FUTURE COMMUNICATION ARCHITECTURE FOR MOBILE CLOUD SERVICES

Acronym: MobileCloud Networking
Project No: 318109
Integrated Project
FP7-ICT-2011-8
Duration: 2012/11/01-2015/09/30

D6.2 Initial Report on Testbeds, Experimentation, and Evaluation

Type	Report
Deliverable No:	6.2
Workpackage:	6
Leading partner:	ONE
Author(s):	David Palma, Paulo Simões (Editors), list of authors overleaf
Dissemination level:	Public
Status:	Final
Date:	31 October 2014
Version:	1.0

MCN
Mobile Cloud Networking

List of Authors

- Bruno Sousa (ONE)
- David Palma (ONE)
- Luís Cordeiro (ONE)
- Paulo Simões (ONE)
- André Gomes (UBERN/ONE)
- Zhongliang Zhao (UBERN)
- Mohammadjavad Valipoor (STT)
- Negar Sehatbakhsh (STT)
- Santiago Ruiz (STT)
- Yago Ruiz (STT)
- Aiko Pras (UT)
- Morteza Karimzadeh (UT)
- Luuk Hendriks (UT)
- Alex Georgiev (CS)
- Peter Gray (CS)
- Marius Corici (FHG)
- Alexandru Russu (FHG)
- João Soares (PTInS)
- Carlos Parada (PTInS)
- Giuseppe Carella (TUB)
- Paolo Secondo Crosta (ITALTEL)
- Andrea Marcarini (ITALTEL)

Reviewers:

- Andy Edmonds (ZHAW)
- Paolo Crosta (ITALTEL)
- Uwe Riss (SAP)

Versioning and contribution history

Version	Description	Contributors
0.1	First Draft Version	ONE, STT
0.2	Core update	ONE, STT
0.3	Updates on Integration and Evaluation	ONE, FHG, PTInS
0.4	Pre-review version	ONE, CS
0.5	Peer review ready version	ONE
0.6	Peer reviewed	ONE, ZHAW, ITALTEL
0.7	GA review ready version	ONE, FHG, STT, CS
0.8	Final Edits	ONE
0.9	GA reviewed	SAP
1.0	Final version ready for submission	ONE

Executive Summary

The goal of this second deliverable of Work Package 6 (WP6) is to highlight the developments regarding the Proof-of-Concept prototypes and demonstrators for the MobileCloud Networking (MCN) project, providing a fine-grained analysis of the current status and future planning for upcoming activities.

Details about available testbeds that will support the MCN demonstrators and their evaluation are also presented, addressing the main developments towards an integrated testbed between the different resources available by each partner. This work is focused on providing a distributed testbed for performing the necessary MCN experimentation and evaluation activities. An analysis of service-specific requirements for testbeds is also presented.

Additionally this deliverable presents the current status of the defined MCN Integrated Prototype System along with the progress towards the evaluation and experimentation procedures to be conducted. A list of integration activities and the goals for upcoming milestones are also presented.

Finally, in accordance with the objectives of WP6, which consist in guaranteeing that a functional prototype is properly achieved and evaluated, this deliverable provides an updated roadmap and delivery milestones of the upcoming integration and evaluation activities, presenting also a first definition of the relevant key performance indicators and some preliminary results for some services.

Table of contents

EXECUTIVE SUMMARY 4
TABLE OF CONTENTS 5
TABLE OF FIGURES 9
TABLE OF TABLES 9
ACRONYMS 11
1 INTRODUCTION 14
1.1 MOTIVATION, OBJECTIVES AND SCOPE 14
1.2 STRUCTURE OF THE DOCUMENT 14
2 PROOF-OF-CONCEPT AND PROTOTYPES 15
2.1 INTRODUCTION 15
2.2 DSS POC STATUS 15
2.2.1 DSS Requirements per Service for M24 16
2.2.2 Implementation 17
2.2.3 Demo Scenario for DSS (M24) 19
2.3 IMS POC STATUS 22
2.3.1 General Status 22
2.3.2 IMS Requirements per Service for M24 22
2.3.3 Implementation 23
2.3.4 Demo Scenario for IMS (M24) 24
2.4 ROADMAP AND OPEN ISSUES 27
2.4.1 DSS 27
2.4.2 IMS 27
3 TESTBED PLANNING 29
3.1 INTRODUCTION 29
3.2 TESTBED CHARACTERISTICS AND CAPACITY 30
3.2.1 Matching Requirements to Testbeds 30
3.2.2 Integrated Testbed Architecture 31
3.2.3 Available Resources 31
3.2.4 Access to Cloudsigma's Testbed 32
3.3 RANAAS – OPENAIRINTERFACE (OAI) 33
3.3.1 OpenAirInterface on CloudSigma Infrastructure 33
3.4 EPCAAS AND IMSAAS TESTBED REQUIREMENTS 35
3.5 DSSAAS TESTBED REQUIREMENTS 37
3.6 MAINTENANCE AND OPERATION 38
3.6.1 Usage/Deployment 38
3.6.2 Stress-test Plans and Scalability 38
3.6.3 Maintenance 39
3.6.4 Connectivity 39
3.7 ROADMAP AND OPEN ISSUES 39
4 FIRST INTEGRATION OF COMPONENTS 40
4.1 INTRODUCTION 40
4.2 INTEGRATION STATUS 40
4.2.1 Deviations from the Prototype's Architecture 41
4.2.2 MCN Services Integration 42
4.2.3 Support Services Integration 43
4.2.4 MCN SDK 43
4.3 DEFINED INTERFACES 44
4.4 ROADMAP AND OPEN ISSUES 47

5 EVALUATION PLANS 48
5.1 INTRODUCTION 48
5.2 METHODOLOGY AND EVALUATION TECHNIQUE 49
5.3 EVALUATION AREAS, IMPACTS AND INDICATORS 50
5.3.1 Requirements and Strategic Evaluation Objectives 50
5.3.2 Definition of Evaluation Areas, Impacts and Selection of Indicators 50
5.3.3 Definition of Success and Targets 51
5.3.4 Definition of Evaluation Technique 51
5.3.5 Set-up of Data Source and Data Collection 51
5.3.6 Data Analysis / Identification of Success / Impact Analysis 51
5.4 ROADMAP AND OPEN ISSUES 51
6 SUMMARY AND OUTLOOK 53
A APPENDIX A: UPDATED TESTBED DESCRIPTIONS 54
A.1 FRAUNHOFER/TUB 54
A.2 PTINS 55
A.3 FT/ORANGE 56
A.4 CLOUDSIGMA 57
A.5 NEC 59
A.6 INTEL 59
A.7 TI 60
A.8 UTWENTE 62
A.9 UBERN 63
A.10 BT 65
A.11 ZHAW 65
A.12 ONE 66
A.13 STT 68
B APPENDIX B: DETAILED EVALUATION PLANS 70
B.1 DSS 70
B.1.1 Requirements and Strategic Evaluation Objectives 70
B.1.2 Definition of Evaluation Areas, Impacts and Selection of Indicators 70
B.1.3 Definition of success and targets 71
B.1.4 Definition of evaluation technique 71
B.2 EPC - PGW, SGW, MME, HSS 72
B.2.1 Requirements and Strategic Evaluation Objectives 72
B.2.2 Definition of Evaluation Areas, Impacts and Selection of Indicators 73
B.2.3 Definition of Success and Targets 74
B.2.4 Definition of Evaluation Technique 75
B.3 EPC - ANDSF 76
B.3.1 Requirements and Strategic Evaluation Objectives 76
B.3.2 Definition of Evaluation Areas, Impacts and Selection of Indicators 77
B.3.3 Definition of Success and Targets 78
B.3.4 Definition of Evaluation Technique 79
B.4 IMS 80
B.4.1 Requirements and Strategic Evaluation Objectives 80
B.4.2 Definition of Evaluation Areas, Impacts and Selection of Indicators 81
B.4.3 Definition of Success and Targets 82
B.4.4 Definition of Evaluation Technique 83
B.5 RAN 84
B.5.1 Requirements and Strategic Evaluation Objectives 84
B.5.2 Definition of Evaluation Areas, Impacts and Selection of Indicators 85
B.5.3 Definition of Success and Targets 87
B.5.4 Definition of Evaluation Technique 88
B.5.5 Set up of Data Source and Data Collection 89
B.5.6 Data Analysis / Identification of Success / Impact analysis 89
B.6 CDN 90

B.6.1 Requirements and Strategic Evaluation Objectives ... 90
B.6.2 Definition of Evaluation Areas, Impacts and Selection of Indicators ... 90
B.6.3 Definition of Success and Targets ... 91
B.6.4 Definition of Evaluation Technique ... 92
B.7 ICN ... 93
B.7.1 Requirements and Strategic Evaluation Objectives ... 93
B.7.2 Definition of Evaluation Areas, Impacts and Selection of Indicators ... 94
B.7.3 Definition of Success and Targets ... 95
B.7.4 Definition of Evaluation Technique ... 96
B.7.5 Set up of Data Source and Data Collection ... 98
B.7.6 Data Analysis / Identification of Success / Impact Analysis ... 98
B.8 INFRASTRUCTURE ... 99
B.8.1 Requirements and Strategic Evaluation Objectives ... 100
B.8.2 Definition of Evaluation Areas, Impacts and Selection of Indicators ... 100
B.8.3 Definition of Success and Targets ... 100
B.8.4 Definition of Evaluation Technique ... 101
B.9 MONITORING ... 101
B.9.1 Requirements and Strategic Evaluation Objectives ... 101
B.9.2 Definition of Evaluation Areas, Impacts and Selection of Indicators ... 102
B.9.3 Definition of Success and Targets ... 103
B.9.4 Definition of Evaluation Technique ... 104
B.10 ANALYTICS ... 105
B.10.1 Requirements and Strategic Evaluation Objectives ... 106
B.10.2 Definition of Evaluation Areas, Impacts and Selection of Indicators ... 106
B.10.3 Definition of Success and Targets ... 106
B.10.4 Definition of Evaluation Technique ... 106
B.11 MOB ... 106
B.11.1 Requirements and Strategic Evaluation Objectives ... 107
B.11.2 Definition of Evaluation Areas, Impacts and Selection of Indicators ... 107
B.11.3 Definition of Success and Targets ... 107
B.11.4 Definition of Evaluation Technique ... 108
B.11.5 Set-up of Data Source and Data Collection ... 109
B.11.6 Data Analysis / Identification of Success / Impact Analysis ... 110
B.12 DNS ... 111
B.12.1 Requirements and Strategic Evaluation Objectives ... 111
B.12.2 Definition of Evaluation Areas, Impacts and Selection of Indicators ... 111
B.12.3 Definition of Success and Targets ... 112
B.12.4 Definition of Evaluation Technique ... 113
B.12.5 Set-up of Data Source and Data Collection ... 114
B.12.6 Data Analysis / Identification of Success / Impact Analysis ... 114
B.13 RCB ... 115
B.13.1 Requirements and Strategic Evaluation Objectives ... 115
B.13.2 Definition of Evaluation Areas, Impacts and Selection of Indicators ... 115
B.13.3 Definition of Success and Targets ... 117
B.13.4 Definition of Evaluation Technique ... 117
B.14 AAA ... 118
B.14.1 Requirements and Strategic Evaluation Objectives ... 119
B.14.2 Definition of Evaluation Areas, Impacts and Selection of Indicators ... 119
B.14.3 Definition of Success and Targets ... 120
B.14.4 Definition of Evaluation Technique ... 121
B.15 SLA ... 122
B.15.1 Requirements and Strategic Evaluation Objectives ... 122
B.15.2 Definition of Evaluation Areas, Impacts and Selection of Indicators ... 122
B.15.3 Definition of Success and Targets ... 123
B.15.4 Definition of Evaluation Technique ... 123
B.16 IMS PLUS RAN/EPC ... 124
B.16.1 Requirements and Strategic Evaluation Objectives ... 124
B.16.2 Definition of Evaluation Areas, Impacts and Selection of Indicators ... 125

