Grant Agreement No.: 318600

SODALES

SOftware-Defined Access using Low-Energy Subsystems

Funding Scheme: ***Small or medium-scale focused research project STREP - CP-FP-INFSO***
Activity: **ICT-8-1.1 - Future Networks**

D4.3 Report on installation parameters for the field service validation

Due date of the Deliverable: Month
Actual submission date: 14th August 2015
Start date of project: November 1st 2012 Duration: 36 months
Project Manager: Carlos Bock | i2CAT
Version: 1.0

Author List:
André Brízido (PTIN), Wojciech Hajduczenia (PTIN), Tiago Mendes (PTIN), Victor Marques (PTIN), Pedro Mendes (PTIN), Carlos Bock (i2CAT), Jordi Ferrer Riera (i2CAT), David Levi (ETH), Michael C. Parker (UEssex), Volker Jungnickel (HHI)

Project co-funded by the European Commission in the 7th Framework Programme (2007-2013)		
Dissemination Level		
PU	Public	✓
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Abstract

This deliverable D4.3, "Report on installation parameters for the field service validation" describes the SODALES demonstration architecture, both physical (including diagrams and pictures of the several locations involved) and control layers, and the services that are available on the demonstration network.

The service setup procedure (element configuration and service provision) on each network element, through the use of the SODALES Open Access management platform, is also described. Finally, the validation infrastructure is presented and described, including the test equipment and end-user QoS and QoE objective and subjective evaluation tools.

This deliverable also describes the setup of the commercial validator that SODALES is preparing for the final project review. It is located in Centelles, Catalonia, where there is an Open Access Network with several ISPs offering ultra-high speed services.

This demonstrator will be implemented once the validator in Aveiro is complete, and will contain most of its elements in order to verify the performance of the SODALES developments in a real commercial deployment context.

Table of Contents

List of Figures

List of Acronyms

3GPP	Third Generation Partnership Project
AAI	Authentication, Authorisation, and Identification
ADSL	Asymmetric Digital Subscriber Line
ADSL2	Asymmetric Digital Subscriber Line and Annex J
AGP	Aggregation Point
aGW	Advanced Gateway
AIS	Alarm Indication Signal
APON	ATM-PON
ARN	Active Remote Node
ARPU	Average Revenue Per User
ATM	Asynchronous Transfer Mode
AWG	Arrayed Waveguide Grating
BBU	Baseband Unit
BER	Bit Error Rate
BoF	Broadband over Fibre
BPON	Broadband PON
BS	Base Station
BTS	Base Transceiver Station
BW	Bandwidth
CAPEX	Capital Expenditure
CATV	Cable Television
CCM	Continuity Check Message
CDMA	Code Division Multiple Access
CDU	Customer Demarcation Unit
CE	Customer Equipment

CEN	Customer Ethernet Network
CFM	Connectivity Fault Management
CFP	C Form-factor Pluggable
CO	Central Office
CoMP	Co-ordinated Multi-Point
CoS	Class of Service
CPE	Customer Premises Equipment
CPRI	Common Public Radio Interface
CSI	Channel State Information
C-VLAN	Customer VLAN
DBA	Dynamic Bandwidth Allocation
DGD	Differential Group Delay
DMM	Delay Measure Message
DMR	Delay Measure Reply
DSL	Digital Subscriber Line
E2E	End-to-End
EC	Ethernet Channel
EFM	Ethernet First Mile
E-LAN	Ethernet Local Area Network
EMS	Element Management System
ENNI	External Network Network Interface
EoS	Ethernet over SDH
EPL	Ethernet Private Line
EPON	Ethernet PON
EVC	Ethernet Virtual Channel/Connection
EVPL	Ethernet Virtual Private Line
FDD	Frequency Division Duplex
FSAN	Full Service Access Network

FSO	Free-Space Optics
FTTB	Fibre-To-The-Building
FTTC	Fibre-To-The-Curb
FTTCab	Fibre-To-The-Cabinet
FTTH	Fibre-To-The-Home
FTTx	Fibre-To-The-x
GEM	GPON Encapsulation Method
GFP	Generic Framing Procedure
GPON	Gigabit-PON
GSM	Global System for Mobile
HDTV	High Definition Television
H-NID	Hybrid NID
HO	High Order
HQoS	Hierarchical QoS
HIS	High Speed Internet
HSPA	High Speed Packet Access
HW	Hardware
ICI	Inter-Cell Interference
ID	Identifier
IEEE	Institute of Electrical & Electronic Engineers
IETF	Internet Engineering Task Force
iid	independent & identically distributed
IoF	Intermediate frequencies over Fibre
IP	Internet Protocol
IPTV	IP Television
IR	Infra-Red
IRN	Intermediate RN
ISP	Internet Service Provider

ITU	International Telecommunications Union
JD	Joint Detection
JT	Joint Transmission
LAN	Local Area Network
LB	Loop Back
LCAS	Link Capacity Adjustment Scheme
LED	Light Emitting Diode
LMM	Loss Measure Message
LMR	Loss Measure Reply
LO	Low Order
LoS	Line of Sight
LTE	Long Term Evolution
MAC	Medium Access Control
MEF	Metro Ethernet Forum
MEG	Maintenance Entity Group
MEN	Metro Ethernet Network
MEP	Maintenance End Point
MPO	Multiple Performance Objectives
NID	Network Interface Device
MIB	Management Information Base
MIMO	Multiple-Input Multiple-Output
MPLS	Multi-Protocol Label Switching
NGPON	Next-Generation PON
NC	Network Controller
NMI	Network Management Interface
NMS	Network Management System
NNI	Network Node Interface
NUNI	NID UNI

O&M	Operations and Maintenance
OAM	Operations, Administration and Maintenance
OAN	Open Access Network
OBSAI	Open Base Station Architecture Initiative
ODN	Optical Distribution Network
OFDM	Orthogonal Frequency Division Multiplexing
OLT	Optical Line Termination
OMCI	ONT Management and Control Interface
ONT	Optical Network Termination
ONU	Optical Network Unit
OPEX	Operational Expenditure
ORI	Open Radio equipment Interface
OVC	Operator Virtual Channel
P2MP	Point-to-Multi-Point
P2P	Point-to-Point
PB	Provider Bridge
PBB	Provider Backbone Bridge
PDH	Plesiochronous Digital Hierarchy
PDU	Protocol Data Unit
PE	Provider Edge
PHY	Physical layer
PM	Performance Monitoring
POI	Point of Interconnect
PON	Passive Optical Network
POTS	Plain Old Telephone Service
PPB	Parts Per Billion
PTFR	Poly-Tetra-Fluoro-Ethylene
PTN	Packet Transmission Network

PTP	Precision Time Protocol
PW	Pseudo Wire
PWE	Pseudo Wire Emulation
QoS	Quality of Service
RAN	Radio Access Network
RBS	Radio Base Station
RDI	Remote Defect Indication
RF	Radio Frequency
RF	Radio Frontend
RMON	Remote Performance Monitoring
RN	Remote Node
ROADM	Reconfigurable Optical Add/Drop Multiplexer
RoF	Radio over Fibre
ROSA	Receiver Optical Sub-Assembly
RRH	Remote Radio Head
RSP	Retail Service Provider
RT	Remote Terminal
SCN	Signalling Communication Network
SDR	Software Defined Radio
SLA	Service Level Agreement
SLM	Synthetic Loss Measurement Message
SLR	Synthetic Loss Measurement Reply
SME	Small- to Medium-sized Enterprise
S-NID	Service NID
SNMP	Simple Network Management Protocol
SOAM	Service OAM
SoC	System-on-a-Chip
SODALES	SOftware-Defined Access using Low-Energy Subsystems

SOHO	Small Office/Home Office
SONET/SDH	Synchronous Optical Network/Synchronous Digital Hierarchy
SP	Service Provider
SSM	Synchronous State Message
STM	Synchronous Transfer Mode
S-VLAN	Service VLAN
SW	Software
TCO	Total Cost of Ownership
TDD	Time Division Duplex
TDM-PON	Time-Division-Multiplexing PON
TD-SCDMA	Time Division Synchronous Code-Division Multiple-Access
TE	Traffic Engineering
T-NID	Transport NID
ToP	Time over Packet
TWDM-PON	TDM/WDM-PON
UDWDM	Ultra-Dense WDM
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
UNI	User Network Interface
VCAT	Virtual Concatenation
VDSL	Very-high-data-rate DSL
VLAN	Virtual LAN
VLC	Visible Light Communication
VMEN	Virtual Metro Ethernet Network
VoD	Video on Demand
VoIP	Voice over IP
VUNI	Virtual User Network Interface
WAN	Wide Area Network

WCDMA	Wideband Code Division Multiple Access
WDM	Wavelength Division Multiplexing
WiMAX	Worldwide Interoperability for Microwave Access
XFP	10G small Form-factor Pluggable module
XG-PON	10-Gb/s capable PON
ZF	Zero Forcing

1 Introduction

This deliverable D4.3, “SODALES: Report on installation parameters for the field service validation” starts by describing the SODALES demonstration architecture, at both the logical and physical layers (including pictures of the several locations involved), and the services that are available on this demonstration network. The service setup procedure (element configuration and service provision) on each network element is described, as is the use of the Open Access management platform. Finally, the validation infrastructure is presented and described, including the test and measurement equipment and end-user QoS and QoE objective and subjective evaluation tools.

