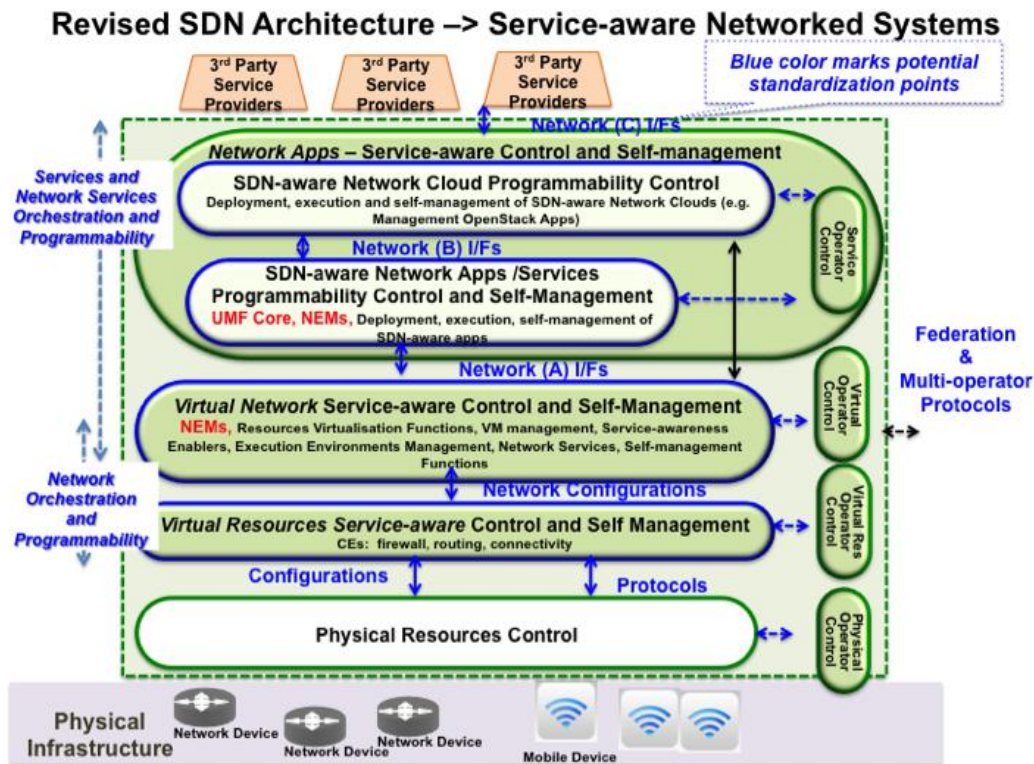


Figure 67: SdN Revised Framework

## 5.2 UMF deployment in Network Functions Virtualization Architecture

Network Functions Virtualisation aims to address the problems related to maintenance of a large and increasing variety of proprietary hardware appliances which are rapidly reaching end of life, requiring much of the procure design- integrate-deploy cycle to be repeated with little or no additional revenue benefit. These problems are addressed by leveraging standard virtualisation technology to consolidate many network equipment types onto industry standard high volume servers, switches and storage, which could be located in Datacentres, Network Nodes and in the end user premises. It should be noted that an industry specifications group - Network Functions Virtualisation (NFV) [42] - was formed in January 2013 under the European Telecommunications Standards Institute (ETSI). The ETSI NFV work is highly complementary to Software Defined Networking (SdN) as these topics are mutually beneficial but are not dependent on each other. Network Functions can be virtualised and deployed without an SdN being required and vice-versa. The basic NFV architecture is presented in the following figure.

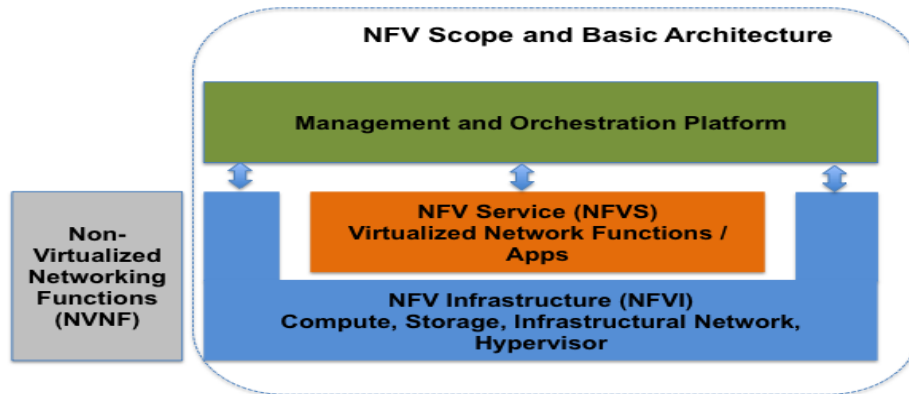


Figure 68: Network Functions Virtualization (NFV) Basic Architecture

A mapping and deployment of UMF functions and mechanisms to Network Functions Virtualization Architecture is presented in the following figure.