B.16.3 Definition of Success and Targets *126*
B.16.4 Definition of Evaluation Technique *126*
B.17 MCN-ENABLED DIGITAL SIGNAGE PLUS RAN/EPC 127
B.17.1 Requirements and Strategic Evaluation Objectives *128*
B.17.2 Definition of Evaluation Areas, Impacts and Selection of Indicators *128*
B.17.3 Definition of Success and Targets *128*
B.17.4 Definition of Evaluation Technique *129*

REFERENCES **130**

Table of figures

Figure 1 – Current DSS implementation status 18
Figure 2 – DSS deployment and provisioning 20
Figure 3 – DSS and CMS configuration 21
Figure 4 – DSS scaling using MaaS 21
Figure 5 – Current implementation status of IMS 24
Figure 6 – IMS Deployment and Provisioning 25
Figure 7 – IMS call 26
Figure 8 – IMS using MaaS 26
Figure 9 – Heat integration across clouds 31
Figure 10 – Distributed-cloud MCN service deployment 32
Figure 11 – CloudSigma web app advanced setting screenshot 34
Figure 12 – (Private) clouds, offering real-time guarantees, positioned close to RHHs 36
Figure 13 – MCN Services distributed over multiple clouds 37
Figure 14 – Month 24 Mobile Cloud Networking Integrated Prototype 41
Figure 15 – MCN SDK 44
Figure 16 – Evaluation process 49

Table of tables

Table 1 – Relationship Service-Concept for DSSaaS 15
Table 2 – Relationship Service-Concept for IMS 22
Table 3 – DSS M27 Roadmap 27
Table 4 – DSS detailed roadmap M24-M27 27
Table 5 – IMS M27 Roadmap 27
Table 6 – IMS detailed roadmap 28
Table 7 – Defined interfaces and their status 45
Table 8 – Evaluation Activities (according to adopted methodology) Gantt Chart 52
Table 9 – Fraunhofer/TUB testbed characterization and capacity 54
Table 10 – PTIN testbed characterization and capacity 55
Table 11 – FT/Orange testbed characterization and capacity 57
Table 12 – CloudSigma testbed characterization and capacity 57
Table 13 – INTEL characterization and capacity 59
Table 14 – TI testbed characterization and capacity 61
Table 15 – UTWENTE testbed characterization and capacity 62
Table 16 – UBERN testbed characterization and capacity 64
Table 17 – ZHAW testbed characterization and capacity 65
Table 18 – ONE's testbed characterization and capacity 67
Table 19 – DSS testbed characterization and capacity 68
Table 20 – Requirements and Evaluation Objectives 70
Table 21 – Evaluation Area, Impact and Indicators definition 70
Table 22 – Impacts Target 71
Table 23 – Indicators assessment 71
Table 24 – Requirements and Evaluation Objectives 72
Table 25 – Evaluation Area, Impact and Indicators definition 73
Table 26 – Impacts Target 74
Table 27 – Indicators assessment 75
Table 28 – Requirements and Evaluation Objectives 76
Table 29 – Evaluation Area, Impact and Indicators definition 77
Table 30 – Impacts Target 78
Table 31 – Indicators assessment 79
Table 32 – Requirements and Evaluation Objectives 80
Table 33 – Evaluation Area, Impact and Indicators definition 81
Table 34 – Impacts Target 82
Table 35 – Indicators assessment 83

Table 36 – Requirements and Evaluation Objectives 84
Table 37 – Evaluation Area, Impact and Indicators definition 85
Table 38 – Impacts Target 87
Table 39 – Indicators assessment 88
Table 40 – Requirements and Evaluation Objectives 90
Table 41 – Evaluation Area, Impact and Indicators definition 90
Table 42 – Impacts Target 91
Table 43 – Indicators assessment 92
Table 44 – Requirements and Evaluation Objectives 93
Table 45 – Evaluation Area, Impact and Indicators definition 94
Table 46 – Impacts Target 95
Table 47 – Indicators assessment 96
Table 48 – Scenarios 98
Table 49 – Indicators results 99
Table 50 – Requirements and Evaluation Objectives 100
Table 51 – Evaluation Area, Impact and Indicators definition 100
Table 52 – Impacts Target 100
Table 53 – Indicators assessment 101
Table 54 – Requirements and Evaluation Objectives 101
Table 55 – Evaluation Area, Impact and Indicators definition 102
Table 56 – Impacts Target 103
Table 57 – Indicators assessment 104
Table 58 – Requirements and Evaluation Objectives 106
Table 59 – Evaluation Area, Impact and Indicators definition 106
Table 60 – Impacts Target 106
Table 61 – Indicators assessment 106
Table 62 – Requirements and Evaluation Objectives 107
Table 63 – Evaluation Area, Impact and Indicators definition 107
Table 64 – Impacts Target 107
Table 65 – Indicators assessment 108
Table 66 – Requirements and Evaluation Objectives 111
Table 67 – Evaluation Area, Impact and Indicators definition 111
Table 68 – Impacts Target 112
Table 69 – Indicators assessment 113
Table 70 – Indicators Values 114
Table 71 – Requirements and Evaluation Objectives 115
Table 72 – Evaluation Area, Impact and Indicators definition 115
Table 73 – Impacts Target 117
Table 74 – Indicators assessment 117
Table 75 – Requirements and Evaluation Objectives 119
Table 76 – Evaluation Area, Impact and Indicators definition 119
Table 77 – Impacts Target 120
Table 78 – Indicators assessment 121
Table 79 – Requirements and Evaluation Objectives 122
Table 80 – Evaluation Area, Impact and Indicators definition 122
Table 81 – Impacts Target 123
Table 82 – Indicators assessment 123
Table 83 – Requirements and Evaluation Objectives 124
Table 84 – Evaluation Area, Impact and Indicators definition 125
Table 85 – Impacts Target 126
Table 86 – Indicators assessment 126
Table 87 – Requirements and Evaluation Objectives 128
Table 88 – Evaluation Area, Impact and Indicators definition 128
Table 89 – Impacts Target 128
Table 90 – Indicators assessment 129

Acronyms

AAA – Authentication, Authorization and Accounting

API – Application Programming Interface

BBU – Base Band Units

CC – Cloud Controller

C-RAN – Cloud Radio Access Network

CDN – Content Delivery Network

CLI – Command Line Interface

COTS – Commercial Of The Shelf

CPU – Central Processing Unit

DB – Database

DHCP – Dynamic Host Configuration Protocol

DMM – Distributed Mobility Management

DNS – Domain Name System

DSN – Digital Signage Network

DSS – Digital Signage System

E2E – End-to-end

EPC – Evolved Packet Core

EU – End-User

EEU – Enterprise End-User

HDD – Hard Disk Drive

HSS – Home Subscriber Server

HTTPS –Hypertext Transfer Protocol Secure

ICN – Information Centric Network

IMS – IP Multimedia Subsystem

IP – Internet Protocol

IT – Information Technology

ITG – Infrastructure Template Graph

iSCSI – Internet Small Computer System Interface

LB – Load Balancer

LTE – Long Term Evolution

KVM – Kernel-based Virtual Machine

MCN – MobileCloud Networking

MCR – Main Content Repository

MPLS – Multi Protocol Label Switching

NCP – Network Connectivity Provider

NFS – Network File System

NUMA – Non-Uniform Memory Access

OAI – OpenAirInterface

OCCI – Open Cloud Computing Interface

OTT – Over-The-Top

PoC – Proof of Concept

PoP – Point of Presence

PXE – Pre-Execution Environment

QCOW – QEMU Copy On Write

QEMU – Quick Emulator

QoS – Quality of Service

RAM – Random Access Memory

RAN – Radio Access Network

RCB – Rating Charging Billing

REST – Representational State Transfer

RRH – Remote Radio Head

SAN – Storage Area Network

SFTP – SSH File Transport Protocol

SCP – Secure Copy Protocol

SDK – Service Development Kit

SIMD – Single Instruction Multiple Data

SIP – Session Initiation Protocol

SIC – Service Instance Component

SLA – Service-Level Agreement

SO – Service Orchestrator

SM – Service Manager

STG – Service Template Graph

SSD – Solid State Drive

SSH – Secure Shell

VM – Virtual machine

VLAN – Virtual Local Area Network

WP – Work Package

XaaS – X as a Service

1 Introduction

The Integration and Evaluation activities steered by Work Package 6 are presented in this deliverable. This document aims laying the grounds for the upcoming integration, experimentation and evaluation activities. Moreover, this document complements D6.1 (D6.1, 2014) by providing detailed information about the testbed facilities and resources that will be made available for the MCN project.

1.1 Motivation, Objectives and Scope

Following the milestone for the first integration of the MCN components in M21 (DoW, 2012), which consisted in the cloudification of the defined services and integration with the MCN framework, this deliverable presents the current status of the overall integration status.

Additionally, D6.2 also reports on the developments achieved regarding the Proof-of-Concept (PoC) prototypes – these being the Digital Signage System (DSS) and Internet Protocol Multimedia Subsystem (IMS), presented in D5.1 (D5.1, 2013) and D5.2 (D5.2, 2014) – internally aligned to have a first version ready by M24. This date corresponds to the submission date of this deliverable and represents the latest outcomes from the performed integration towards the next integration report in M26.