The current SODALES prototype scenario demonstrator already employs the following project specific equipment developments:

- The ARN from PTI
- The carrier Ethernet CPE from ETHERNITY
- The VLC link from HHI
- The Open Access Management Platform from i2CAT

By way of background context, the basic SODALES architecture scenario is illustrated in Figure 1, where:

- There is the possibility for dedicated wavelength connections to the CO for SMEs and RBSs using 10G wavelengths
- The ARN serves a customer capacity of up to 136 Gb/s, consisting of:
  - 96 statistically-multiplexed residential customers served using 96xGE ports from two TA48GE cards
  - 3 SME and one RBS served using 10GE ports from one TU40G card
- In this scenario, the ARN offers 20-Gb/s uplink capacity
  - by means of 2×10Gigabit Ethernet ports in a 1+1 redundant configuration, using the single CXO160 card ports. The uplink capacity can be increased by using a second CXO160G card.
  - The oversubscription between the uplink and the customer ports is lower than 1:7

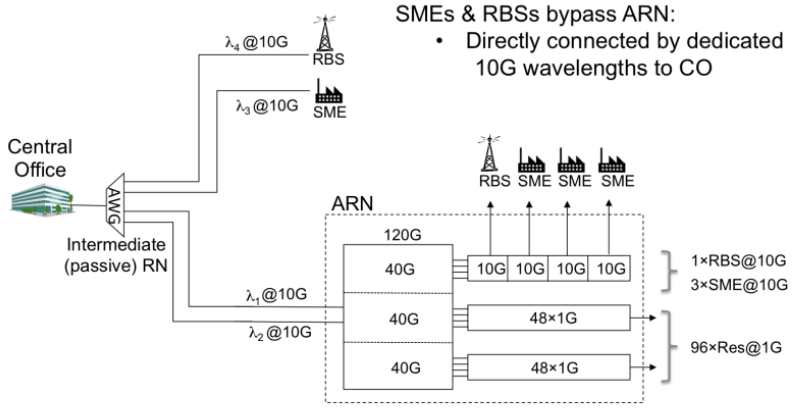

Figure 1: The basic SODALES scenario

The SODALES architecture uses a point-to-point topology DWDM fibre network configuration without add-drop multiplexers.

1.1 Introducing other SODALES components

The next step in the demonstration is to include both the CPE, which has already been demonstrated in a standalone way, and the OpenNaaS Advanced Control and Management Plane into the scenario. This allows the full establishment and monitoring of services in the SODALES architecture, fulfilling the project objectives.

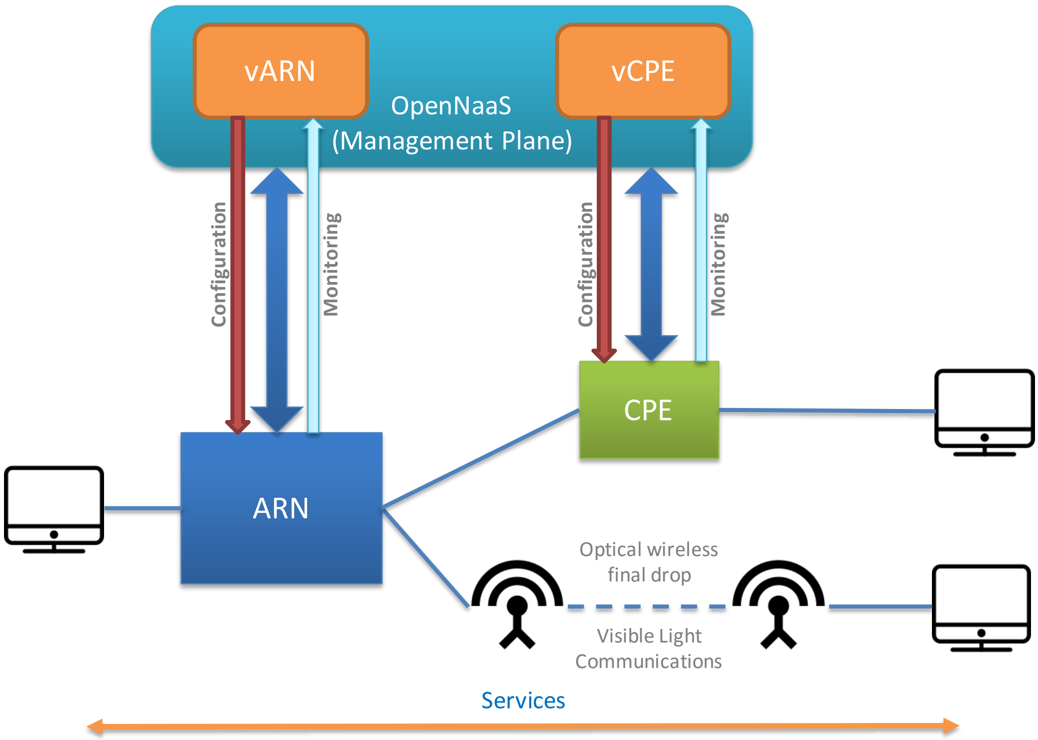

Figure 2: The SODALES service configuration and monitoring model

OpenNaaS demonstrations in Brussels have already been scheduled at the beginning of 2015, preceding its introduction in the PTI DELTA platform.

The management plane based on OpenNaaS is focused on creating a virtual slice, based on the SODALES open access principle, and then enabling the corresponding service provider to configure the slice (i.e. the virtual ARN and the virtual CPE representations). More details of the associated control and management plane implementation are provided in deliverable D3.4.

2 Sodales Demonstrator Architecture

2.1 Physical and logical infrastructure

End-to-end connectivity using all the various SODALES-developed components will be validated using the existing DELTA platform covering PTI's own campus and three end-office switch locations of the Portugal Telecom service provider in Aveiro.

This extensive optical and copper network covers around 300 customers (out of the more than 2000 PTI employees) who are are provided with high-speed Internet and IP TV services. A number of DSLAM units provide service over copper with ADSL2+ to Portugal Telecom customers that also join the trial. The PTI campus, where most of its employees work, is covered by the OLT1T0, a low density OLT capable of delivering multi-play services to 512 customers (assuming a 1:64 splitting ratio) over ITU-T G.984.x GPON. The possibility exists of expanding the DELTA trial to the *Instituto de Telecomunicações* (IT) premises on the Aveiro university campus through both existing fiber and radio links.

The GPON and DSL equipment in DELTA are managed by PTI's Agora-NG NMS solution, installed at a cluster of virtual machines in the PTI campus data center. A TR-069 Auto Configuration Server (ACS) at the data center is responsible for the provisioning of services at the GPON and DSL CPEs. Physical and virtual machines exist at the data center for the validation of an array of services other than IP TV or IPv4 and IPv6 access to the internet.

There is also an interface to the SALINA full NGN/IMS reference demonstration platform, designed as a technological showcase of the different PTI solutions in this area and enabling interoperability tests with third-party systems with the objective of encouraging synergies with industrial partners and scientific communities.

SALINA already includes an Enhanced Packet Core (EPC) for LTE, and multiple initiatives exist to install a cell site with an enhanced Node B (eNB) in the PTI campus for the introduction of mobile services into the DELTA and SALINA demonstrators.

The SODALES scenario introduces into DELTA the OLT1T3 large scale OLT as the Central Office (CO) equipment, a number of ARN equipments, and several CPEs using a variety of final-drop technologies, to validate all the services and functionalities that have been developed during the project.

Depending on the overall DELTA network load, its architecture can be be adjusted in order to choose the component to be subject to traffic stressing. Thus, the OLT and DSLAM used can all eventually uplink to a single ARN, for example, in order to analyse its statistical multiplexing gains. Traffic generators can also be used to create additional loading.