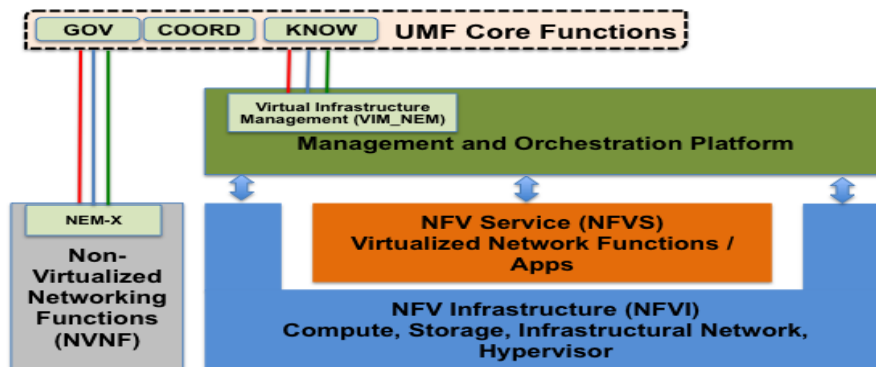


Figure 69: UMF Mapping to NFV

UMF would add autonomicity characteristics to the basic NFV architecture and it would also provide a management integration framework for the management of NV with Non NV Networking Functions.

### 5.3 Conclusions

The UMF deployment in the SdN environment involves hosting and positioning of the UMF functionality into appropriate abstractions for the management and control planes. UMF core blocks and applicable NEMs to the SdN enabled infrastructures are clearly related to management plane which is proposed to be separated from the control plane. We identified three scenarios for the UMF deployment in SdN enabled infrastructures: the first scenario considers the NEMs as SdN applications; the second scenario considers UMF as a management integration framework for SdN with Non SdN Enabled Infrastructures; the third scenario considers the UMF core positioning in a revised SdN architecture.

The key point is the positioning of UMF as a unique unifying management framework of both non SdN and SdN enabled infrastructures and domains. UMF core blocks could also provide an efficient way to manage SdN applications (inc. NEMs).

On the other hand, NFV is a complementary approach to SdN, and can benefit with the introduction of UMF, which provides means for the Governance, Coordination and for the Knowledge exchange. Furthermore, UMF can easily manage both virtual and non virtual network functions in an integrated way.

## 6 UMF positioning on existing network and management architectures

This section presents a view of UMF possible positioning on existing network and management architectures. Specifically, it focuses on mapping of UMF system to 3GPP LTE network architecture and eTOM model, indicating relevant issues/challenges for further work/cooperation from/between the both sides (UniverSelf and standardization bodies).

### 6.1 UMF functions in 3GPP-LTE systems

The first approach of mapping of UMF system to 3GPP LTE network architecture has already been accomplished in the context of D4.6 [6], along with the identification of the need for evolution in interfaces and protocols, since there is no direct link between E-UTRAN and PCRF/ PCEF.

In order to position UMF with respect to 3GPP (up to Release 12 which is presently on-going), we consider the each of the UMF core blocks and the related activities in 3GPP. The topic of coordination has received a particular interest in 3GPP [39][43] due to the urgent need to enable simultaneous operation of different SON functionalities while avoiding conflicts.

#### 6.1.1 Governance

Governance mechanisms are covered in 3GPP in three areas: Policy and Charging Control (PCC), Access Network Discovery and Selection Function (ANDSF) Managing Object (MO), and SON. A brief summary of these three topics is first presented, followed by the corresponding positioning of UMF.

##### **PCC:**

3GPP specifies the PCC functionality that encompasses two main functions [44]:

- Flow Based Charging, including charging control and online credit control, for service data flows and application traffic;
- Policy control (e.g. gating control, QoS control, QoS signalling, etc.).

The policies are enforced by the Policy and Charging Enforcement Function (PCEF). The policies / rules for control and charging are provided by the functional PCRF functional entity.

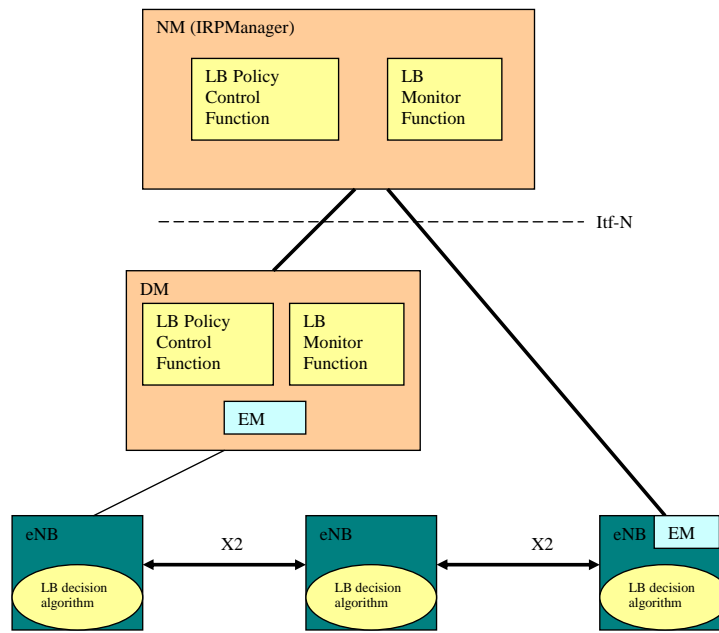
##### **ANDSF:**

Governance tools have also been introduced in 3GPP within the ANDSF MO. The ANDSF MO is used to manage Inter-System Mobility Policy (ISMP) and Inter-System Routing Policy (ISRP) as well as access network discovery information stored in a UE supporting provisioning of such information from an ANDSF [45].