Taking into account the forthcoming milestone for the final integrated version of the overall MCN framework and services, in M30, this deliverable also lays the grounds for the remainder of the integration activities as well as the supporting infrastructure and testbed resources to perform the required system evaluation and testing, due in M36.

1.2 Structure of the Document

The advances regarding the Digital Signage System (DSS) and IP Multimedia Subsystem (IMS) proof-of-concept prototypes are presented in Section 2, followed by the specification of the MCN testbed and available resources, in Section 3. In Section 4 a description of the integration activities that took place in WP6 is provided. The established experimentation and evaluation plans, following the directives from D6.1 are presented in Section 5, presenting the first results that were focused on the definition of Key Performance Indicators (KPIs) as well as on a preliminary evaluation of selected services. Finally, some concluding thoughts are provided in Section 6.

2 Proof-of-Concept and Prototypes

This section presents and explains the Proof-of-Concept (PoC) demonstrator and prototypes. It also intends to define the main concepts to be covered by the two PoCs expected by the MCN Project. These PoCs involve the majority of the MCN services, already developed by M24 (1st version of the PoCs), including all the developed services by M27, in the last version of the PoCs.

2.1 Introduction

The scope of the proof of concept goes beyond the mere demonstration of applications (IMS, DSS), representing also the entire MCN functionality and its benefits. In the first iteration of the PoC these benefits will be shown partially, while in the last PoC version all of these benefits will be showed and verified.

The main objective of the PoC is to provide a set of satisfactory and reliable results that demonstrate the capabilities and enhancements of the MCN approach over other service-provider approaches. According to this, some aspects such as On-Demand, Flexibility, Scalability or End-to-End (E2E) Service Composition, will be evaluated and analysed after the PoCs.

To achieve this main goal, two technical PoC have been defined and detailed. In both of them, what is intended is to have the entire MCN platform working not only together, but also in a consistent and reliable way, providing to the user an estimable enhancement of the overall Quality of Service QoS and other improvements over the traditional approaches offered by Telcos and Over-the-Top (OTT) services.

The remainder of the section presents the general status for the defined PoCs, followed by the roadmap and identified open issues for both IMS and DSS PoCs.

2.2 DSS PoC Status

At the moment of writing this deliverable, the general status of the demonstrator corresponds to a fundamental Cloud infrastructure supporting a minimal set of services to provide to DSS its main functionality and features, which are consist in the upload and download of contents, creation of playlists (grids) using this content and displaying it in players that are organized in groups. Some services have been integrated with the service to reach a consistent scenario that shows the benefits and advantages of the MCN Cloud.

Table 1 shows the relationship between services and cloud-computing concepts that will be proved on the first DSS PoC.

Table 1 – Relationship Service-Concept for DSSaaS

SERVICE	CONCEPT
Cloud Controller/IaaS	On-Demand
Service Orchestrator	Scalability/Elasticity
LBaaS	Scalability/Elasticity
DBaaS	Shared storage
MaaS	Measured Service/QoS
CDNaaS	Improved DL performance/QoS/Efficiency
RAN+EPCaaS	Remote Control/E2E Service Composition

DNSaaS	Will not prove any MCN concept by itself

2.2.1 DSS Requirements per Service for M24

2.2.1.1 Satisfied Requirements for M24 DSS PoC

At the moment of writing this deliverable, the following services are already available for M24 DSS PoC:

- **MCN Service Development Kit (SDK)**
  - Capabilities to deploy, update, dispose and get info of DSS Infrastructure Template Graphs (ITGs) from the Cloud Controller.
- **Content Delivery Network (CDN)**
  - Capability to upload and download contents through Representational State Transfer (REST) Application Programming Interface (API) calls. Origin server and Point of Presence (PoP) instances are available.
- **Load Balancer (LB)**
  - Capabilities provided by LBaaS (Load Balancer/HealthMonitor, etc.), supported by the OpenStack Neutron service.
  - LB will be managed by updating stacks by a newly generated ITG.
- **Evolved Packet Core (EPC) + Radio Access Network (RAN)**
  - HTTP Connectivity between the player and the PoP CDN Instances so that DSS players will be able to download contents.
  - Connectivity between the players and the CMS to download the playlist (grids) and others.
- **Monitoring (M)**
  - Capability to add metrics of the DSS Service.
  - Capability to get monitored data.
  - Capability to define triggers to ease management of the monitored data
- **Domain Name System (DNS)**
  - Capability to resolve domain names to make easier the communication management between different services and service components.
- **Database (DB)**
  - Capabilities provided by DBaaS, supported by Trove, which is will be available in OpenStack testbeds.
  - DB will be managed by updating stacks by a newly generated ITG.

2.2.1.2 Non-Integrated Requirements (Post-M24)

Support services Authentication, Authorization and Accounting (AAA), SLA and Rating Charging and Billing (RCB) are not a part of M24 PoC. According to the DSS integration roadmap, these will be available on M27 for the final PoC. MCN SDK will also provide some new functionality like heat template management for the final PoC**.** Also the integration of some already available services for M24 PoC will be enhanced for M27.

According to the current architectural decisions, the AAA, SLA, Mobility and Bandwidth Availability Prediction (MOB) and RCB services will provide the following functionalities:

For AAAaaS:

- **One point Registration:** Registering for one MCN service and being able to authenticate in all other MCN services using same credentials;
- **Single sign on:** Authenticating in one MCN service and being able to use the same token for all other MCN services;
- **Application level authentication:** Provides application level authentication for services**;**
- **Telco level authentication:** Provides Telco reinforcement also for Information Technology (IT) service authentication, which means that some Telco related information will be used also for application level authentication;

For RCBaaS:

- **Collect and Process RCB data:** RCB data will be pushed to MaaS by DSS, using provided interface by MaaS. RCB collects this data from MaaS and processes it;
- **Bill Generation:** Bill will be available to be requested using an interface provided by RCB;

For SLAaaS:

- **Creation and management of SLAs:** DSS EEU will create SLAs according to its needs for establish policies for monitoring services.
- **SLA templates management:** DSS EEU will be able to create templates gathering them, and offering the possibility to modify or delete them.
- **SLA subscription management:** DSS EEU will be notified of any change in the SLAs agreement state.

For MOBaaS:

- **Collect MOB data:** DSS will collect mobility data (such as location tracking, movement history, traffic generated or bandwidth, for example) for later mobility management.
- **Prediction of resources to be used:** MOB will predict the resources used by DSS, such us bandwidth or location tracking, establishing an estimation of the location and the bandwidth that will be available at the predicted location in a future moment of time.

2.2.2 Implementation

Figure 1 depicts the current status of the DSS implementation, presenting a simplified architecture that considers the main relationships between DSS and other MCN components.

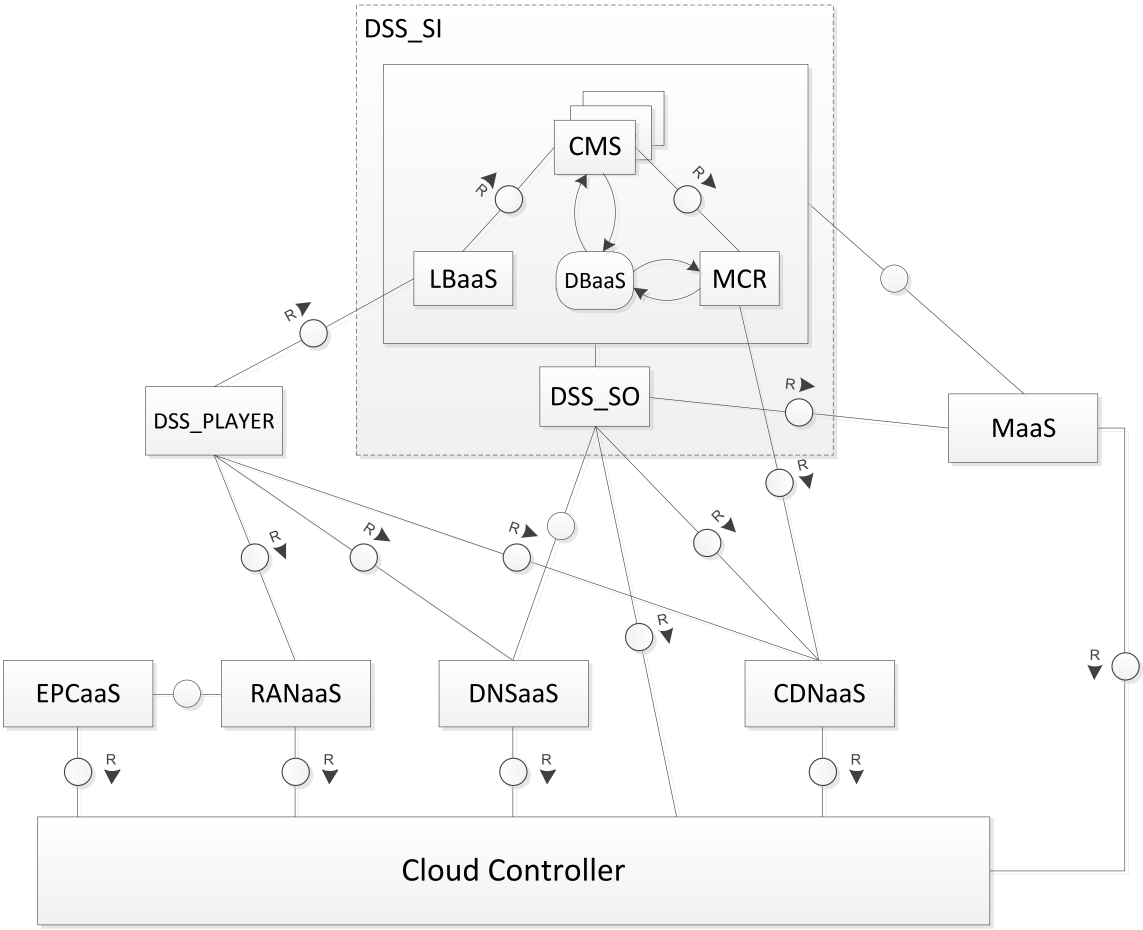

Figure 1 – Current DSS implementation status

At the moment of writing this deliverable, the implementation and integration status of each component from the DSS service is as follows:

DSS Service Orchestrator (SO)

Currently, the **DSS SO** is integrated with **CDNaaS**, **MCN SDK**, **LBaaS**, **MaaS** and **DNSaaS**. The detailed information about their integration is provided below:

- **DNSaaS:** DSS SO is getting required information (IP addresses and domain names), transforms it into DNS A-records and pushes them into DNS instance.
- **MaaS:** DSS SO pushes DSS metrics into monitoring instance and provides it with monitoring data. Later DSS SO will fetch these data and uses them for scaling decisions.
- **CDNaaS:** The DSS SO has been modified to get CDN endpoint from the Cloud Controller, perform initial DSS related CDN configurations: registration (account creation) and PoP instances assignation. Also, SO configuration component pushes required CDN data into the Service Instance Components (SICs) at provisioning time.
- **MCN SDK:** Integration of MCN SDK currently makes the DSS service able to perform main actions toward the service, which consist in deployment, update and disposal of service.
- **LBaaS:** Load Balancing is provided by Neutron in the OpenStack Platform, and is managed by the DSS Service Orchestrator.