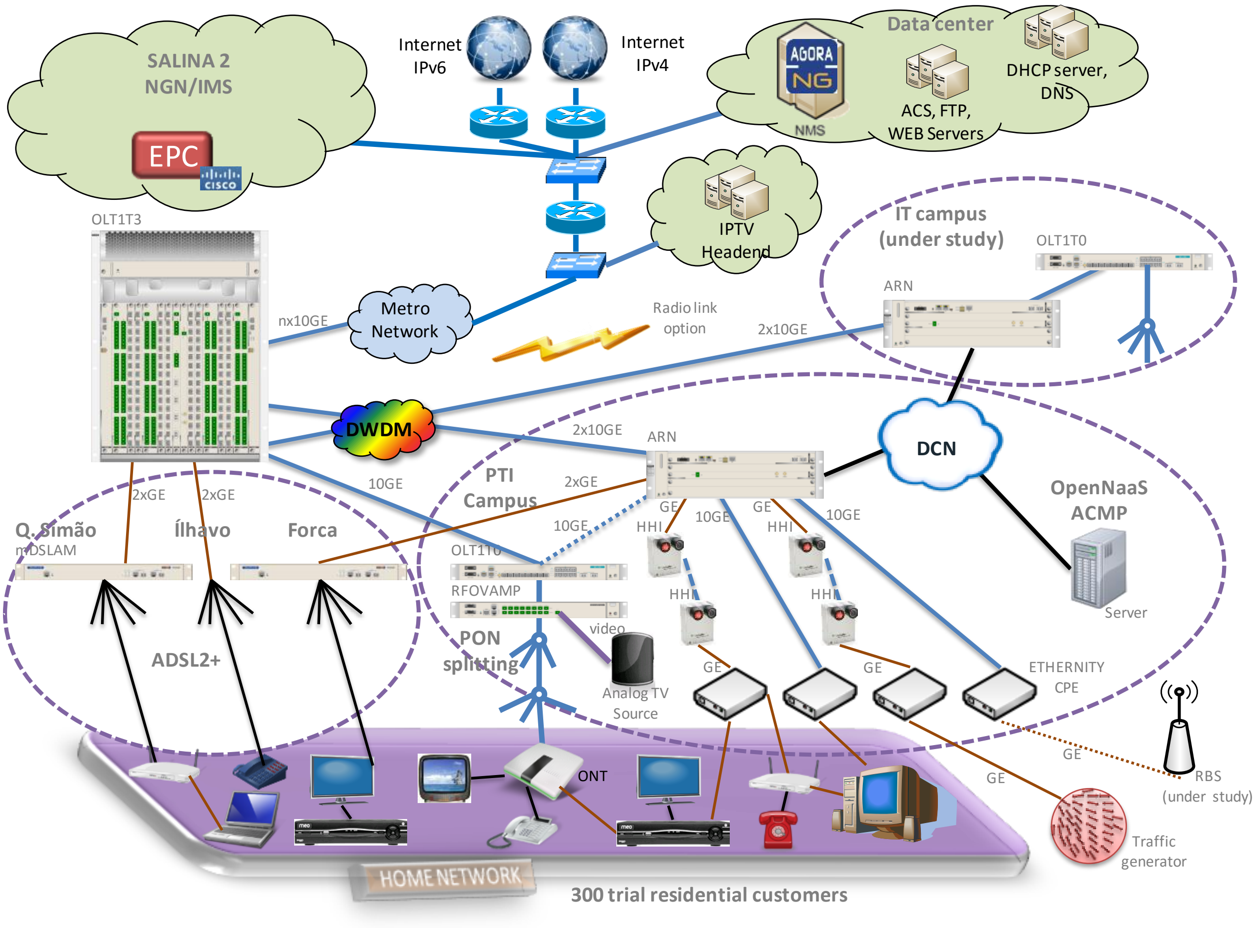

Figure 3: The SODALES DELTA scenario

At this stage, the functionality of the SODALES ARN, as according to the original project specifications and expectations, is relatively minimal. Nevertheless, the demonstrator not only validates the correct working of the SODALES equipment components, but also especially allows for a real-life confirmation of the projected architecture advantages.

2.1.1 PTIN Campus

The PTIN Campus in Aveiro is composed of several buildings. In order to demonstrate the SODALES Architecture, an ARN has been installed on the roof level of the highest building where, on a real network, a RBS would be also tend to be present. From the ARN, several fibres distribute GbE links across several locations of the campus, where the CPEs have been installed, emulating residential customers.

An optical wireless final-drop has also been installed between the roof-top where the ARN is located and one of the other buildings.

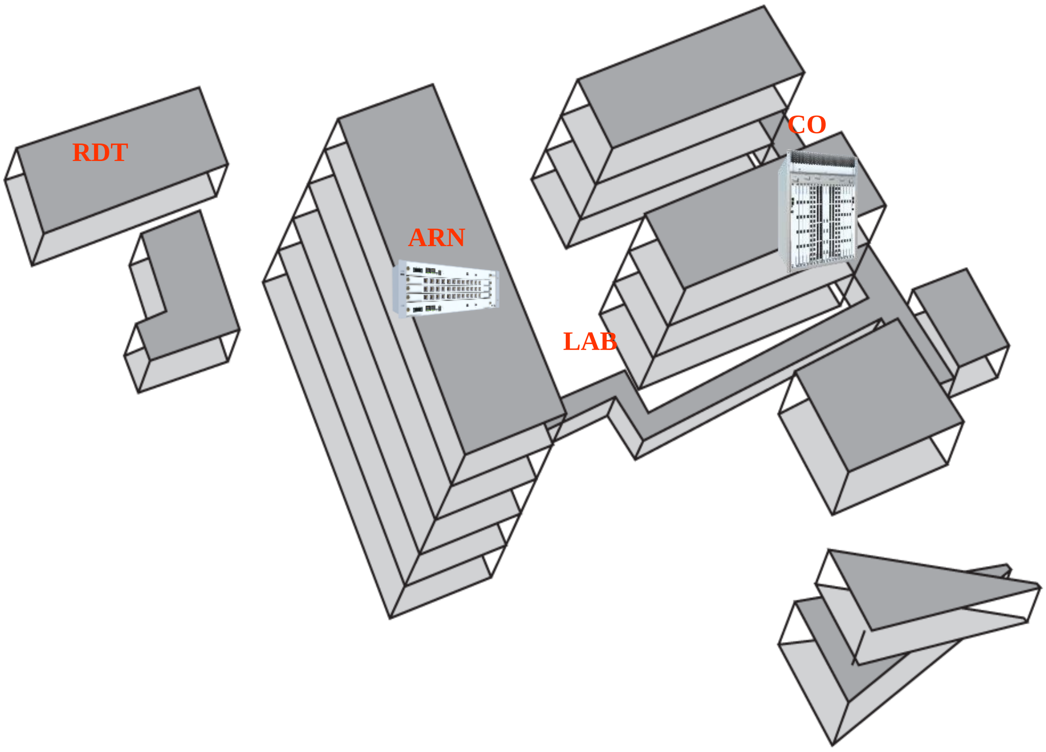

Figure 4: PTIN Campus

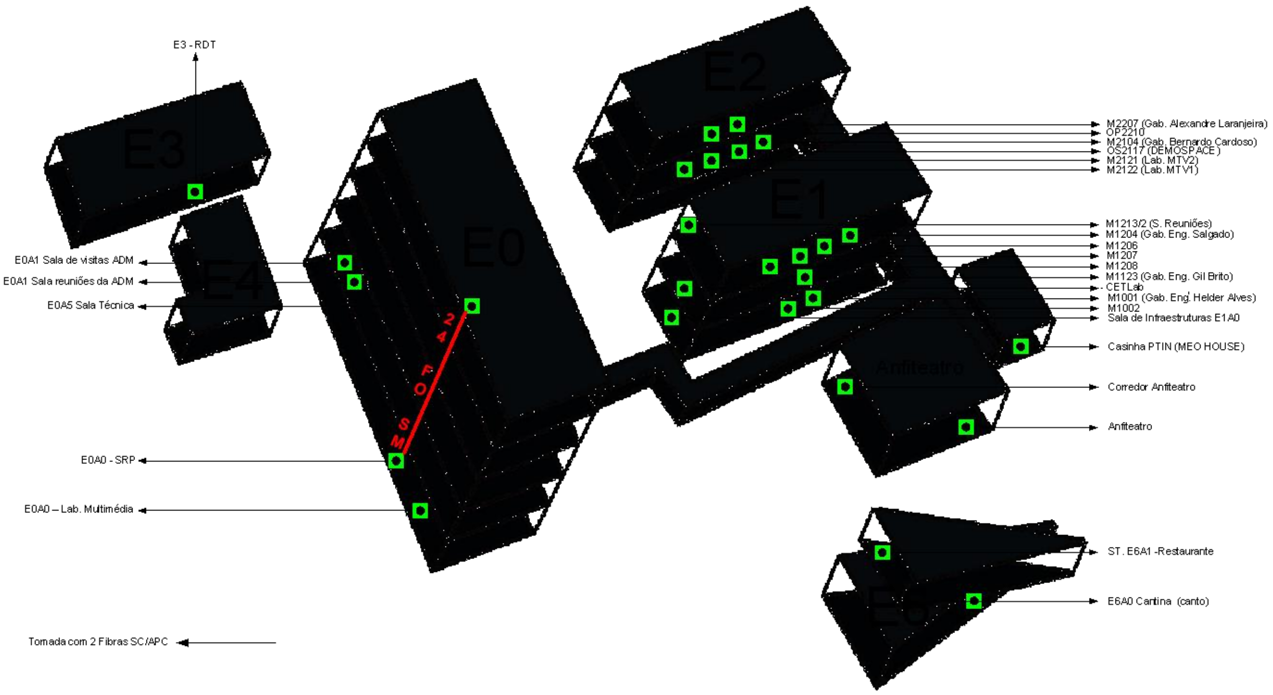

Figure 5: Fiber Accesses

All the fiber access points identified in the previous figure may be used as end-user locations. One special reference is made to the location on the "E3" building, since this is where PTIN has the RDT (Reliability Duration Tests laboratory), where batches of 100 CPEs can be connected to the SODALES for special testing purposes.

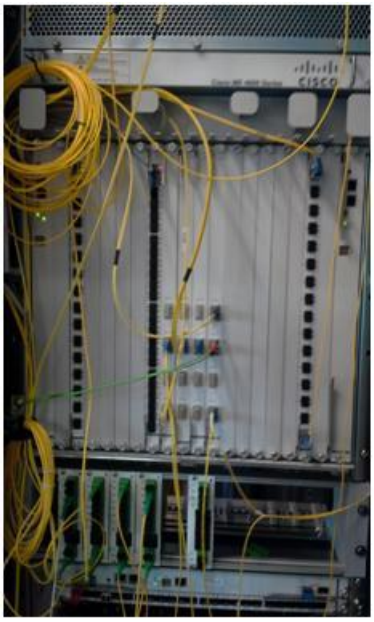

Figure 6: Central Office

Figure 6 presents the Central Office equipment, one high capacity OLT (that features up to several dozens of 10G ports and can go up to 2.2 Tbps of processing power) on the left side, and one Cisco ASR 9000 series that belongs to the DELTA CORE network on the right side.

Figure 7: Roof top Equipment Shelter

Figure 7 presents the roof-top cabinet location where the ARN is installed, on the main building of PTIN campus. From this roof-top, several cables of fibre go down through an internal fibre duct infrastructure (Figure 8) and are terminated on one ODF (Figure 9). From this point, the fibres may be routed to any of the locations presented in Figure 5.

Figure 8: Main building internal fiber duct

Figure 9: Main building ODF

On end of the optical link is installed on the roof of the roof-top shelter on the main building and the other is installed on one of the neighbouring buildings inside the PTIN campus. Figure 10 shows the view from the ARN side of the optical link to the client (CPE). Figure 11 shows the view from the opposite direction. The two ends of the optical link are at about 80 metres distance from each other.

Figure 10: View from the installation location for the optical link (ARN side) to the installation location of client (CPE) side.

Figure 11: View from the installation location for the optical link (client (CPE) side) to the installation location of the ARN side.

2.2 Available Services

The basic SODALES scenario allows for the delivery of the required services for optimal network utilization, as shown in Figure 12.