##### **Policies governing SON:**

For each SON functionality, an associated policy handles its detection and optimization [43]. The detection and optimization specific policies are not defined, but rather the related Performance Metrics (PM) that can be used. The logical functions including policies for Load Balancing (LB) are shown Figure 70. Similarly, conflict avoidance and coordination is handled by means of policies. Policies can for example assign priorities for different SONs.





**Figure 70: Policies in SON functional architecture in 3GPP [43].**

3GPP approach for governance utilizes the Network Resource Model (NRM). The TMF model used by UMF covers also policies, which have the advantage of unifying the policy models for the different managed entities. Governance functions specified in UMF and implemented in the project (see Case Study-II on Governance[38]) enables dynamic definition of high level objectives, their translation to policy rules, the enforcement of the derived rules/actions and their evaluation.

The autonomic functions (SON in the 3GPP nomenclature) for dynamic resources administration, configuration and optimization are encapsulated into Network Empowerment Mechanisms (NEMs). The NEMs are described by a manifest, which forms semantic information that specifies the managed equipment, the acquired inputs, the produced outputs and the kind of actions that a NEM can implement. Based on the manifest, Governance facilitates the seamless deployment, management and supervision of NEMs through their functions. Policy Derivation and Management (PDM) defined in UMF covers the following functionalities that could be useful in the framework of 3GPP:

- (i) providing storage for the policies and facilitating the management of the Policy Repository,
- (ii) checking whether the different policies have conflicts and resolving them according to the UMF conflict resolution mechanisms,
- (iii) translating the policies to lower level policies,
- (iv) activating NEMs to configure, optimize or self-heal network elements, resources usage and services
- (v) evaluating if the enforcement of the network actions fulfils the defined high level objectives and service requirements

In summary, UMF paves the way to a natural evolution of the policy management as defined till Release 12 of 3GPP, its unification for the different management tasks (PCC, ANDSF, SON servers / entities, and other to come), and the interfaces that can ease the deployment. Governance functions can be deployed in

- a dedicated server in the NMS-Network Management System / OMC- Operation and Maintenance Centre
- the RAN nodes (eNodeB, heterogeneous network nodes)
- the CORE network nodes such as the MME – Mobility Management Entity



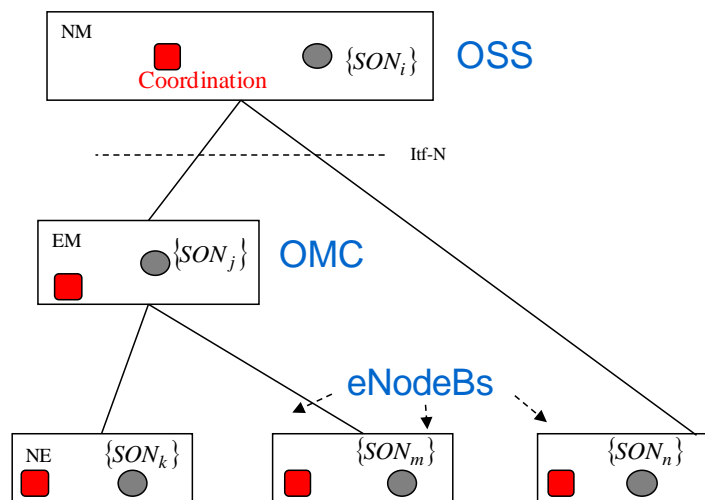
### 6.1.2 Coordination

3GPP defines the coordination functionality to manage conflicts between SON functionalities. Three possible architectures can be compliant to 3GPP specifications (with possible combination of these):

- (i) Distributed, where the SON functionalities are deployed at eNodeBs
- (ii) EM-centralized (EM for Element Management), where the SON functionalities are deployed at the EMS – Element Management System (i.e. the OMC)
- (iii) NM-centralized (NM for Network Management), where the SON functionalities are deployed at the NMS.

While the first two solutions are in the domain of traditional network equipment vendors, solution (iii) involves also OSS software vendors and network planning and optimization/SON tool companies.

Coordination functions can be introduced to all the three above architectures, and be located at eNodeBs, MME, OMC (EMS) and OSS (NMS), as shown in Figure 71.