DSS Content Management System (CMS) and Main Content Repository (MCR)

At the moment, **CMS** is integrated with **LBaaS** and **MCR** is integrated with **CDNaaS** and both of them are integrated with **DBaaS, MaaS**, **and DNSaaS**. The detailed information about their integration is provided below:

- **DNSaaS:** An agent has been developed for DSS instance components, which is responsible for configuring different value. These values and configuration make DSS SICs able to connect to DNS instance and resolve domain names.
- **MaaS:** An agent (zabbix-agent) is installed on CMS and MCR SIC and is responsible for sending monitoring data to monitoring instance in defined intervals for each metric. Available metrics are Central Processing Unit (CPU) utilization for both CMS and MCR and available free storage for MCR. For the final PoC more metrics will be added, including monitoring of the DSS response times.
- **CDNaaS:** A REST interface has been designed and developed, between the MCR DSS Component and the CDN Service. This API provides DSS service with all the functionalities related to contents. DSS MCR is currently using this API to push contents into CDN instance.
- **LBaaS:** The CMS application on all CMS ICs is available through load balancer public IP address, which balances the load between all them.
- **DBaaS:** DSS CMS and MCR applications have been modified to share the same database instance, which is provided by DBaaS.

DSS Player

At the moment, this component is integrated with **EPC+RANaaS**, **CDNaaS** and **DNSaaS**. The detailed information about their integration is provided below:

- **DNSaaS:** DSS CMS application and Player have been modified and currently, after a first configuration, players are able to connect to DNS instance and resolve domain names.
- **CDNaaS:** Using the REST interface of the CDN service, DSS players are able to download the data from CDN POPs by sending a download request, which is made by a plugin attached to DSS players.
- **EPC+RANaaS:** Thanks to mobility manager module (provided by EPCaaS), which is currently available, installed and configured on DSS players, the necessary connectivity is available to access CMS applications and CDN PoP instances**.**

2.2.3 Demo Scenario for DSS (M24)

The demo scenario proposed for the M24 PoC will cover the MCN concepts and benefits detailed in the previous section (Section 2.2.1). In this context, the DSS PoC demonstration should realize the following tasks:

1. DSS on-demand deployment (IaaS/CC integration, see Figure 2)
2. DSS Provisioning using the ITG (see Figure 2)
3. DSS Usage (see Figure 3):
    a. User creation

b. Player creation
c. Group of players creation (DBaaS for storing elements):
d. Content upload. Due to CDN integration contents will be cached and available in nearby players, significantly improving the overall performance of DSS players.
e. Playlist (grid) of contents creation
f. Content download from the CDN PoP instance to the player. This player will be connected to RANaaS/EPCaaS (Remote Control/Connectivity)
g. Content displaying (Flexibility)

4. Increase system load, triggering scalability using MaaS trigger information (Measured Service, see Figure 4).
   a. The service orchestrator decision component will be able to automatically scale out/in or scale up/down the service, depending on the needs of the system in each moment.
   b. As the deployment and provisioning of the service itself consumes a high percentage of CPU utilization, a mechanism that ensures the provisioning has finished is temporarily disabling SO decision module to prevent unnecessary scale decisions.
   c. After each horizontal scaling, the LBaaS will rebalance the load between remaining CMS SICs. (Scalability/Elasticity).
5. DSS disposal

This scenario, and sequence of the most relevant steps, is depicted in Figure 2, Figure 3 and Figure 4.

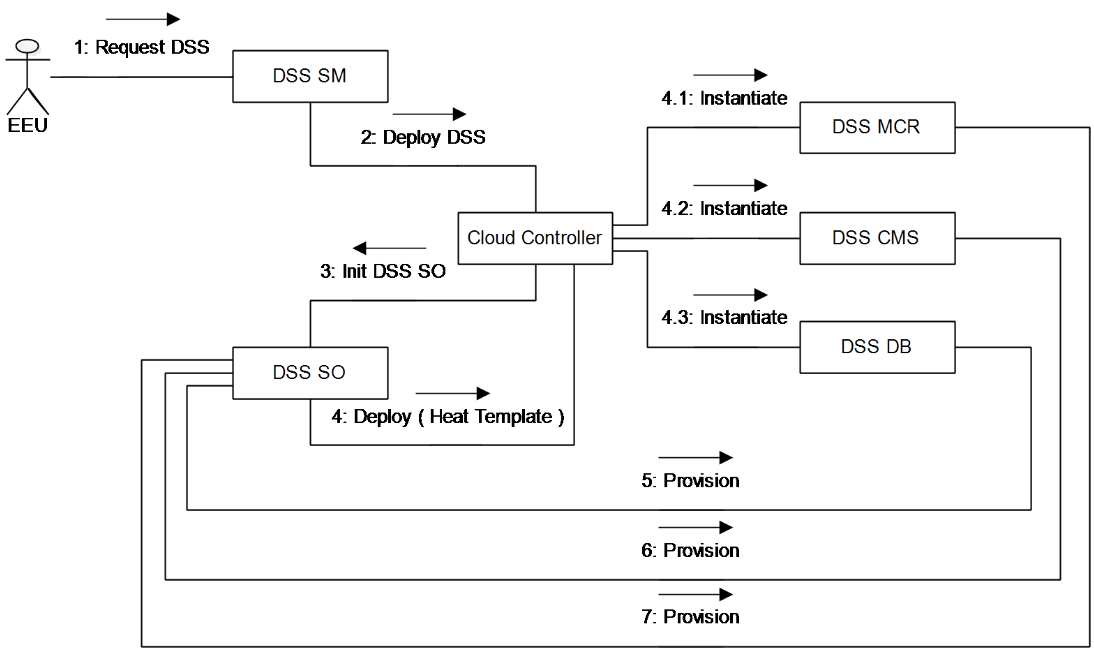

Figure 2 – DSS deployment and provisioning

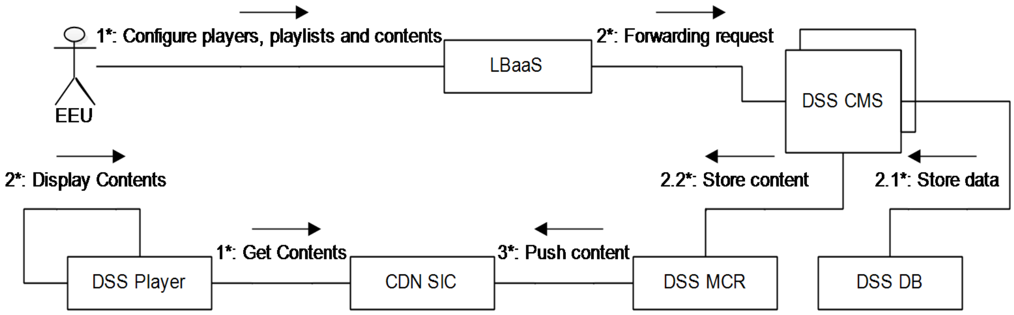

Figure 3 – DSS and CMS configuration

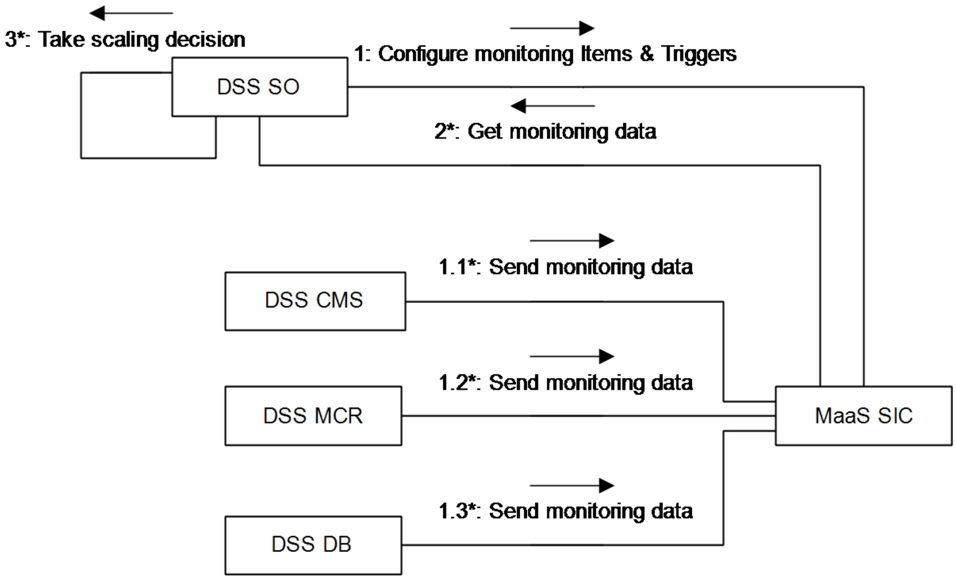

Figure 4 – DSS scaling using MaaS

Use case:

Initially, the EEU, on-demand, will request a DSS, to provide the End-User (EU) a Digital Signage Network as a Service for displaying HTML5 contents in different players. The provisioning of the instances is done using user-scripts inside the Heat template of the service. There are some initial configuration-values such as database name, super users' data, API URLs, among others, which these user scripts inject into instances. Also, an agent is pushed into each instance, which is responsible for listening to the service orchestrator requests and sending back corresponding responses. Endpoint of the requested DSS service will be provided to the EEU. Once the DSS instance is deployed and provisioned, the EU will have access to the DSS Content Management System (CMS) to register, and once registration is done, add the players' information, create a group for them, upload some contents to the Main Content Repository (MCR), create a grid (playlist) and assign uploaded contents to it. All these tasks will be done through a web UI. Then, the players will be started. Players should get the connectivity through RANaaS/EPCaaS. Having the connectivity, they will be able to download the playlist according to the configuration performed in CMS. The request for the contents will be done through a CDN PoP instance. When all the contents of the chosen playlist have been downloaded, they will be displayed on corresponding players.

For exemplifying scaling in and out characteristics, the load from the CMS instance will be manually increased to demonstrate how MaaS and scalability (LBaaS/Service Orchestrator) work together.

The last step in the process presented by this use case will be to trigger and validate the DSS service disposal.