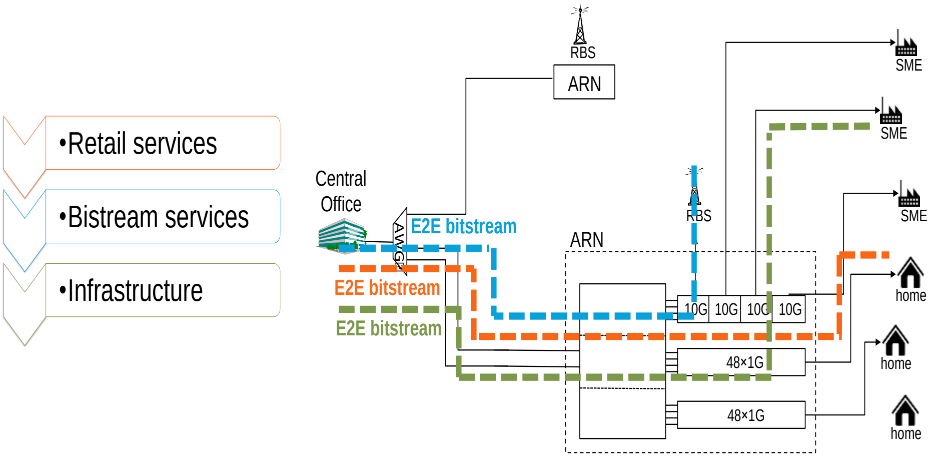

Figure 12: SODALES services

The establishment of the SODADES scenario prototype trials for providing Metro Ethernet Forum (MEF) services, aims at obtaining a real-life validation of the project objectives, such as:

- Creating an open architecture suitable for multiple operators
- Enabling infrastructure sharing via network virtualization
- Using heterogeneous access technologies supporting different physical layers, wired/wireless feeder and access links with Ethernet as a common integration platform
- Providing generic operation and maintenance features
- Employing a software-defined architecture

2.3 Logical Network Details

The DELTA demonstration environment is able to provide user services on a Q-in-Q configuration (one client VLAN per client and service) or on a single VLAN (802.1q) topology, with multiple clients sharing a single service VLAN. The IPTV platform uses a multicast distribution VLAN and a unicast VLAN for Video-on-Demand and for sending the IGMP control messages to the uplink.

Besides this particular aspects of the network, the logical view of the network is presented in Figure 3.

3 SODALES Network Elements Configuration and Service Provision

This chapter describes the different network elements configuration mechanisms for the SODALES demonstrator. The elements enabled through the Open Access management platform are the ARN, and the CPE, which have been the focus of the project over its entire duration.

3.1 Active Remote Node (ARN)

The ARN is the key innovative component in the SODALES architecture, as has been stated in previous documents [1, 2, 3, 4]. It has been developed as an active node in order to provide convergence in the access between wireless and wired access. The architecture, the associated characteristics, and the different features have been well described in the different deliverables [5, 6, 7, 8].

In order to configure the ARN device, the SODALES management plane enables the provisioning of different Ethernet services over the ARN. Basically, the ARN supports the following Ethernet services:

- S-VID – Service VLAN ID. VLAN ID Used to identify an Ethernet service
- C-VID – Client VLAN ID. VLAN ID used to identify a Client within an Ethernet Service

Different types of services are supported, i.e. multicast, unicast, and bitstream. It is worth mentioning at this point that all the services, except for the multicast one, can be stacked or unstacked, i.e. unstacked services use one VLAN ID to forward the traffic – it can have more VLANs but they are considered as payload – at the uplink/network side; while stacked services use two VLAN IDs at the uplink/network side. They are also known as single and double tagging mechanisms.

The description of the supported types of services follows:

- **Unicast:** Bidirectional service. These services are used in Residential applications, where some kinds of packets can be trapped and processed (ex. DHCP, IGMP). If the service is unstacked (N:1) forwarding is based on S-VID+DMAC and MAC Learning is performed, downstream unknown traffic (Multicast/Broadcast/Unknown Unicast) is replicated to all the Clients that belong to the same S-VID. If the service is Stacked (1:1) traffic is forwarded using S-VID + C-VID, so no S+C VLANs can be repeated in the system, no MAC learning is preformed in the Line Card.
- **Bitstream:** Bidirectional service. These services are used in Business applications, where all packets pass transparently through the system. If the service is unstacked (N:1) forwarding is based on S-VID+DMAC and MAC Learning is performed, downstream unknown traffic (Multicast/Broadcast/unknown Unicast) is replicated to all the Clients that belong to the same S-VID. If the service is Stacked (1:1) traffic is forwarded using S-VID + C-VID, so no S+C VLANs can be repeated in the system, no MAC learning is preformed in the Line Card.

- **Multicast:** Unidirectional service, where one frame can be delivered to several destinations. These destinations are controlled by the IGMP Snooping/Proxy at a different service level.

Additionally, for each type of service, different options can be configured:

- **DHCP** – Activates DHCP snooping. With the DHCP snooping enabled the ARN builds a binding table based on the DHCP snooped messages. DHCP relay can also be performed in order to include customer information to the DHCP packets, in the case of DHCPv4, option 82 can be added with the circuit-ID and remote-ID information, in the case of DHCPv6 the client requests are encapsulated in Relay Forward Messages and Option 18 (Interface Id) and Option 37 (Remote ID) can be added. IPSG and DAI only work properly if the DHCP snooping is enabled, because this is the only way the equipment Binding Table can be automatically built.
- **PPPoE –** Activates PPPoED snooping. Used to implement the PPPoE intermediate agent that will add customer information to the PPPoED packets, option 105 that contains the circuit-ID and remote-id information.
- **IGMP** – Activates IGMP snooping. Used to process IGMP packets in order to open/close IPv4 Multicast Streams of the Multicast Services.
- **Multicast Flood** – For Unicast and MAC Bridge services this flag allows or blocks Multicast traffic in this VLAN. It must be enabled if that VLAN has IPv6, Control Protocols (Routing, STP, CDP, among others) if the intention is to drop this kind of packets it must be disabled.

In order to provision services on the ARN, the management plane enables a two-step procedure. First, the network service provisioning: basically a network service must be configured with the desired VLANs, set of uplinks ports and definition of the service type and options (i.e. unicast, multicast, etc.). Then, once the service is set, the client service provisioning is performed: for each client a service must be configured to map the former network service with the desired C-VLAN.

The following interface specifies the XML interface for the network service and client service configuration as enabled in the ARN.

```
REQUEST:
Reading:
/request/operation [@type=”show” AND @entity=”networkService”]/networkService
[@equipmentId=”value” AND @serviceId="value"]


Changing:
/request/operation [@type=”config” AND @entity=”networkService”]/networkService
[@equipmentId=”value” AND @serviceId="value"]/config/networkService[@...]


For Creating:
/request/operation [@type=”create” AND @entity=”networkService”]/networkService
[@equipmentId=”value” AND @serviceId=”value”]/config/networkService[@...]
```

```
For Delete:
/request/operation [@type="remove" AND @entity="networkService"]/networkService
[@equipmentId="value" AND @serviceId="value"]

For attaching members:
/request/operation [@type="attach" AND @entity="networkService/port"]/networkService
[@equipmentId="value" AND @serviceId="value"]/attach/port[@ifIndex="value"]

For detaching members:
/request/operation [@type="detach" AND @entity="networkService/port"]/networkService
[@equipmentId="value" AND @serviceId="value"]/detach/port[@ifIndex="value"]

RESPONSE:
/response/operation/networkServiceList
Or
/response/operation/networkServiceList/oltService
```

On the other hand, the next interface specifies the different available options in the interface for client service configuration, i.e. for the second step required in any complete service configuration.

```
REQUEST:
Reading:
/request/operation [@type="show" AND @entity="clientService"]/clientService
[@equipmentId="value" AND @serviceId="value"]

Changing:
/request/operation [@type="config" AND @entity="clientService"]/clientService
[@equipmentId="value" AND @serviceId="value"]/config/clientService[@...]

For Creating:
/request/operation [@type="create" AND @entity="clientService"]/clientService
[@equipmentId="value" AND @serviceId="value"]/config/clientService[@...]

For Delete:
/request/operation [@type="remove" AND @entity="clientService"]/clientService
[@equipmentId="value" AND @serviceId="value"]

For attaching members:
/request/operation [@type="attach" AND @entity="clientService/port"]/clientService
[@equipmentId="value" AND @serviceId="value"]/attach/port[@ifIndex="value"]
```

```
For detaching members:
/request/operation [@type=”detach” AND @entity=”clientService/port”]/clientService
[@equipmentId=”value” AND @serviceId=”value”]/detach/port[@ifIndex="value"]

RESPONSE:
/response/operation/clientServiceList
Or
/response/operation/clientServiceList/clientService
```

Appendix A contains an example of service configuration through the management plane and the ARN. The example depicts how the interfaces are utilized and what the messages are that are exchanged between the management plane and the ARN when configuring one service.

3.2 Customer Premises Equipment (CPE)

The CPE is the client device in the SODALES equipment. It is the equivalent to the optical network unit in the typical passive optical network (PON) architecture. The CPE that is based on the ENET Fabric Flow Processor family (also known as MEA API)[1] supports Carrier Ethernet switching, traffic management, and OAM functionalities, amongst others.

In order to support the SODALES configuration and service provisioning requirements, we have focused the implementation in the Carrier Ethernet switching functionalities. Due to the lower granularity of the CPE available API, the implemented functionalities for SODALES have been focused and limited to the Port elements, which provide an API set that is responsible for the configuration of the Ingress and Egress ports of the device. Essentially, the ingress port includes definition on how to extract field information from the packets that arrive at that port, policing per port, and control protocol per port; whilst the egress port includes information about queuing, shaping, and editing in four scheduling hierarchies.

The service-provisioning module of the CPE interface mainly allows the user to configure classification and switching of the incoming packets to the Egress cluster and to monitor its performance.