**Figure 71: Coordination can be implemented at different levels in the network**

3GPP/SA5 specifies high-level solution for avoiding conflicts between SON functionalities, by means of policies set by the operator to a coordination function [39][43]. The SON Coordination is a logical function, which means it can be implemented as a separate entity or as part of a SON function. When the SON Coordination Function is implemented as a separate function entity, all the SONs send the necessary information to the SON Coordination Function which coordinates these SONs as a centralized control point. Specific policies that can be of the form of weights / priorities and specific actions are defined for a set of identified possible conflicts. For example, among cases addressed are:

- COC-ES (Cell Outage Compensation, Energy Saving)
- COC-CCO-ES (CCO – Coverage Capacity Optimization)
- HOO-LBO (Handover Optimization – Load Balancing Optimization)

#### UMF mapping

3GPP coordination functionality is in its early stages in the sense that it is based on prioritization and on conflict avoidance strategies. UMF COORD solution can be seen as a big step forward with respect to the current 3GPP coordination solutions. It is geared by a *Joint Optimization Approach* that, in addition to a proactive conflict identification solution, provides a run time joint optimization solution of two or more SON (NEMs).

Furthermore, UMF provides the orchestration elements allowing the full integration of the coordination block into a unified autonomic solution with interfaces and knowledge exchange with GOV and NEMs.

UMF COORD CORE block can be located at (and be mapped into) different entities of 3GPP architecture:

- (i) eNodeB (distributed and short time scale)
- (ii) MME (centralized and short time scale)
- (iii) OMC (centralized and long(er) time scale)
- (iv) OSS (centralized and long time scale)

### 6.1.3 Knowledge

3GPP are not dealing with functions that aggregate, collect, store, register, and produce information and knowledge. The main alignment could be with the exchanged data we modelled using the TMF SID and the 3GPP NRM which is already tackled for the resource layer in 3GPP TR.32828 [46].

## 6.2 UMF mapping in eTOM

For an effective mapping of UMF (functions and a set of data as inputs and outputs) to eTOM (a business process model that describes all the enterprise processes required by a service provider, and analyses them to different levels of detail according to their significance and priority for the business), we need to first understand the relationships between processes and functions according to [47]:

Process is defined as “the flow of activities to solve a particular business problem, or part of it. At early analysis stages for processes, the means of availability and how the data flows are not significant. Whether or not data is handed over or accessed in a central database is not addressed. However, processes are concerned with the triggers that set them into action”.

Function is described as a unit of processing together with its associated data inputs and outputs.

A process will typically make use of activities in a number of functions. Multiple processes may employ a given function. Thus, there is in principle a many-to-many mapping between process and function.

The picture below depicts the application of these definitions to UMF functions and data and eTOM processes:

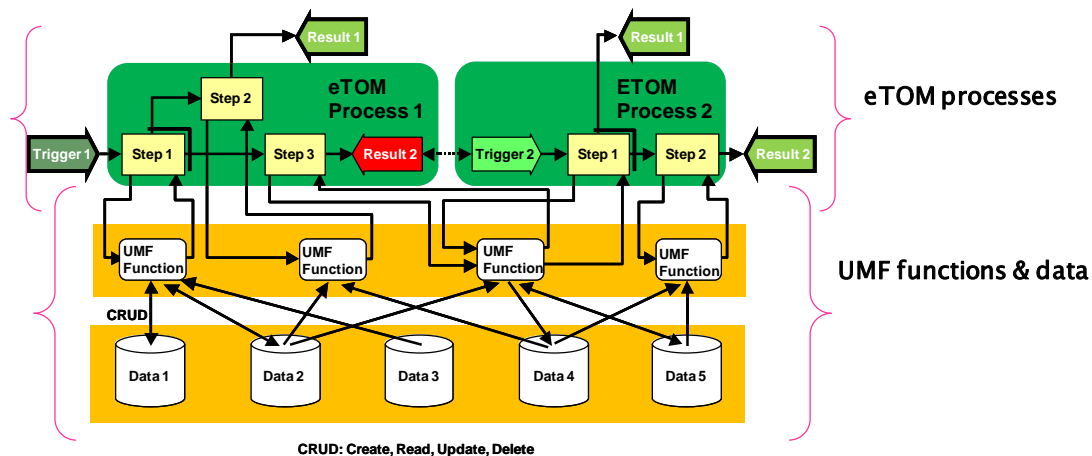


Figure 72: eTOM Processes and UMF functions based on GB921

As stated a one to one mapping is not appropriate, in the following we map the adequacy of UMF functions to eTOM processes: mainly the UMF functions and data are called by the eTOM operational part.

Furthermore, UMF introduces functions to manage the so-called NEM, while NEM are assigned to Resources to deliver specific services. In this regard, we add the “NEM management & Operations”. This layer gathers specific functions to manage NEMs such as coordination related functions for conflicts management or Knowledge related to NEM information analysis or building.

Other functions are concerned with all eTOM layers including the “NEM management & Operations layers”. The policy derivation function, the knowledge functions about services, resources and NEMs, etc. The picture below illustrates the high level mapping of eTOM to UMF functions.