2.3 IMS PoC Status

2.3.1 General Status

The general status for the IMS demonstrator corresponds to a basic functionality of an IMS service to provide a simple phone and multimedia services through IP. On this context, the following table shows the concepts that will be covered by the first IMS PoC. They are basically the same as DSS PoC, adding Rating, Charging and Billing to cover the Pay-as-you-Go concept.

Table 2 – Relationship Service-Concept for IMS

SERVICE	CONCEPT
Cloud Controller/IaaS	On-Demand
Service Orchestrator	Scalability/Elasticity
DBaaS	Shared storage
MaaS	Measured Service
RAN+EPCaaS	E2E Service Composition
RCBaaS	Pay-as-you-Go[1]
DNSaaS	Will not prove any MCN concept by itself

2.3.2 IMS Requirements per Service for M24

2.3.2.1 Satisfied Requirements for M24 IMS PoC

At the moment of writing this deliverable, the following services are already available for M24 IMS PoC:

- **MCN SDK**
  - Capability to deploy, update, dispose and get info of IMS stacks from the Cloud Controller
- **DNS**
  - Capability to provision domains/records and then lookup queries among service instance components
- **EPC+RAN**
  - It guarantees the connectivity of the EU Session Initiation Protocol (SIP) devices to the IMS service.
- **Monitoring**
  - Capability to add metrics of the IMS Service

[1] **NOTE:** Pay-as-you-Go is not provided directly by RCBaaS, but it may be demonstrated thanks to RCBaaS.

  - Capability to get monitored data
- **RCB**
  - IMS sends billing information through an HTTP interface to RCB service
  - EPC interface has been integrated

2.3.2.2 Non-Integrated Requirements (Post-M24)

The following requirements are planned for M27 of the IMS PoC.

- **DB**
  - DBaaS: Trove, which is available in OpenStack, will be used.
  - DB will be managed by updating stacks by a newly generated ITG.
- **AAA**
  - IMS is already integrated into the AAA architecture defined by 3GPP, therefore this will be considered differently than it was for DSS.
- **SLAaaS**
  - See section 2.2.1.2
- **MOB**
  - See section 2.2.1.2

2.3.3 Implementation

IMS current implementation status is depicted by Figure 5, which illustrates the existing links between the following IMS and MCN components:

- **IMS-EPC and RAN integration:** Currently, IMS+EPC/RAN are totally integrated, and its demonstrator was showed on the End-To-End MCN Demo on month 20 of the project.
- **IMS-RCB integration:** RCB core logic has been implemented. Data is sent from IMS to the RCB Message Front End through the RabbitMQ messaging platform, which uses an open HTTP-based API. The RabbitMQ client processes data and then sends it to the external Billing system (a JBilling plug-in has been developed) through FTP interface.
- **IMS-MaaS integration:** monitoring agent will be installed in each service instance. They will collect the data and send it to MaaS. Later the data will be requested by SO to decide and perform consequent operations.
- **IMS-DB integration:** IMS service instance components have been modified to share same database instance, which is provided by DBaaS.
- **IMS-DNS integration:** IMS use DNSaaS for registering names of the IMS instances for further management.

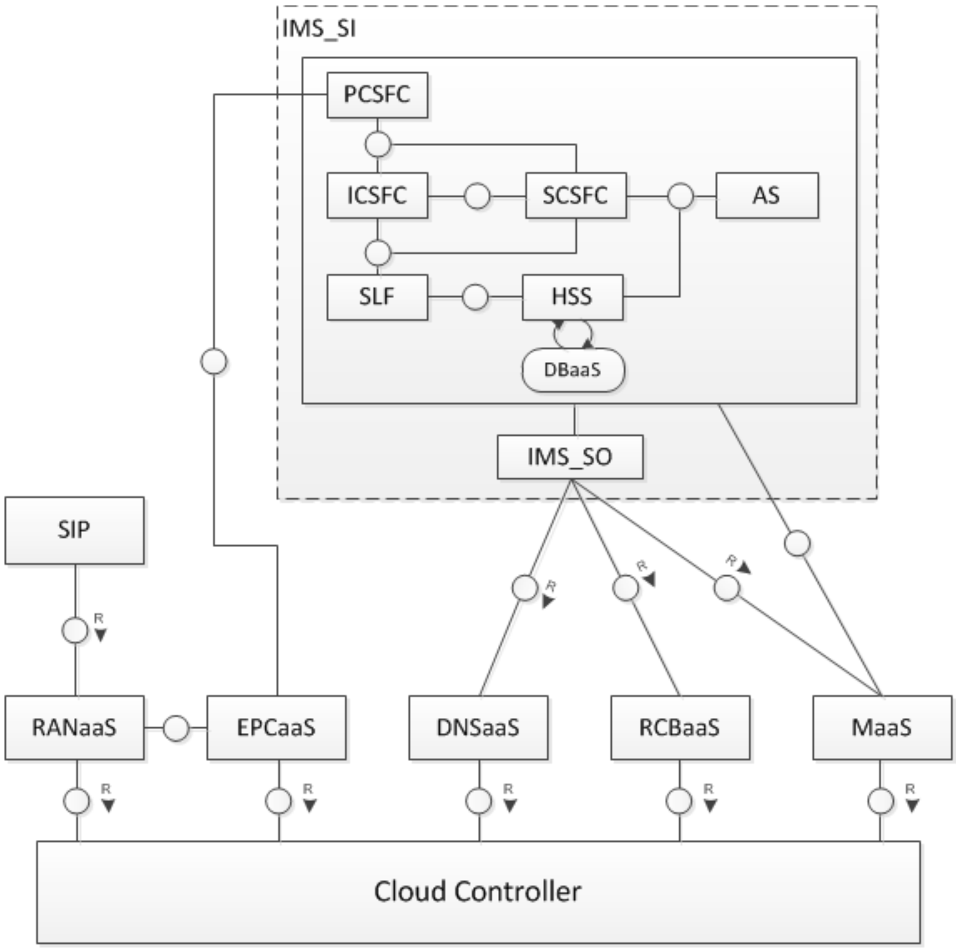

Figure 5 – Current implementation status of IMS

2.3.4 Demo Scenario for IMS (M24)

The demo scenario proposed for the M24 IMS PoC will cover the MCN concepts and benefits detailed in the previous section and in D6.1, illustrated by Figure 6.

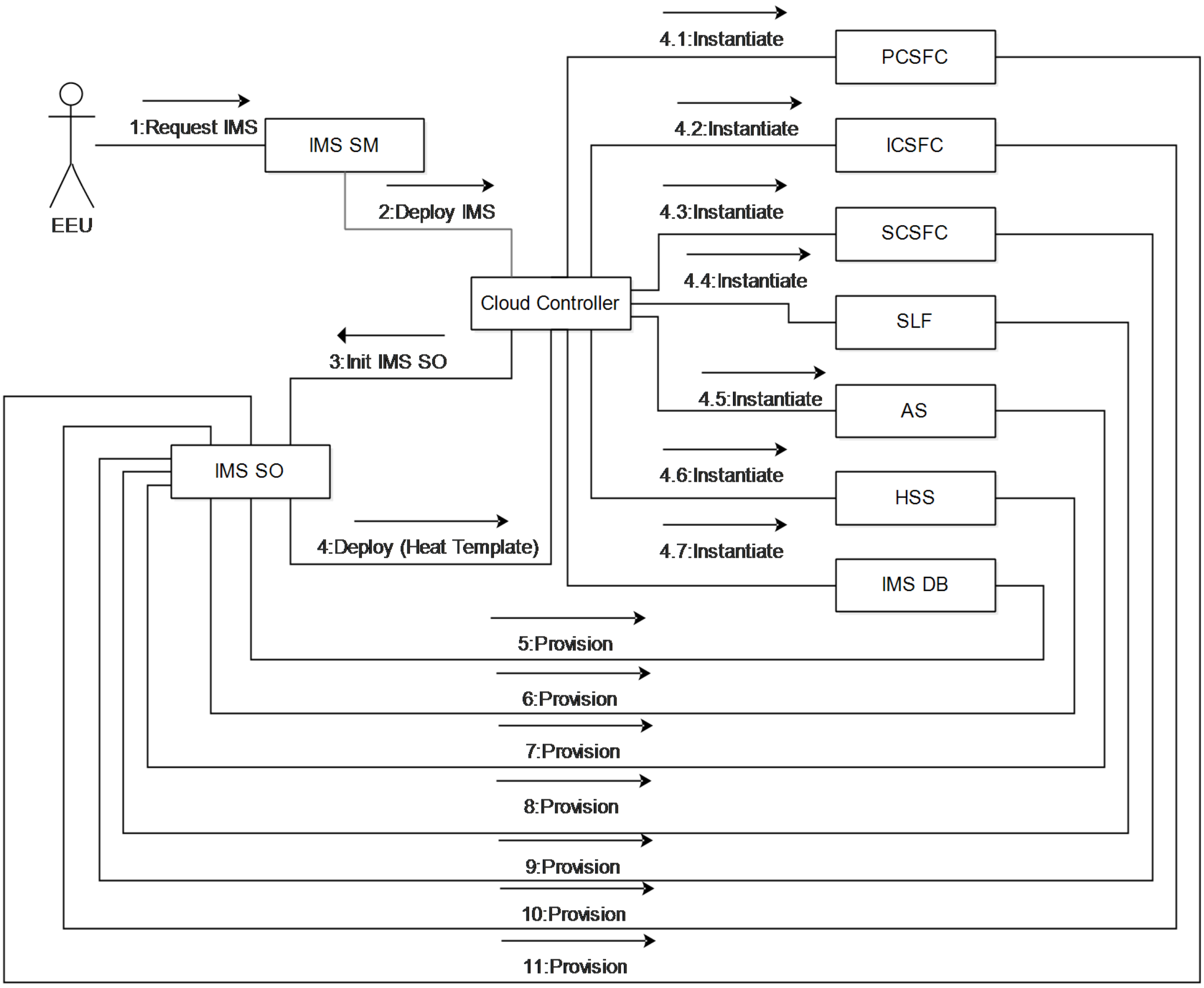

Figure 6 – IMS Deployment and Provisioning

Bearing this in mind, the IMS PoC demonstration should realize the following tasks:

1. IMS on-demand deployment (IaaS/CC integration, see Figure 6).
2. IMS provisioning (see Figure 6).
3. IMS Usage (Figure 7 and Figure 8):
    a. Register two SIP devices connected through RANaaS/EPC (Remote Control/ Connectivity).
    b. One of the devices will start the communication.
    c. Connection will be established and the end users will talk for a few seconds.
    d. The call will finish.
4. IMS disposal
5. RCBaaS will charge the service according different metrics and fees (e.g.: session_start, total_traffic_in_Mb, customer_name, service_type)