The service management configuration functions are responsible for:

- Creating a new service, which allows the user to configure the input port and tag range to apply the service to, service attributes, output ports/logical port/cluster, and some policing parameters;
- Modifying existing service, which allows the user to modify an existing service, and its parameters;
- Retrieving function, which allows the user to retrieve the service configuration;

[1] Product is the property of Ethernity Networks

- Deleting service, which deactivates and removes the service from the system.

The following snippets contain the header for each one of the operations.

```
Create
MEA_API_Create_Service (Unit_t unit)
Set
MEA_API_Set_Service (Unit_t unit, serviceId id)
Get
MEA_API_Get_Service(serviceId id)
Delete
MEA_API_Delete_Service(serviceId id)
```

Appendix B contains the complete specification of the service provisioning interface and an example on how to create one service through the available APIs.

3.3 Open Access Management Platform

The Open Access management platform is responsible for enabling, and controlling access to the different equipment components (both virtual and physical) considered within the SODALES architecture. All the required operations are performed through the open access management platform. Service configuration, creation of virtual slices, resource monitoring, and user management are the key elements considered in the open access environment of the SODALES project. The first three capabilities of the SODALES management plane have been largely described in the earlier deliverables [1-4].

User management is the last feature of the platform which enables the infrastructure owner to create different user profiles for each one of the customer service providers, as well as to distribute or lease resources (in the form of virtual infrastructures) to each one of them. In order to perform these operations, the management plane contains an implementation of the basic user management functionalities, i.e. create/delete/modify user, as well as assign resources to a given user.

Appendix C describes the interface with the basic operations regarding the User Management.

4 Validation

4.1 Testing Equipment

PTIN has IXIA and Agilent N2X testing equipment that is used for the performance and protocol testing.

Figure 13: IXIA testing equipment

Figure 14: Agilent N2X testing equipment

These items of equipment have 1Gbps and 10Gbps network ports that are used to send and collect traffic on the SODALES demonstration network infrastructure.

4.2 End-User facilities and equipment

The list of equipment used at the “customer premises” includes PCs, laptops, smartphones, IP phones and TVs with set-top-boxes for IPTV service.

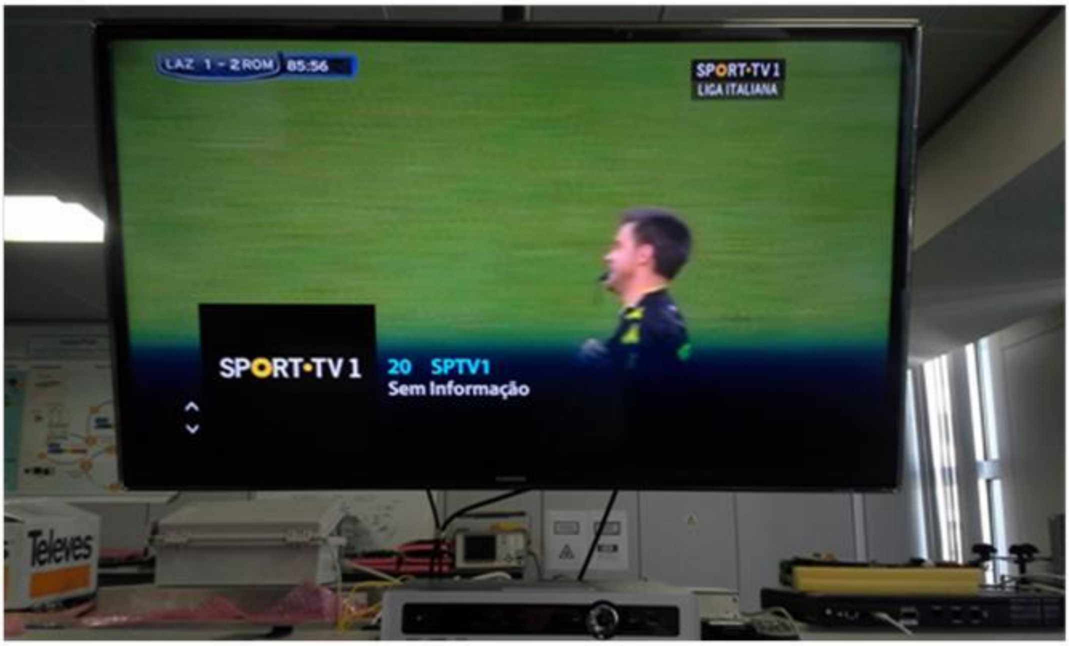

Figure 15: End-user equipment - IPTV Set-top-box and TV.

The IPTV platform used on the DELTA environment is the Ericson MediaRoom (previously Microsoft MediaRoom) and the end-user facilities have certified set-top-boxes for this platform.

4.3 Network Element debug

High level and low level debugging procedures and information are available on the ARN.

Low level debugging procedures and information are available on the CPE.

High level and low debugging procedures and information are available on the Open Access Management Platform.

5 Validation in Centelles

After the validations have been carried out in the DELTA demonstrator in Aveiro, the SODALES equipment and systems will be installed in Centelles, Barcelona, Catalonia, to perform the final SODALES verifications in a commercial environment. Optical wireless links will not be installed, as once the validation will be finished, they will be sent back to HHI.

The idea is to replicate the SODALES demonstrator in a commercial environment to demonstrate its correct performance within a commercial exploitation context. The reason to choose Centelles for this additional pilot is because there is a fibre Open Access network in the town, which is at present operated by two ISPs: Fibra Òptica Centelles and Gurbtec.

5.1 Location and context

The validations that we want to be carried out will consist of the deployment of an ARN at the network Point of Presence (PoP), in order to validate both P2P fibre connections and radio links, together with the SODALES control and management plane. Also, 4G LTE services will be validated, taking advantage of the fixed 4G infrastructure that Fibra Óptica Centelles owns in Collsuspina (a radio tower that covers Osona county which is close to Centelles).

The following picture describes the optical fibre point-to-point and the radio link that will be deployed to validate the SODALES performance in a real production network:

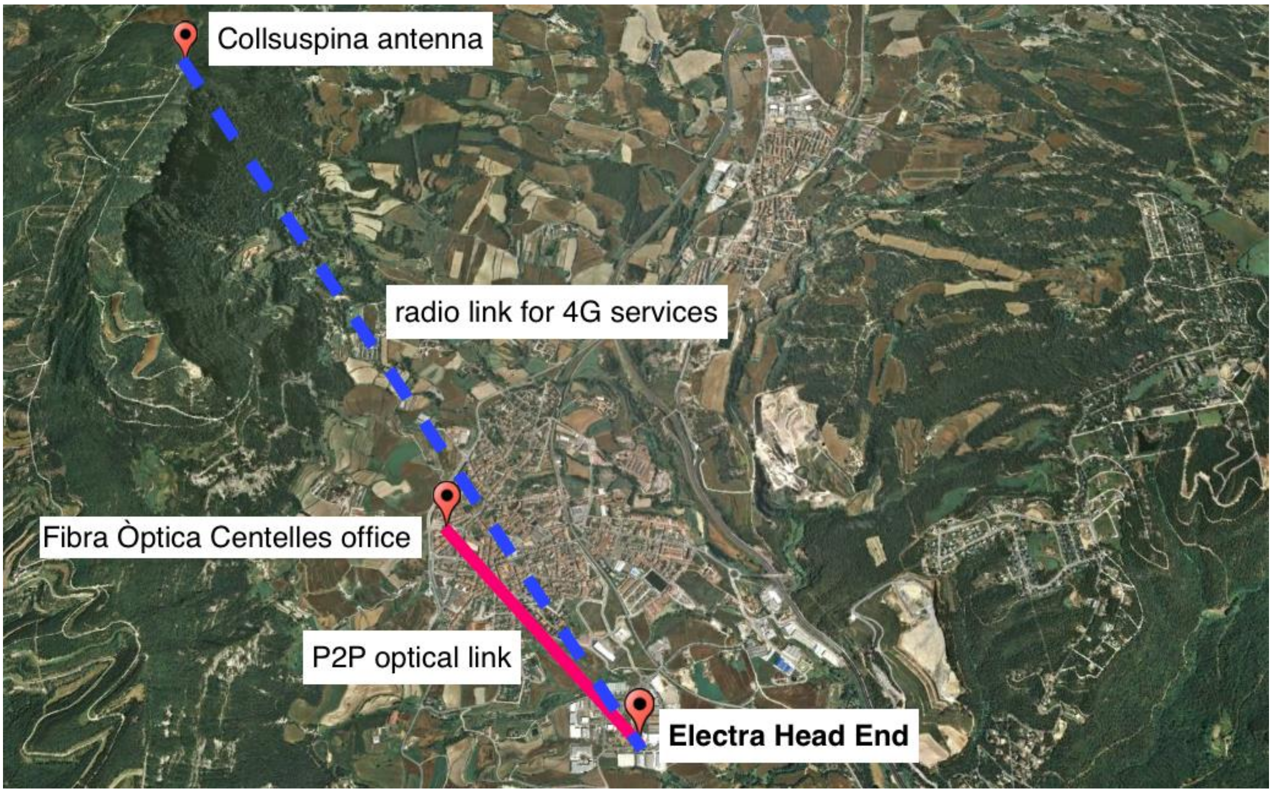

Figure 16: Locations of the validator in Centelles

The network in Centelles is in operation with more than 500 connected users, so the validation traffic will be completely isolated from the production traffic. This will be done by separating the traffic using different VLANs in the common links.

Electra Distribució is the company that operates the Open Access fibre network and provides physical connectivity from its head-end to each of the subscribers. On top of this infrastructure, Fibra Óptica Centelles and Gurbtec offer HSI services.

5.2 Existing equipment

The following figure describes the existing equipment that Fibra Òptica Centelles, the ISP that will help to perform the validation, has in the Electra head-end.