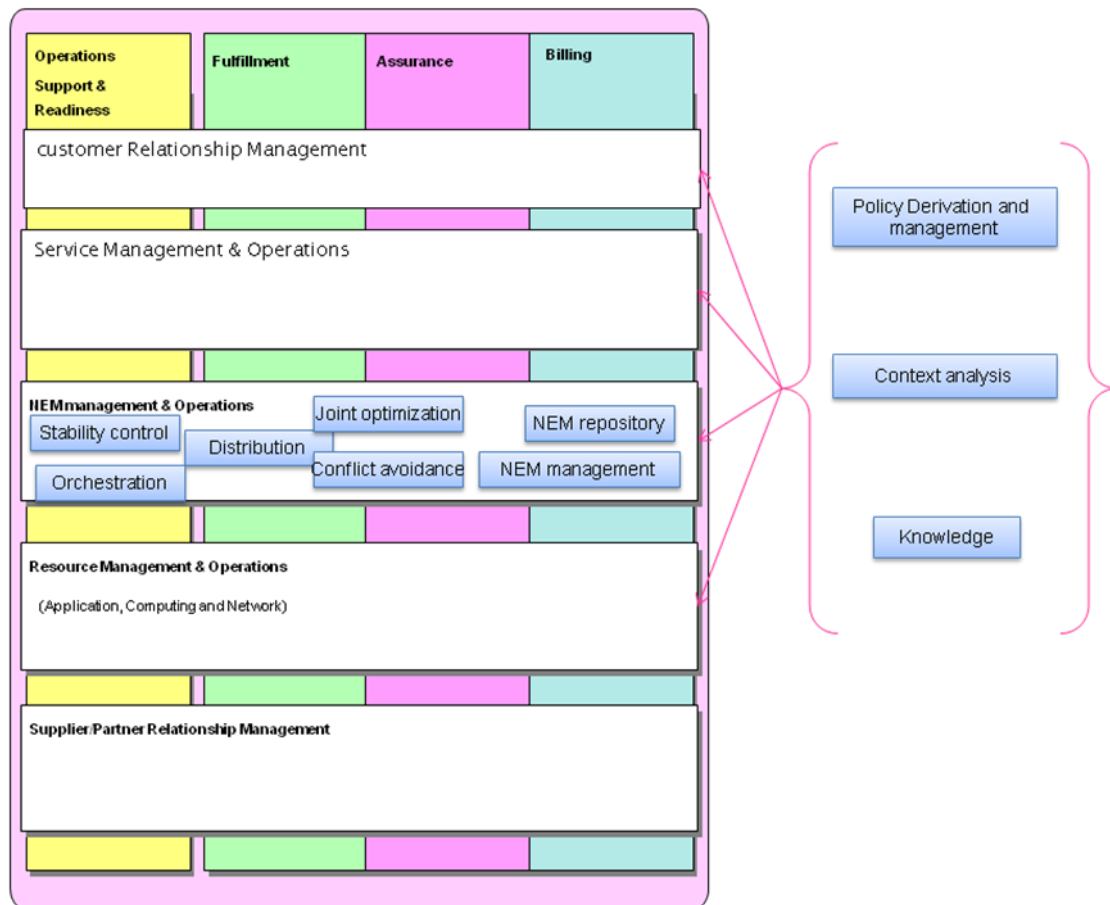


Figure 73: Mapping of eTOM to UMF functions

### 6.3 Conclusions

This section presented a view of a possible positioning of UMF on existing network and management architectures, namely 3GPP LTE network architecture and eTOM model.

As analysed in this chapter, UMF paves the way to a natural evolution of the 3GPP policy management, its unification for the different management tasks, and the interfaces for the NEM lifecycle management. On the other hand, UniverSelf coordination mechanisms present more advanced solutions than the coordination functionality in 3GPP, based on prioritization and on conflict avoidance strategies only. Finally, 3GPP does not include facilities for the Knowledge production, storage and exchange.

The mapping to eTOM requires adding a new layer, called “NEM management & Operations”, which embeds some of the functions and operations in Governance, Knowledge and Coordination. NEMs are assigned to Resources layer and deliver specific services.

## 7 Conclusions

Deliverable D4.15 “Synthetic analysis of deployment results and impacts” provides the final complete synthesis of deployment and impact results of UniverSelf. In this context, evaluation of deployed UniverSelf final integrated prototype was performed. The final prototype allows for the demonstration of three scenarios, where different NEMs are deployed in SDN core + WLAN access, LTE network and virtual infrastructure testbeds.