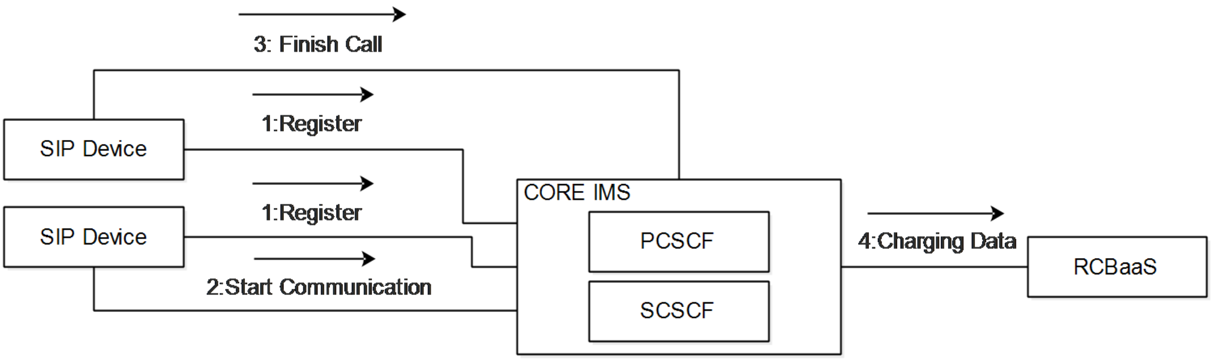

Figure 7 – IMS call

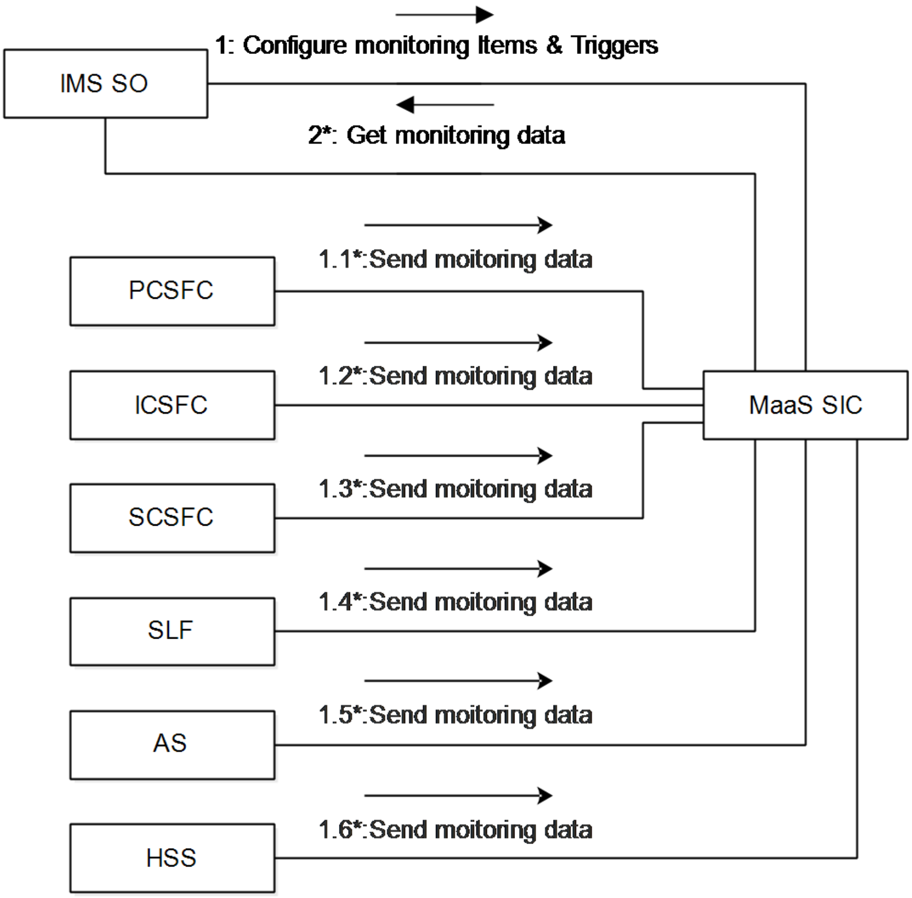

Figure 8 – IMS using MaaS

Use case:

The Enterprise End-User (EEU) will deploy an IMS instance on-demand using the Cloud Controller MCN framework and through its SO/Service Manager (SM) and IMS specific Heat template. Provisioning will take place using configuration scripts. Once deployed, two SIP devices are registered and connected to the system (Remote Control/Connectivity). One of the two end users will start the communication sending SIP INVITE. Once established (End-to-End), the end users will talk for a few seconds.

During the process, RCBaaS will charge the service according different metrics and fees (Pay-as-you-Go through MaaS).

For showing scalability, the load of IMS instances will be manually increased to demonstrate how Service Orchestrator works.

The process concludes with the disposal of the IMS service, which will terminate all the used IMS instances.

2.4 Roadmap and Open Issues

2.4.1 DSS

The final milestone, which will be reached by M27, is a complete integration between the cloudified DSS and the support and MCN services, to realize the final PoC. This will demonstrate the complete functionality and benefits of the MCN platform. This means, according to D6.1 (D6.1, 2014), covering all the MCN concepts for an OTT application. Table 3 presents the planning for DSS towards M27.

Table 3 – DSS M27 Roadmap

SERVICE	CONCEPT
SLAaaS	Measured Service/QoS
MOBaaS	Follow-me-Cloud/Prediction
RCBaaS	Pay-as-you-Go[2]
AAAaaS	Authentication/SSO
ICNaaS	Follow-me-Cloud/QoS

According to this table, what will be intended for the months between M24 and M27 corresponds to the following figure:

Table 4 – DSS detailed roadmap M24-M27

TH	EXPECTED INTEGRATION
M25	RCBaaS
M26	ICNaaS, AAAaaS
M27	MOBaaS, SLAaaS

2.4.2 IMS

The expected roadmap for IMSaaS for M27 is the complete integration with the MCN platform, offering a complete IP multimedia subsystem services. According to Table 5, it is possible to observe the MCN concepts that will be proved in the last IMS PoC.

Table 5 – IMS M27 Roadmap

SERVICE	CONCEPT
SLAaaS	Measured Service/QoS
MOBaaS	Measured Service/Prediction

[2] **NOTE:** Pay-as-you-Go is not provided directly by RCBaaS, but it may be demonstrated thanks to RCBaaS.

Within this context, the planned roadmap is presented in Table 6.

Table 6 – IMS detailed roadmap

TH	EXPECTED INTEGRATION
M26	MOBaaS
M27	SLAaaS

3 Testbed Planning

This section relates to the deployment and maintenance of the testbeds that will be used for experimentation and evaluation purposes throughout the remaining project time and beyond. One of the objectives of Work Package 6 is to provide a process for matching the resources, made available via testbeds offered by MCN partners with the requirements of MCN services and their components. The requirements gathering process has been executed by T6.4 and is considered an on-going process as long as services continue to require testbeds.

Initially, all service owners decided to deploy their services on a single testbed to ensure a consistent working environment and to ease on-going integration work. As service provisioning and integration progressed on one testbed, it was decided to incrementally migrate services onto other testbeds, beginning with the non-OpenStack based CloudSigma testbed. Currently work focuses on service integration on a per-testbed basis, with the coming months delivering inter-testbed service integration, addressing the multi-cloud PoC for MCN services in the DoW.

3.1 Introduction

In this section, we will firstly look at the process of gathering and updating information on the individual testbed in terms of their characteristics and capacity. We will briefly describe how we created an online form to capture and update descriptions of all the testbeds, as well as document the process of creating a separate online form to help understand the testbed requirements particular to each MCN service, their interconnections and dependencies. We also document the process that we have proposed for the matching of service requirements to testbeds and how the resulting spreadsheets can provide a “quick look” comparison throughout the project lifecycle.

In section 3.2 we will describe the progress made in integrating multiple testbeds into one distributed-cloud environment and outline the development work done to facilitate this. Heat integration is outlined, as well as limitations with the current distributed-cloud solution. Considerations, framework limitations and platform benefits to be born in mind by service owners when making use of CloudSigma’s testbed are also discussed.

Section 3.3 looks at RANaaS (OpenAirInterface) and the progress we have made configuring resources on CloudSigma infrastructure, as well as outlining potential issues that are currently foreseen, while offering possible solutions. Section 3.4 is concerned with the requirements of EPCaaS and IMSaaS, and predicts two potential end-to-end scenarios. The same analysis is provided for DSSaaS, presented in Section 3.5. Maintenance of the updated testbed components is discussed in section 3.6, along with strategies for facilitating inter-cloud connectivity.

Upcoming work is discussed in Section 3.7, which describes the details involved in converting existing OpenStack targeting Heat templates to match CloudSigma resource types. Potential opportunities for benchmarking these new distributed-cloud setups are also discussed.

Additionally Appendix A lists and describes the current testbeds offered according to the original project proposal. Information is updated here, where necessary, according to the data provided via the online form to include deviations and changes to the testbeds, extensions and upgrades, either planned or undertaken to date.

3.2 Testbed Characteristics and Capacity

The individual testbed descriptions have been updated to provide detailed information regarding the technologies used, resources, and capacity of each testbed available to service owners. This was achieved by issuing each testbed provider with an online form designed to capture and store all the information submitted into a single spreadsheet for the benefit of T6.4 and is made available to all service developers. The gathered information regarding each testbed can be found in Appendix A.

A second online form was created in parallel and distributed among MCN service owners. The purpose of this form is to collect the testbed resource requirements for unit testing, integration testing and finally for the end-to-end evaluation of the working MCN prototype developed in WP3, WP4 and WP5, according to the set of demonstration scenarios defined in WP2. Like the form used to capture up-to-date testbed information, a spreadsheet is automatically populated. The information captured with this form is helpful for matching specific experiments with the appropriate testbed or a combination of testbeds, as both sets of data can be viewed in relation with each other.

3.2.1 Matching Requirements to Testbeds

Most partners offer OpenStack Havana on their testbed infrastructure, as such providing support for Heat among other modules. Using the information from the testbed form, it was concluded that the only testbed offering OpenStack Havana, Heat, OpenShift and Ceilometer (integrated with Zabbix) support, required for the implementation of the Service Manager / Service Orchestrator, is the ZHAW testbed. TI's testbed was also considered, but lacked Ceilometer API support. Bearing in mind the known infrastructure requirements of the Service Orchestrator, infrastructure providers closely but not fully matching these requirements will be requested to deploy the missing infrastructure components in order to provide an alternative testbed for the Service Orchestrator. CloudSigma has since developed Heat support and talks are ongoing with ZHAW to use experience gathered in deploying the service orchestrator on their testbed onto the CloudSigma testbed.