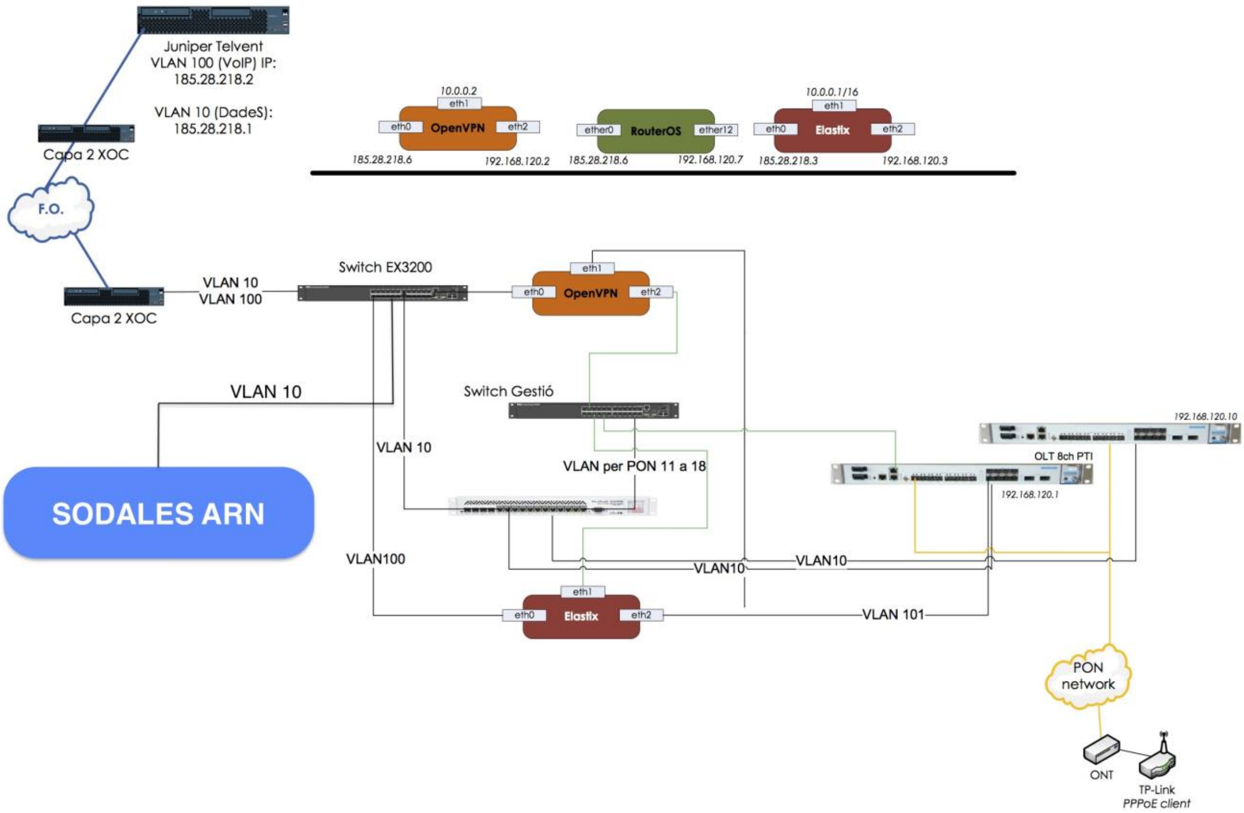

Figure 17: Equipment and interconnections of Fibra Òptica Centelles

The validation will consist in interconnecting an ARN to the aggregation switch (Juniper EX3200) tagging the traffic with VLAN #10, so that it is routed to the core network and Internet Exchange. This configuration also connects the SODALES ARN with the servers located at the head-end.

The servers are deployed on top of a VMWare ESX hypervisor. This will allow the deployment of a validation virtual machine to test the performance of the SODALES network. The virtual machine configuration is presented in the following figure:

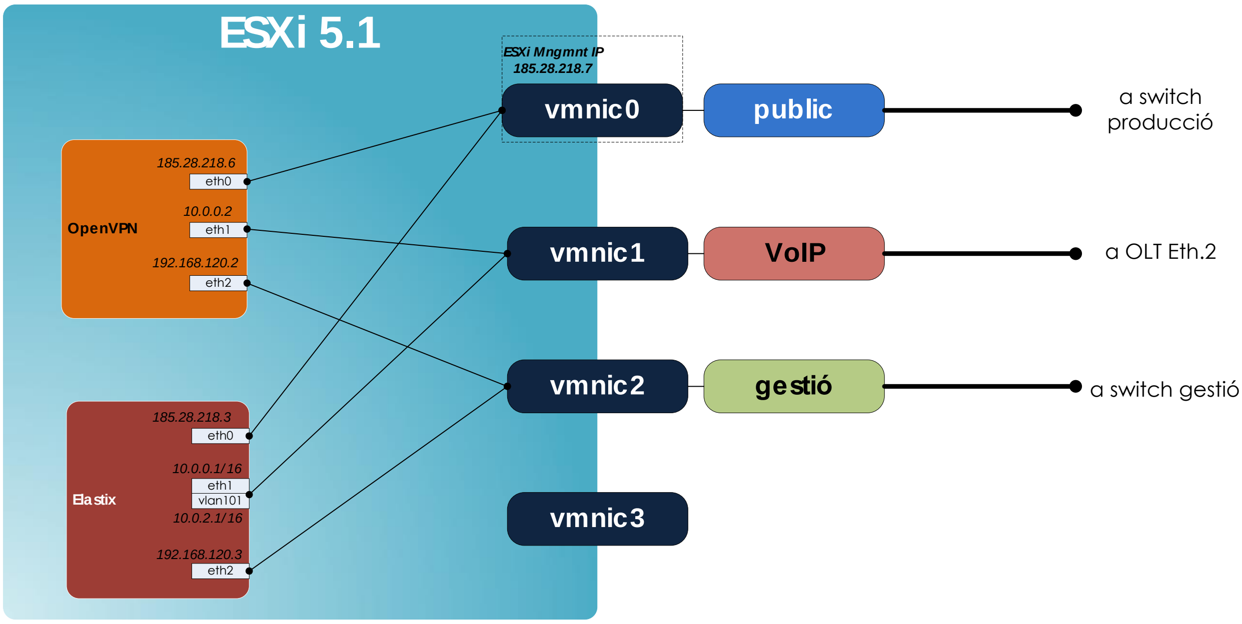

Figure 18: Virtual machine configuration

During the SODALES validation, an additional virtual machine will be deployed to offer video and data services to test the performance of the SODALES connections.

5.3 Validation plan

The validation plan will consist, from the physical layer perspective, on:

1. Installing the ARN at the network head-end
2. Installing a business CPE in the Fibra Optica Centelles office
3. Interconnecting the ARN with the radio infrastructure in Collsuspina

5.3.1 Connectivity tests

Once the connectivity is established, performance tests will be carried out. These tests will consist of:

- Network connectivity verification: end-to-end network throughput and delay. The validation server (iperf) will be installed on a virtual machine in the Fibra Optica Centelles virtual server infrastructure.
- Video transmission: the video pump will be deployed in the virtual server infrastructure of Fibra Óptic Centelles in order to validate correct transmission along the SODALES equipment.
- Internet access: the SODALES system will be connected to the internet so HSI service will also be validated.

5.3.2 Control and management tests

Once the physical and service layers are verified, the control and management plane will also be validated.

The validation environment will consist on two virtual ISPs, one of which will offer HSI (OP#1) and the other video services (OP#2)

Tests will consist on the following:

- Service provision of HSI to the business CPE located in the Fibra Optica Centelles office by OP#1
- Video Service provision to the business CPE located in the Fibra Optica Centelles office by OP#2
- De-provisioning of video services to the business CPE located in the Fibra Optica Centelles office from OP#2
- Video Service re-provisioning to the business CPE located in the Fibra Optica Centelles office by OP#2
- De-provisioning of video services to the business CPE located in the Fibra Optica Centelles office from OP#2
- De-provisioning of HSI to the business CPE located in the Fibra Optica Centelles office from OP#1

5.3.3 Validation checklist

The following table presents the validation checklist that will be filled during the validation in Centelles:

Test	Date and time of the test	OK	KO
L2 Network connectivity			
Video service connectivity			
HSI connectivity			
Service provision of HSI from OP#1			
Video provision by OP#2			
Video de-provision from OP#2			
Video re-provision by OP#2			
Video de-provision from OP#2			
De-provision of HSI from OP#1			

6 Conclusions

The SODALES project systems developments have culminated in the establishment of a system demonstrator that is well capable of proving the innovative concepts proposed in the original SODALES project specification.

The objective of the validators is to demonstrate the correct performance of the SODALES platform, both from the transmission and operation perspectives, and within a real commercial environment.

The validators will be implemented in order to bring the technical developments close to the market. Actually, the validation taking place in Centelles will be commercially exploited once the project is over, so we are already guaranteed that the SODALES project is ending up with real, commercial exploitation results.

Finally, in this deliverable we have discussed the service provision and configuration of each network element, through the Open Access Management Platform. Additionally, the demonstration network infrastructure in Aveiro and particular locations has been presented and described. The deliverable has concluded with the presentation of the testing facilities/equipment that are being used to guarantee that the SODALES services are being delivered according to the designated specifications, and providing the envisaged quality of experience to the end-user.