The OPEX analysis of one of the scenarios of the integrated prototype was performed using *a posteriori* ‘Toy Model’ analysis. The ‘Toy Model’ incorporates simulation results in the OPEX impact expectation analysis. Based on the metric results provided by NEM developers, it assesses the impact on eTOM processes, and, together with the expert evaluations from the QFD exercise, allows for an assessment of the impact on the different OPEX categories, and subsequently on the total OPEX for a typical network operator. The results are presented as ranges instead of single figures. For the scenario analysed, the NEM with the highest results is expected, in isolation with the UMF, to deliver 3.8 and 4.6 per cent OPEX savings typically. In addition, a formula is proposed to assess the total impact of all NEMs of the scenario together with the UMF. This formula starts from the most promising NEM, and adds new ones in decreasing order of expected impact. With all NEMs being active, the predicted OPEX saving for a typical network operator will fall between 11 and 13 per cent.

In the second part of the deliverable, UMF deployment was studied for existing and emerging network infrastructures: Metro Ethernet, FTTH and Heterogeneous Mobile networks. The analysis provides a view of the future behavior of network and services, providing for instance preventive diagnosis of possible future malfunctioning, and how UMF can enhance the network operation with functionalities that are not in general implemented in the current OSS: prediction and self-healing capabilities, enhancing the repair of the network elements in case of deficient behavior. In essence, the three studied scenarios feature UMF implementation maturity, which is essential for boosting industry adoption.

The mapping of UMF to emerging network infrastructures, SdN and NFV, demonstrates that the incorporation of UMF and NEMs to those can enhance them with self-management capabilities. Finally, the positioning of UMF with respect to 3GPP and eTOM architectures highlights relevant challenges that might drive the standardization work of UniverSelf solutions.

In summary, the combined examination of the aforementioned work confirms the efficient operation and great level of reusability of the developed NEMs, their smooth cooperation with other NEMs and UMF core mechanisms, their significant contribution to address the network problems and OPEX reduction, and in short, UMF credibility. The deployment results ensure the federation of the chosen methods to experiment scenarios and UMF applicability to different networking architectures.

## 8 References

- [1] UniverSelf Deliverable D2.4, 'UMF Design – Release 3'.
- [2] UniverSelf Deliverable D3.5, 'Adaptation and fine tuning of the identified parameter optimization methods'
- [3] UniverSelf Deliverable D3.7, 'Adaptation of learning and operation methods to specific needs of networks and services'
- [4] UniverSelf Deliverable D3.8, 'Impact of reusable communication mechanisms and hierarchies in cooperation strategies and incentives'
- [5] UniverSelf Deliverable D3.9, 'Handbook on optimization, learning, operation and cooperation methods'
- [6] UniverSelf Deliverable D4.6, 'Synthesis of deployment results – Release 1'.
- [7] UniverSelf Deliverable D4.12, 'Synthesis of deployment results – Release 2'.
- [8] UniverSelf Deliverable D4.7, "Analysis of the impact of deployment of autonomic networking functionalities – Release 1".
- [9] UniverSelf Deliverable D4.14, "Analysis of the impact of deployment of autonomic networking functionalities – Release 2".
- [10] UniverSelf Deliverable D4.4, 'First prototype of a use case'.
- [11] UniverSelf Deliverable D4.5, 'Leaflet of the first prototype of a use case'
- [12] UniverSelf Deliverable D4.8, 'Second prototype of a use case'.
- [13] UniverSelf Deliverable D4.9, 'Leaflet of the second prototype of a use case'
- [14] UniverSelf Deliverable D4.10, 'Third prototype of a use case'.
- [15] UniverSelf Deliverable D4.11, 'Leaflet of the third prototype of a use case'
- [16] UniverSelf Deliverable D4.13, 'Integrated demonstration of a combination of use-case prototypes'
- [17] S. Davy, B. Jennings, J. Strassner, 'The Policy Continuum – A Formal Model', 2nd IEEE International Workshop on Modelling Autonomic Communications Environments, MACE 2007
- [18] Guerrero, A., Villagra, V.A., de Vergara, J.E.L., Sanchez-Macian, A., Berrocal, J., "Ontology based Policy Refinement Using SWRL Rules for Management Information Definitions in OWL" Proc. 17th IFIP/IEEE International Workshop on Distributed Systems, Operations and Management (DSOM), Dublin, Ireland (October 2006)
- [19] López de Vergara, A. Guerrero, V.A. Villagrà, J. Berrocal, "Ontology Based Network Management: Study Cases and Lessons Learned", Computer Science Journal of Network and Systems Management, Volume 17, Number 3, pp. 234-254, 2009
- [20] Future Network and Mobile Summit (FUNEMS), Lisbon, Portugal, 2013 (<http://www.futurenetworksummit.eu/2013/>)
- [21] OWL Web Ontology Language Overview. W3C Recommendation 10 February 2004. Available at: <http://www.w3.org/TR/owl-features/>. Last accessed: 15th July 2013
- [22] SWRL: A Semantic Web Rule Language Combining OWL and Rule ML. W3C Member Submission 21 May 2004. Available at: <http://www.w3.org/Submission/SWRL/>. Last accessed: 15th July 2013
- [23] UniverSelf document, "Case Study – Part II, SON and SON collaboration according to operator policies", May 2013.
- [24] TM Forum, Business Process Framework (eTOM), GB921 Addendum E: Application Note: End-to-End Business Flows, Release 9.1, TM Forum Approved Version 9.4, April, 2011.
- [25] L. P. Sullivan, "Quality Function Deployment," *Quality Progress*, vol. 19, no. 6, pp. 39–50, 1986.
- [26] L. K. Chan and M. L. Wu, "Quality function deployment: A literature review," *European Journal of Operational Research*, vol. 143, no. 3, pp. 463–497, 2002.
- [27] S. Verbrugge, S. Pasqualini, F.-J. Westphal, M. Jäger, A. Iselt, A. Kirstädter, R. Chahine, D. Colle, M. Pickavet, and P. Demeester, "Modeling operational expenditures for telecom operators," in *9th Conference on Optical Network Design & Modelling (ONDM 2005), Milan, Italy, 2005*.