Bearing in mind the real-time performance requirements that IMSaaS has of EPCaaS, quantitative performance testing is expected to be conducted on these virtualised services, in order to gauge under what circumstances the testbeds no longer meet the services' QoS criteria. This will be done in consultation with Task 3.2, for which the suggested testing methodology and workflow has been documented in D3.2.

CloudSigma's testbed will offer a public cloud environment, onto which the MCN Services can be deployed and performance tested. This is important for different reasons, namely public cloud cannot typically offer real-time performance guarantees. During performance testing, it was established that some MCN services require this. Quantitative testing over prolonged periods of time will provide empirical data as to the performance variability of MCN Services when undertaking different workloads on public cloud infrastructure. Also, the feasibility and suitability of MCN service bursting onto public cloud infrastructure will be identified. Moreover, many MCN Services do not have real-time performance guarantee requirements. As such, Telco-operated clouds run close to Remote Radio Heads (RRHs) could focus on providing infrastructure for serving these low-latency services, with limited support for services not requiring real-time performance guarantees. Services without real-time performance guarantees could be hosted remotely in cheaper second tier data centres or provisioned on-demand in public cloud as and when local private cloud capacity to provide these services is exhausted.

3.2.2 Integrated Testbed Architecture

The diagram in Figure 9 illustrates how Heat is to be deployed across multiple clouds in order to facilitate the provisioning of resources across multiple locations. Once provisioned, the network between VMs in multiple locations will need to implement a pan-cloud secure network in order to obfuscate the fact that they are not all running within the same local network. This can be seen illustrated in the diagram implemented as an IPSec based VPN tunnel.

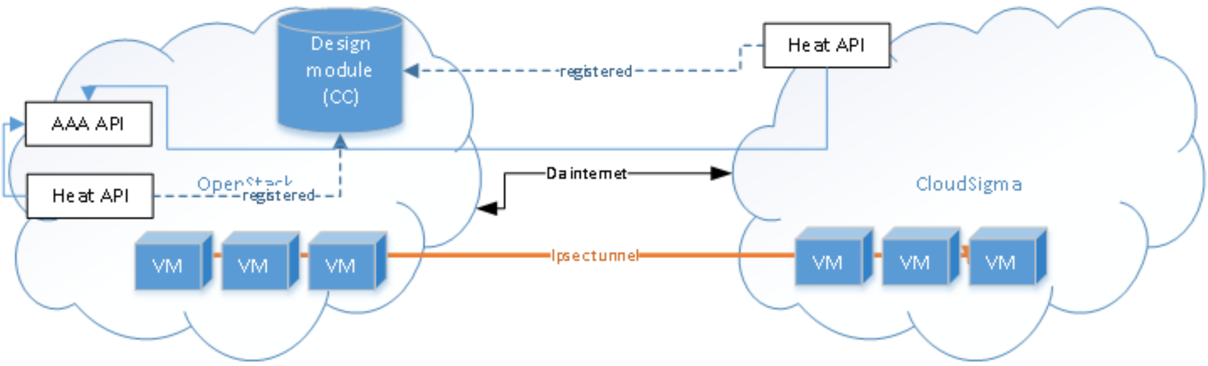

Figure 9 – Heat integration across clouds

3.2.3 Available Resources

One of MCN's primary goals is to facilitate the on-demand elastic provisioning of telecoms services on geographically aware cloud-agnostic infrastructure. As such, the underlying technologies used to implement the MCN architecture has focused on using open source solutions, as well as leading and emerging industry standards to facilitate this end goal. One such provisioning technology and templating model is used to implement MCN's Service Template Graph (STG) concept into a deployable stack is Amazon's AWS CloudFormation, which in MCN is implemented via Heat.

CloudSigma's cloud uses a proprietary stack and as such does not inherently support AWS templates nor does it offer Heat support out of the box. This offers MCN a great opportunity to explore and overcome obstacles in implementing the MCN stack on cloud technologies other than OpenStack and Amazon EC2.

In order to meet MCN's software technology choice implementation requirements, CloudSigma has developed a Heat plugin, which enables service owners to develop Heat templates that provision resources transparently on CloudSigma's cloud.

This implementation has been documented, describing in detail how to provision CloudSigma resources using Heat templates, facilitating service stack deployments on CloudSigma's infrastructure which when parameterised appropriately can provision CloudSigma resources with the same degree of granularity and flexibility as if requesting these resources programmatically using the CloudSigma API directly. Work is ongoing to develop distributed-cloud Heat templates, offering service deployment on both CloudSigma and, as a current example, the ZHAW testbed.

Figure 10 presents a high-level overview of a possible MCN distributed-cloud deployment pattern, facilitated by Heat, by the CloudSigma Heat-plugin and PyCloudSigma

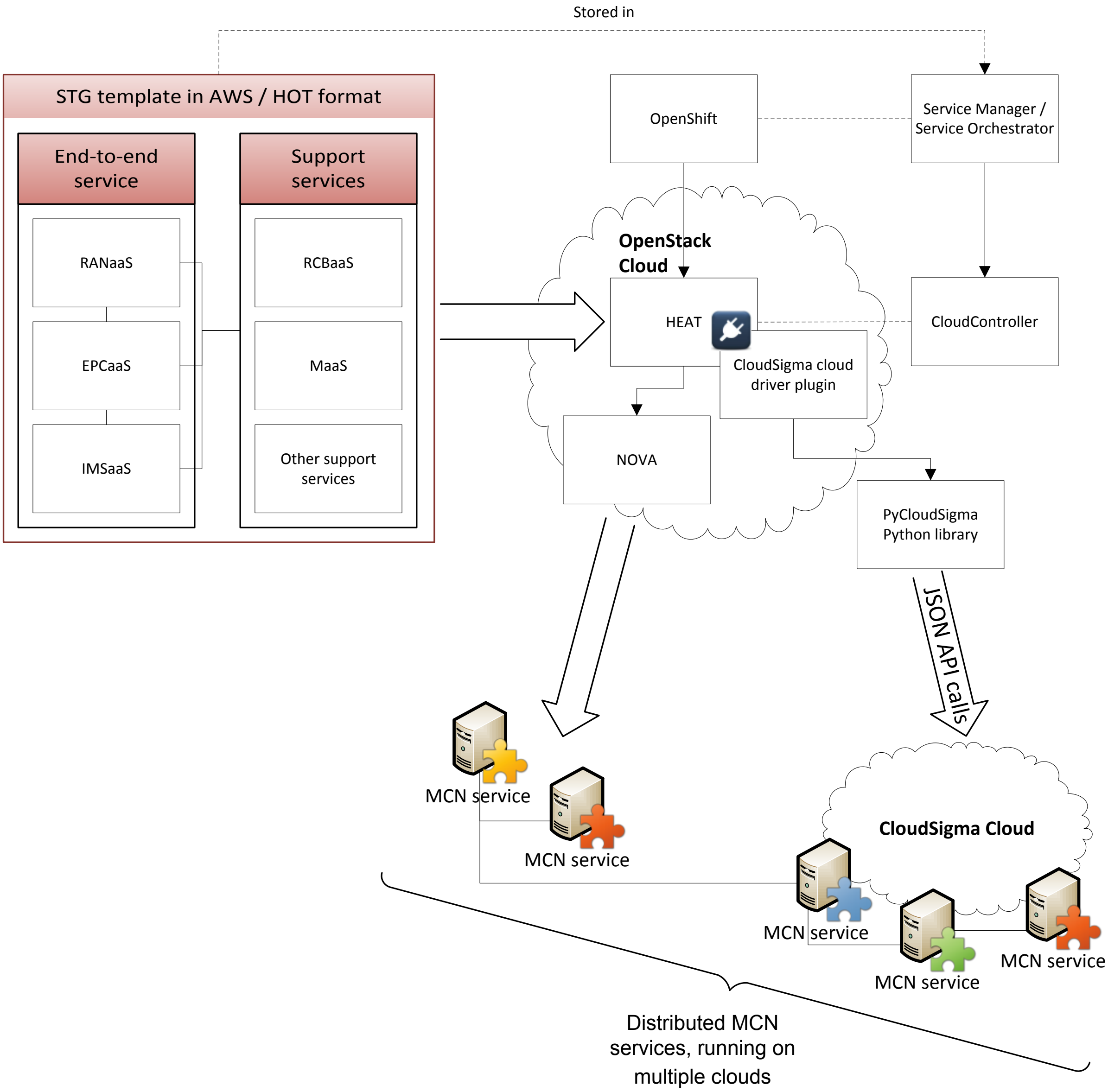

Figure 10 – Distributed-cloud MCN service deployment

3.2.4 Access to Cloudsigma's Testbed

In order to make the CloudSigma testbed available to all service owners, the CloudSigma Heat-plugin deployment has been registered with Keystone at ZHAW as a separate region. Regions are an ideal concept to be used to integrate a cloud with multiple sites, where we will schedule VMs to a particular site and desire a shared infrastructure. Regions also offer a robust way to share some infrastructure between OpenStack Compute installations or in this case, between OpenStack Compute and CloudSigma. This also allows for a high degree of failure tolerance, as long as STGs are available for both clouds with their respective resource definitions for the services being deployed.

The OpenStack dashboard (Horizon) currently uses only a single region, and as such the dashboard service is being run at CloudSigma to complement the one running at ZHAW.

Work is ongoing to migrate the existing service orchestrators to CloudSigma's cloud. This will enable local cloud deployment of the MCN stack orchestration elements.

3.3 RANaaS – OpenAirInterface (OAI)

At the time of writing this deliverable Task 3.5 (RANaaS) is still considered to be the most advanced in terms of having a clear understanding of the current testbed requirements. This section will document the process in which we assessed the requirements determined by Task 3.5 concerning the OpenAirInterface (OAI), a software-based Long Term Evolution base station. We document the basic and advanced configurations then attempt to solve any foreseen issues. Using the deployment of OpenAirInterface onto CloudSigma infrastructure as an example, the following sections offer a preview of the matching process that is currently being followed or will be followed by all other MCN services.