7 References

[1] FP7-ICT-8 318600 SODALES D1.2 “SODALES ARN modelling”

[2] FP7-ICT-8 318600 SODALES D2.1 “ARN Design and Interfacing”

[3] FP7-ICT-8 318600 SODALES D2.2 “Demarcation Unit Design and WDM and radio interface definition”

[4] FP7-ICT-8 318600 SODALES D2.3 “SODALES aggregation interface design”

[5] FP7-ICT-8 318600 SODALES D1.1 “SODALES architecture, service catalogue and network specifications”

[6] FP7-ICT-8 318600 SODALES D1.3 “SODALES OAM and Control Plane parameters for Open Access Networks”

[7] FP7-ICT-8 318600 SODALES D3.1 “Control Plane Service Requirements”

[8] FP7-ICT-8 318600 SODALES D3.4 “Control and Management Plane Software”

Appendix A: Service Configuration through the ARN

This appendix contains an example of an Ethernet service configuration through the interface implemented over the ARN by means of the XML messages sent by the SODALES management plane. The figure below contains the scenario where the services will be created.

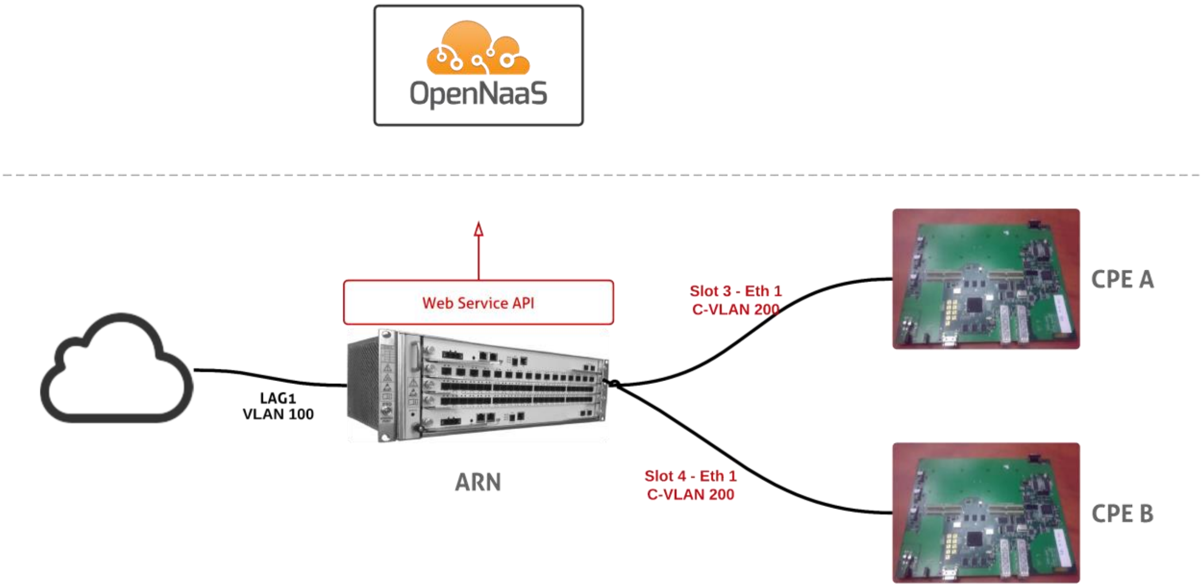

Figure 19: Service configuration scenario

Here, the figure contains the SODALES management plane on the upper part, based on the Open NaaS framework. On the bottom part, are located one ARN and two client CPEs. The objective of this example is to create one Ethernet network service with S-VLAN 100, and attach to this network service two different client services, each one in one of the two 48 GbE slots, with C-VLAN 200. It is worth mentioning that this Appendix only depicts the configuration examples for the ARN.

First step, is to create the network service. The following XML message corresponds to the network service creation in such an environment (OpenNaaS and the ARN).

```
<?xml version="1.0" encoding="utf-8"?>
<request>
<!-- -->
<!-- CREATE and CONFIG are equal, excepted for the operation type -->
<!-- -->
<operation type="create" entity="networkService">
      <networkService equipmentId="0" id="3" admin="2" name="Unicast service
      1" type="0" uplinkVlanId="100" uniVlanId="200" > </networkService>
</operation>
</request>
```

Whereby the values for the attribute type can be 0 for unicast, 1 for multicast, and 2 for bitstream. The uplink VLAN ID and uni VLAN ID represent the VLAN ID in the uplink side and the downlink side of the network service. The next step consists of adding the corresponding ports to the network service, i.e. the LAG 1 and the two 48 GbE cards. The following message contains such an operation.

```
<?xml version="1.0" encoding="utf-8"?> <request>
<!-- Only one interface per operation. Use several operations to attach several
interfaces -->
<operation token='1' type="attach" entity="networkService">
  <networkService equipmentId="0" id="1">
   <attach>
    <interface cardId='0' interfaceId='8781824' role='1' />
   </attach>
  </networkService>
</operation>
<operation token='2' type="attach" entity="networkService">
  <networkService equipmentId="0" id="1"> <attach>
   <interface cardId='3' interfaceId='58982400' role='2' /> </attach>
  </networkService>
</operation>
<operation token='3' type="attach" entity="networkService">
  <networkService equipmentId="0" id="1">
   <attach>
    <interface cardId='4' interfaceId='75759616' role='2' />
   </attach>
  </networkService>
</operation>
</request>
```

The XML message contains three operations. As described in [D3.4], the interface has been implemented to support such messages. The uplink interface (LAG 1) must have the role set to 1, which means root of the service, while downlink interfaces must have role set to 2, which identifies the leaf of the service. The first operation, with token 1, adds the uplink interface to the network service, while the second and third operations add the two 48 GbE cards to the network service. The interfaceId field for these types of cards must be the id of the first internal Ethernet interface (intEth 1).

Finally, there is the need to activate the corresponding network service.

```
<?xml version="1.0" encoding="utf-8"?>
<request>
 <operation type="config" entity="networkService">
```

```
   <networkService equipmentId="0" id="1" admin="1">
   </networkService>
 </operation>
</request>
```

These three steps are the first stage in order to provision an Ethernet service through the SODALES management plane. The figure below summarizes them.

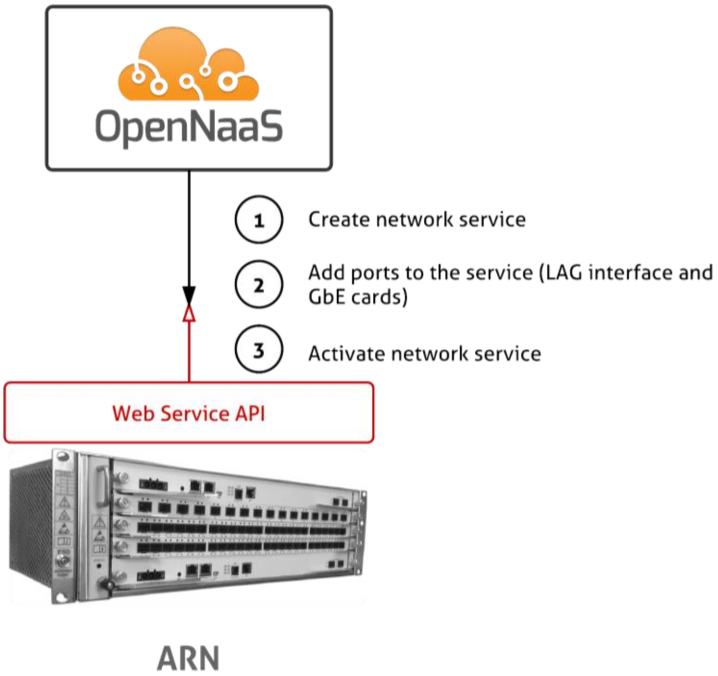

Figure 20: Network service creation and configuration summary

In order to complete the service provisioning, there is still the need to create the client services and attach to them the corresponding Ethernet interfaces in the 48 GbE cards. Thus, it will follow a similar process and message workflow between the management plane and the ARN itself. The figure below summarizes this process.

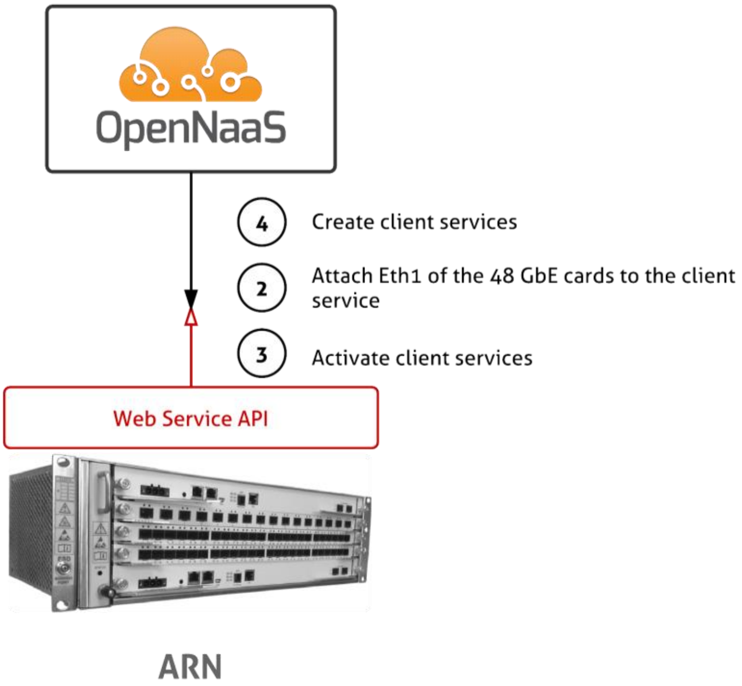

Figure 21: Client service creation and configuration summary

The first step (depicted 4 in the figure) is to create the client services. This is done with the following message.