- [28] C. Cid, M. Ruiz, L. Velasco, and G. Junyent, "Costs and Revenues Models for Optical Networks Architectures Comparison."
- [29] "Motorola LTE Self Organizing Networks. Motorola's revolutionary SON solution for LTE OPEX reduction," Motorola, 2009.
- [30] J. Buvat and S. Basu, "Quest for Margins: Operational Cost Strategies for Mobile Operators in Europe," Capgemini, 42, 2009.
- [31] J. Harno, "Techno-economic analysis of beyond 3G mobile technology alternatives," *info*, vol. 11, no. 3, pp. 45–63, 2009.
- [32] "Rethinking operational processes can offer telcos competitive savings," Deloitte, 2009.
- [33] J. Bernard, T. Broschuk, P. Doane, M. Jadoul, and M. Nespatti, "Optimization. Secrets to network success: small changes deliver big results," Alcatel-Lucent, 2010.
- [34] H. Schwarz and M. Schmitz, "Green Gold. How energy savings are given due consideration in outsourcing agreements," *DMR, Magazine for Management and Technology*, vol. 4, 2011.
- [35] M. Locker, L. Glover, and E. Heisler, "The seven cost management principles for wireless carriers: When your average cost reduction program just won't do," Deloitte, 2011.
- [36] K. Zhu and B. Mukherjee, "Traffic grooming in an optical WDM mesh network," *IEEE Journal on Selected Areas in Communications*, vol. 20, pp.122-133, Jan. 2002 -- X. Zhang and C. Qiao, "Wavelength assignment for dynamic traffic in multi-fiber WDM networks," in *Proc. 7th Int. Conf. Computer*
- [37] ITU-T Recommendation G.984: Gigabit capable passive optical networks (GPON)
- [38] UniverSelf document, "Case Study – Part II, Network and service governance", May 2013.
- [39] 3GPP TS 28.628, Self-Organizing Networks (SON) Policy Network Resource Model (NRM) Integration Reference Point (IRP); Information Service (IS), <http://www.3gpp.org/ftp/Specs/html-info/28628.htm>
- [40] Galis, A., Denazis, S., Brou, C., Klein, C. "Programmable Networks for IP Service Deployment" ISBN 1-58053-745-6, pp450, June 2004, Artech House Books, [www.artechhouse.com/Default.asp?Frame=Book.asp&Book=1-58053-745-6](http://www.artechhouse.com/Default.asp?Frame=Book.asp&Book=1-58053-745-6)
- [41] A.Galis, J. Rubio-Loyola, S. Clayman, L. Mamatas, S. Kukliński, J.Serrat, T. Zahariadis « Software Enabled Future Internet – Challenges in Orchestrating the Future Internet » - MONAMI 2013- 5th International Conference on Mobile Networks and Management, Cork, 23-25 September 2013 ; <http://monami.org/2013/show/home> and A.Galis presentation at the 3rd ETSI Future Network Workshop 8-11 April, Sofia Antipolis ; <http://www.etsi.org/news-events/news/617-2013-fnt-intro>
- [42] Network Functions Virtualisation (NFV) – ETSI Industry Group <http://portal.etsi.org/portal/server.pt/community/NFV/367> & white paper [http://portal.etsi.org/NFV/NFV\\_White\\_Paper.pdf](http://portal.etsi.org/NFV/NFV_White_Paper.pdf)
- [43] 3GPP TS 28.627, Self-Organizing Networks (SON) Policy Network Resource Model (NRM) Integration Reference Point (IRP); Requirements, <http://www.3gpp.org/ftp/Specs/html-info/28627.htm>
- [44] 3GPP TS 23.203, Policy and charging control architecture, Release 12, June 2013, <http://www.3gpp.org/ftp/Specs/html-info/23203.htm>
- [45] 3GPP TS 24.312, Access Network Discovery and Selection Function (ANDSF) Management Object (MO), Release 12, June 2013, <http://www.3gpp.org/ftp/Specs/html-info/24312.htm>.
- [46] 3GPP TR 32.828, Study on alignment of 3GPP generic Network Resource Model (NRM) Integration Reference Point (IRP) and the TeleManagement Forum (TMF) Shared Information/Data (SID) model, Release 10, March 2011.
- [47] GB921-eTOM\_Core\_Standards\_Release\_13-0

## Abbreviations

3GPP	3 <sup>rd</sup> Generation Partnership Project
3GPP LTE	3GPP Long Term Evolution
AN	Access Node
ANDSF	Access Network Discovery and Selection Function
AON	Active Optical Network
AP	Access Point
API	Application Programming Interface
AWG	Array Waveguide Grating
BH	BackHaul
BHO	Backhaul Optimization
BSS	Business Support System
BW	BandWidth
CCO	Coverage and Capacity Optimization
CDN	Content Distribution Network
CFO	Check Feasibility & Optimize
CLB	Core Load Balancing
CLI	Command Line Interface
COC	Cell Outage Compensation
CPU	Central Processing Unit
DA	Directory Agents
DBA	Dynamic Bandwidth Assignment
DSL	Digital Subscriber Line
E2E	End-to-End
eICIC	enhanced Inter-Cell Interference Coordination
EM	Element Management
EMS	Element Management System
eNodeB	Evolved NodeB
eTOM	Enhanced Telecom Operations Map
ES	Energy Saving
ETSI	European Telecommunications Standards Institute
E-UTRAN	Evolved Universal Terrestrial Radio Access Network
FORCES	FORwarding and Control Element Separation
FTTH	Fibre To The Home
FTTx	Fiber to the x (x = H for home, B for building, C for curb and N for node)
FUNEMS	FUTure NEtworks and Mobile Summit
GENA	Generic Event Notification Architecture
GPON	Gigabit-capable PON
GUI	Graphical User Interface
H2N	Human-to-Network
HLO	High Level Objective
HNB	Home Base Stations
HOO	Handover Optimization
HTTP	Hyper Text Transfer Protocol
HR	Human Resources
ICIC	InterCell Interference Coordination



ICD	Information Collection & Dissemination
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IFEO	Information Flow Establishment & Optimization
INS	Intentional Naming System
IP	Internet Protocol
IPKP	Information Processing & Knowledge Production
ISI	Information Storage & Indexing
ISMP	Inter-System Mobility Policy
ISRP	Inter-System Routing Policy
ITU	International Telecommunication Union
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
LB	Load Balancing
LBO	Load Balancing Optimization
LNH	Lightweight Network Hypervisor
LTE	Long Term Evolution
LTE-A	LTE Advanced
MDA	Mesh Directory Agents
MIB	Management Information Base
MLB	Mobility Load Balancing
MME	Mobility Management Entity
MO	Managing Object
MRB	Mobility Robustness Optimization
MSA	Mesh Service Agents
mSLP	mesh Service Location Protocol
NEM	Network Empowerment Mechanism
NFV	Network Function Virtualization
NGN	Next Generation Network
NM	Network Management
NMS	Network Management System
NOC	Network Operations Center
NRM	Network Resource Model
OAM	Operations Administration and Maintenance
OFDMA	Orthogonal Frequency-Division Multiple Access
OLT	Optical Line Terminal
ONF	Open Networking Foundation
O&M	Operations & Maintenance
OMC	Operations and Maintenance Center
ONT	Optical Network Terminal
OPEX	Operational Expenditures
OSP	OutSide Plant
OSS	Operations Support System
OWL	Ontology Web Language
P2P	Peer to Peer
PCC	Policy and Charging Control
PCEF	Policy and Charging Enforcement Function
PCRF	Policy and Charging Rules Function

PDM	Policy Derivation and Management
PM	Performance Metrics
PO	Placement Optimization
PON	Passive Optical Network
PtMP	Point to Multi Point
PtP	Point to Point
QFD	Quality Function Deployment
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
REST	REpresentation State Transfer
RN	Remote Nodes
SA	Service Agents
SdN	Software defined Networks
SDN	Software Driven Networks
SDNRG	Software Defined Networking Research Group
SH	Self Healing
SID	Shared Information Model
SINR	Signal to Interference plus Noise Ratio
SLA	Service Level Agreement
SLP	Service Location Protocol
SNMP	Simple Network Management Protocol
SOAP	Simple Object Access Protocol
SON	Self Organized Networks
SSDP	Simple Service Discovery Protocol
SWRL	Semantic Web Rule Language
TCP	Transmission Control Protocol
TDM	Time-Division Multiplexed
TMF	TeleManagement Forum
TT	Trouble Ticket
UA	User Agent
UMF	Unified Management Framework
UPnP	Universal Plug and Play
URL	Uniform Resource Locators
VIM	Virtual Infrastructure Management
VLAN	Virtual Local Area Network
VLSP	Very Lightweight Software Driven Network and Services Platform
VR	Virtual Router
WLAN	Wireless Local Area Network
WLB	Wireless Load Balancing
xDSL	x Digital Subscriber Line, where x=A (Asymmetric), H(High-bit-rate), I (ISDN), M(Multi-rate), P(Power-line), R(Rate-adaptive), S(Symmetric), SH(Single-pair High-speed) or V (Very-high-bit-rate)
XML	eXtensible Markup Language