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <operation type="create" entity="clientService">
       <clientService equipmentId="0" >
         <create>
            <clientService networkServiceId="1" admin="2" upStreamId="0"
                    name="client service 1" uniVlan="200"   >
            </clientService>
         </create>
      </clientService>
  </operation><operation type="create" entity="clientService">
        <clientService equipmentId="0" >
         <create>
           <clientService networkServiceId="1" admin="2" upStreamId="0" name="client
                   service 2" uniVlan="200" >
            </clientService>
         </create>
      </clientService>
  </operation>
</request>
```

In case the *uniVlan* attribute is specified in the previous message, then the UNI C-TAG of the Network Service will be overridden for that particular client service. An Ethernet profile Id can be specified in the *upStreamId*, whereby zero means no profile. Next logical step is to attach the corresponding Ethernet interfaces of the 48 GbE cards to the client services. We can do this with the same message for both client services.

```
<?xml version="1.0" encoding="utf-8"?> <request>
<operation token='1' type="attach" entity="clientService"> <clientService
equipmentId="0" id="1">
            <attach>
                <port portIfIndex="50397184"/>
            </attach>
        </clientService>
</operation><operation token='2' type="attach" entity="clientService">
<clientService equipmentId="0" id="2"> <attach>
                <port portIfIndex="67174400"/>
            </attach>
        </clientService>
    </operation>
</request>
```

Finally, there is only the need to activate the client services attached to the previously created network service.

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <operation type="config" entity="clientService">
     <clientService equipmentId="0" id="1" admin='1' >
     </clientService>
  </operation>
  <operation type="config" entity="clientService">
     <clientService equipmentId="0" id="2" admin='1' >
     </clientService>
  </operation>
</request>
```

Appendix B: Service configuration on the CPE

This Appendix contain the description of the interface on the SODALES management plane that enables one to run the service provisioning module of the CPE, as described in Section 3. The interface contains the basic CRUD operations over the service entities. The operations are specified in the following text.

```
Create
MEA_Status MEA_API_Create_Service (MEA_Unit_t, MEA_Service_t* o_serviceId);
```

Whereby unit is an integer value for the device instance to affect as defined in the corresponding data structures, and O_serviceId is the pointer to the data type that returns the service ID for all the future accesses to this service.

```
Set
MEA_API_Set_Service (Unit_t unit, serviceId id)
```

Whereby unit is an integer value for the device instance to affect as defined in the corresponding data structures, and id is an unsigned integer value for the existing service to modify.

```
Get
MEA_API_Get_Service(Unit_t unit, serviceId id)
```

Whereby unit is an integer value for the device instance to affect and id is an unsigned integer value for the existing service to get the configuration parameters.

```
Delete
MEA_API_Delete_Service(serviceId id)
```

Whereby id is the identifier of the service to be removed.

In order to utilize these functions in the device, the SODALES management plane utilizes an abstraction of the interface, available through REST technology. The operations supported are basically the same, plus the monitoring management. Thus, the northbound interface supports service creation, retrieving, and removal; special policy creation, as well as port management information.

The interface operations are listed below.

```
package net.i2cat.enet38xx.client.nortbound;

import java.util.List;

import net.i2cat.enet38xx.client.model.MeaEgressPortInfo;
import net.i2cat.enet38xx.client.model.MeaGetServiceVlanId;
import net.i2cat.enet38xx.client.model.MeaIngressPortInfo;
```

```
import net.i2cat.enet38xx.client.model.MeaPmCounter;
import net.i2cat.enet38xx.client.model.MeaPolicerAllProfiles;
import net.i2cat.enet38xx.client.model.MeaPolicerProfile;
import net.i2cat.enet38xx.client.model.MeaPortMapping;

/**
 *
 * @author Isart Canyameres Gimenez (i2cat)
 *
 */
public interface Enet38xxClientAPI {

	//port management

	/**
	 * Retrieves all ports in given unit.
	 *
	 * @param unitId Unit affected
	 * @return Collection of port ids in given unit, elements in the collection
contain both the unitid and the portid
	 */
    MeaPortMapping getAllPorts(String unitId);
    MeaIngressPortInfo getIngressPortInfo(String unitId, String portId);
    MeaEgressPortInfo getEgressPortInfoString(String unitId, String portId);
    void configureIngressPort(MeaIngressPortInfo ingressPortInfo);
    void configureEgressPort(MeaEgressPortInfo egressPortInfo);

    // policer profiles management

    /**
     * Retrieves the id of all policer profiles configured in given unit
     *
     * @param unitId Unit affected
     * @return Collection of policer profile ids in given unit.
     */
    MeaPolicerAllProfiles getPolicerProfiles(String unitId);
    MeaPolicerProfile getPolicerProfile(String unitId, String profileId);
    MeaPolicerProfile createPolicerProfile(String unitId, String profileId,
MeaPolicerProfile profile);
    void removePolicerProfile(String unitId, String profileId);

    // service vlan management
```

```
    /**
     * Retrieves the id of all services vlan configured in given unit
     *
     * @param unitId
     * @return Collection of services vlan ids in given unit.
     */
    List<String> getServicesVlan(String unitId);
    MeaGetServiceVlanId getServiceVlan(String unitId, String serviceId);
    MeaGetServiceVlanId createServiceVlan(String unitId, MeaGetServiceVlanId
serviceVlanDetail);
    void removeServiceVlan(String unitId, String serviceId);

    // performance monitoring management

    /**
     * Retrieves the id of all performance monitoring counters configured in given
unit
     *
     * @param unitId unit affected
     * @return Collection of performance monitoring counter ids in given unit.
     */
    List<String> getPerformanceMonitoringCounters(String unitId);
    MeaPmCounter getPerformanceMonitoringCounter(String unitId, String pmId);
    void clearPerformanceMonitoringCounter(String unitId, String pmId);


}
```

In the case that the infrastructure owner or the service provider wants to configure the CPE as depicted in the figure below, it is required to send an XML message to create a service with an inner and outer vlan id configuration as well as the corresponding input and output ports. In order to connect this service at the data plane level with the client services created in the ARN, it is required that the same vlan id is configured in both edges of the link.

Figure 22: CPE configuration example

Service creation


```
<?xml version="1.0" encoding="UTF-8"?>
<meaGetServiceVlanId xmlns="http://www.ethernitynet.com/enet/ServiceParams">
  <serviceDataBase>
    <unit>0</unit>
    <serviceId>6</serviceId>
    <policerId>2</policerId>
    <pmId>3</pmId>
    <eIngressType>1</eIngressType>
    <outer_vlanId>10</outer_vlanId>
    <inner_vlanId>0</inner_vlanId>
    <src_port>105</src_port>
    <serviceOutputPort>
      <cluster>
        <clusterId>104</clusterId>
      </cluster>
    </serviceOutputPort>
    <vlanEdit_flowtype>2</vlanEdit_flowtype>
    <vlanEdit_outer_command>3</vlanEdit_outer_command>
    <vlanEdit_outer_vlan>20</vlanEdit_outer_vlan>
    <vlanEdit_inner_command>0</vlanEdit_inner_command>
    <vlanEdit_inner_vlan>0</vlanEdit_inner_vlan>
  </serviceDataBase>
</meaGetServiceVlanId>
```

Appendix C: SODALES User Management

This Appendix describes the two interfaces implemented in the management plane in order to perform all the required user management operations, i.e. user management and resource assignation between users.

User Management

```
package org.mqnaas.core.api.user;

import org.mqnaas.core.api.ICapability;

/**
 * <p>
 * Capability managing the list of users of the MQNaaS framework.
 * </p>
 * <p>
 * First draft of the capability defines users as {@link String}s representing the
username.
 * </p>
 *
 * @author Adrian Rosello Rey (i2CAT)
 * @author Isart Canyameres (i2CAT)
 *
 */
public interface IUserManagement extends ICapability {

	/**
	 * Return all the usernames of the framework users.
	 *
	 * @return {@link User} wrapper class, containing the list of all usernames of
the MQNaaS framework.
	 */
	Users getUsers();

	/**
	 * Creates one User with the profile {@profile}
	.*
	.* @return {@link String} the internal ID of the User in the MQNaaS framework
	.*/
	public String createUser(UserProfile profile) throws IllegalStateException;
```

```
	/**
	 * Modifies the User identified by {@userId} with the new profile {@profile}
	.*
	.* @return @boolean true if the modification succeeds, false otherwise
	.*/
	public String modifyUser(String userId, UserProfile profile);

	/**
	 * Removes the User identified by {@userId}}
	.*
	.* @return @boolean true if the removal succeeds, false otherwise
	.*/
	public String deleteUser(String userId) throws IllegalStateException;
}
```

Resource assignation

```
package org.mqnaas.core.api.user;

import org.mqnaas.core.api.ICapability;
import org.mqnaas.core.api.IRootResource;

/**
 * <p>
 * Capability managing the assignment of resources to specific users.
 * </p>
 *
 * @author Adrian Rosello Rey (i2CAT)
 * @author Isart Canyameres (i2CAT)
 *
 */
public interface IResourceAssignation extends ICapability {

	/**
	 * Returns a list of the ids of the {@link IRootResource}s assigned to a
specific user.
	 *
	 * @param username
	 *            Id of a framework's user.
	 * @return Wrapper class providing the ids of the resources assigned to given
user.
	 */
```

```
	Resources getByUser(String username);

	/**
	 * Assign a specific {@link IRootResource} to a specific user. Only unassigned resources can be assigned to a user.
	 * 
	 * @param resourceId
	 *            Id of the resource to be assigned.
	 * @param username
	 *            Id of the user that will be bound to the resource.
	 * @throws IllegalStateException
	 *             If the resource is already assigned to this or another user.
	 */
	void assign(String resourceId, String username) throws IllegalStateException;

	/**
	 * Unassign a specific {@link IRootResource} from a specific user. A resource can only be unassigned if it's already assigned to that user.
	 * 
	 * @param resourceId
	 *            Id of the resource to be unassigned.
	 * @param username
	 *            Id of the user the resource was bound to.
	 * @throws IllegalStateException
	 *             If given resource is not assigned to specified user, or it has been not assigned to any user yet.
	 */
	void unassign(String resourceId, String username) throws IllegalStateException;

}
```