3.3.1 OpenAirInterface on CloudSigma Infrastructure

Task 3.5 requires a set of VMs with OpenAirInterface installed, for profiling of the base-station processing resources. Initially, OAI will be run on various VMs to profile the processing resources in the time needed for specific traffic loads and radio resources. The existing limitations in terms of the maximum distance between RRHs (base-station antennas) and the data-centres where the software-based base stations are instantiated is 10km. Data-centres must be available in locations belonging to operators that connect via fibre to neighbouring RRHs. As an example, a data-centre based in Zurich could never be used for a deployment of Cloud Radio Access Network (C-RAN) in Paris. Task 3.5 stipulated that the data-centres require to be distributed in a grid of 10-20km and that the compute resources must be well controlled. Eventually this will be the kind of scenario that would make sense to evaluate the necessity for private dedicated data-centres. However, for the purpose of profiling the base-station processing resources we begin by matching the OpenAirInterface to CloudSigma infrastructure. The following sections outline this process.

3.3.1.1 Basic Configuration

An account was created at CloudSigma for T3.5 for the installation of the OAI. Resources were provisioned using the subscription pricing method according to the testbed requirements. A buffer of 50EUR was also allocated to the account to be used for other services such as purchasing a Virtual Local Area Network (VLAN) or static IP address.

The online form is also intended to highlight other potential requirements. One requirement of OAI is for the ability to scale the compute capacity, most notably the CPU (3GHz to 10GHz) and Random Access Memory, RAM (2GB to 64GB). The actual compute capacity per VM is yet to be determined and should be done in such a way as to match units of 'resource blocks' of known real-world RAN processing capacity to units of VM CPU processing capacity. It has been made clear that the maximum should be able to be allocated at any time. Based on this requirement, CloudSigma has set an upper limit for each resource independently to match the upper limit required. 100GB of storage was also allocated. As documented in the CloudSigma testbed description, all resources are offered on a utility basis and no restriction on VM size is enforced. This unique functionality offers the ability to manually allocate different amounts of CPU and RAM independently of each other using the CloudSigma WebApp or a RESTful API and, once performance testing is performed, should provide visibility as to the optimum and modularised mapping of Cloud compute capacity to virtualised RAN services.

3.3.1.2 Advanced Configuration

CloudSigma makes available to each MCN service the same advanced features available to commercial customers. Advanced CPU options are made available, including exposing the full CPU instruction set for better application performance, Non-Uniform Memory Access (NUMA) visibility, hypervisor timer settings and virtual core size control. Information regarding these advanced features is presented as part of Deliverable D3.1 (D3.1 2013). Should it be possible to run a multithreaded instance of the OpenAirInterface emulation environment, CloudSigma has recommended multi-core processor distribution and NUMA to be enabled. This is especially true should more than 8 CPU cores be provisioned per VM, as the benefits of NUMA are best realised beyond such a threshold. This performance optimisation will lower OAI processing latency, as NUMA minimises CPU cache misses, which leads to the requirement of data to be fetched from the much slower main memory pool.

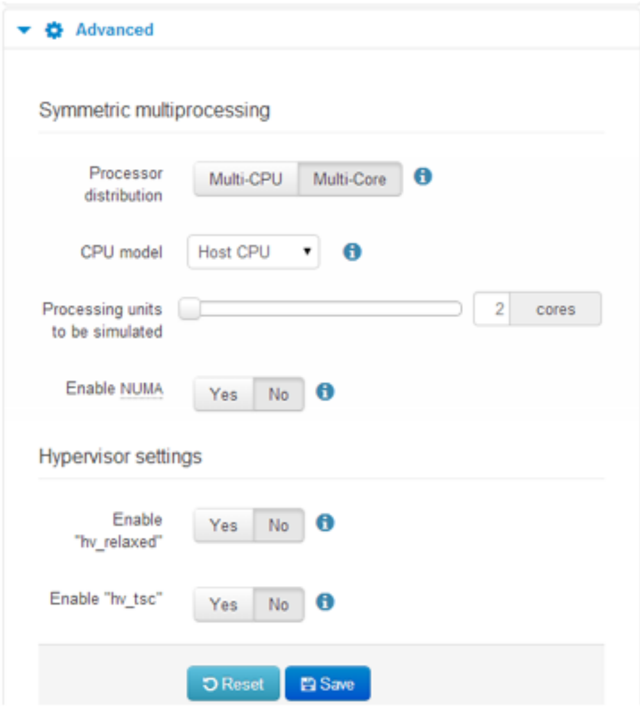

Figure 11 – CloudSigma web app advanced setting screenshot

Figure 11 shows the extended *Advanced* setting window in the CloudSigma Web App with which users can set multiprocessing options and hypervisor settings. To help Service Owners, a series of online tutorials have been made describing the various functionalities of the Web App. They have been made public on CloudSigma's YouTube channel.

3.3.1.3 Potential Issues and Possible Solutions

According to the requirements of OpenAirInterface, support for real-time management and real-time processing is deemed to be essential. However, the OS onto which the cloud stack is deployed, and in turn hypervisor, does not support real-time management and real-time processing. The CPU is physically pinned to a VM and its entire instruction set is exposed to the VM. This in itself yields performance benefits over other cloud stacks. However, being a cloud environment, the CPU can be shared between VMs and is not dedicated. This is an inherent feature of cloud computing, especially in public cloud, where resources are pooled and shared. From there on it is down to features, performance guarantees, and stability that differentiate (public) cloud providers.

The concerns raised within Task 3.5 further highlight the importance of identifying different classes of workload, with hard limits, which must not be breached. Task 3.2 requires these limits to be committed to coded performance tests, so the typical cloud environments that the SICs will run on can be exercised to gauge the feasibility of "cloudifying" the SICs. Performance optimisations and VM parameter tweaking, VM processing capacity, parallelization of workloads and SIC algorithm analysis via profiling tools can be explored by T3.2 to maximise the feasibility of cloudifying these telecoms SICs, with their inherent 'real-time', low latency requirements. Once performance tests are written, the infrastructure can then be tuned to ensure the SICs will be able to perform without breaching latency limits. This is the subject of on-going work between T3.5 (RANaaS) and T3.2 (Performance), the initial findings having been documented as part of Deliverable D3.2 (D3.2, 2014) and more conclusive findings are expected to appear in D3.3 (D3.3, 2014) due in month 26.

3.4 EPCaaS and IMSaaS Testbed Requirements

In order to be able to accurately determine the deployment scenarios of the SICs and have them perform within QoS constraints, their requirements will need to be identified of the underlying testbeds. The effects of over-provisioning and potential variable cloud performance need to be further explored using on-going automated quantitative performance testing, as previously mentioned. This is especially relevant for bursting onto public infrastructure at times of heavy service load. The services' QoS requirements when undertaking different types of workload needs to be defined. Their latency requirements, especially when coupled with other services need to be elaborated. How the different SICs communicate when they are provisioned on different testbeds needs to be developed further. For example:

- For low-latency services provided by IMS such as voice and video, RAN and EPC will have to co-exist in the same testbed. This is due to the low-latency QoS requirements between IMS and EPC of such IMS services.
- RAN and EPC will not necessarily need to be hosted on the same testbed if other services are to be used, for example for data transfer.
- EPC must take into consideration the QoS requirements of services, which will depend on it and the sessions that will need to be instantiated on EPC instances offering these QoS guarantees.

This brings up the following trans-service performance considerations, which must be taken into account when engineering a possible deployment of MCN:

- Services must make clear their QoS requirements of their dependent services by stipulating metrics such as bandwidth requirements, latency and jitter limits, as well as acceptable packet loss boundaries. This will enable them to be best paired with instances of dependent services offering such QoS. This is of significant importance considering the possibility of service instances being able to run in different data centres, meaning inter-data centre network latency will affect QoS.
- A frequent 'heartbeat' ping between data centres hosting the virtualised services should report to MaaS. This data should then be made available to the decision part of the EPC service orchestrator in order for it to be able to allocate appropriate EPC instances to sessions with

specific QoS requirements. This could be implemented as a recommendation service, an application of AaaS.

As previously stated, the real-time performance requirements of RAN dictate that RANaaS will always be running in data centres, never more than 10km from the RRHs. IMS also has real-time performance requirements and depends on EPC to supply them. In such scenarios, EPCaaS and IMSaaS should run in the same data centre as RANaaS, or in a data centre in close proximity, ensuring the latency requirements between RANaaS, EPCaaS and IMSaaS are not breached. The distributed nature of the services not only raises latency concerns, but equally importantly for voice traffic, inter-service network quality and bandwidth availability. In cases where traffic does not have real-time QoS requirements, for example web traffic or media streaming, EPCaaS can process traffic in data centres that do not necessarily offer real-time performance guarantees, potentially in remote locations relative to the RRHs. This is of enormous significance, for three primary reasons:

1. The majority of mobile data traffic today is not voice, and this non-latency sensitive traffic is experiencing exponential growth (D3.2, 2014).
2. Data traffic places the greatest load on Telco infrastructure, especially at spontaneous social events.
3. The non-real-time nature of this traffic makes it possible for its processing to occur in data centres more remotely located than the core data centres placed around the RRHs. This enables bursting to remote, less loaded data centres or even to public cloud providers.

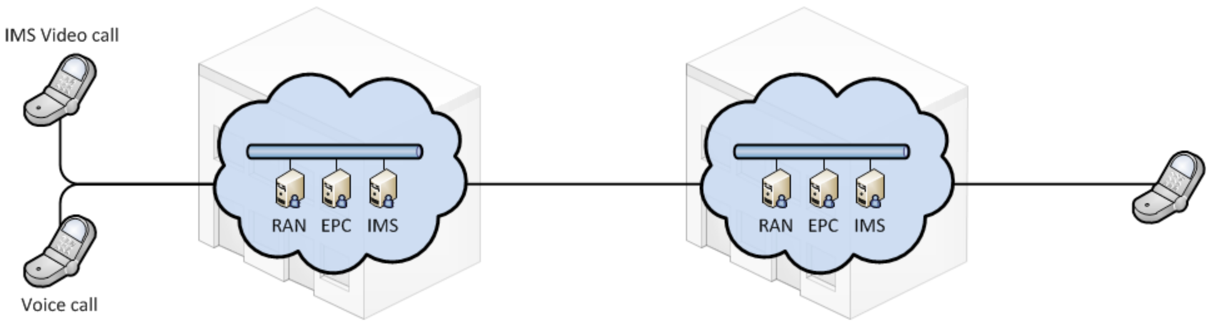

Figure 12 – (Private) clouds, offering real-time guarantees, positioned close to RHHs

Figure 12 illustrates a RAN, EPC and IMS deployment in a private, operator supplied data centre, close to the RRH's of both EUs.

An additional diagram, Figure 13, illustrates an example deployment of EPCaaS on a remote data centre, offloading the processing of latency-insensitive packets to clouds outside of these operator data centres close to RRHs. This ensures that infrastructure is reserved for processing voice data, lowering investment costs per RRH data centre. It also offers a bursting strategy when data traffic grows beyond this RRH data centre's capacity.

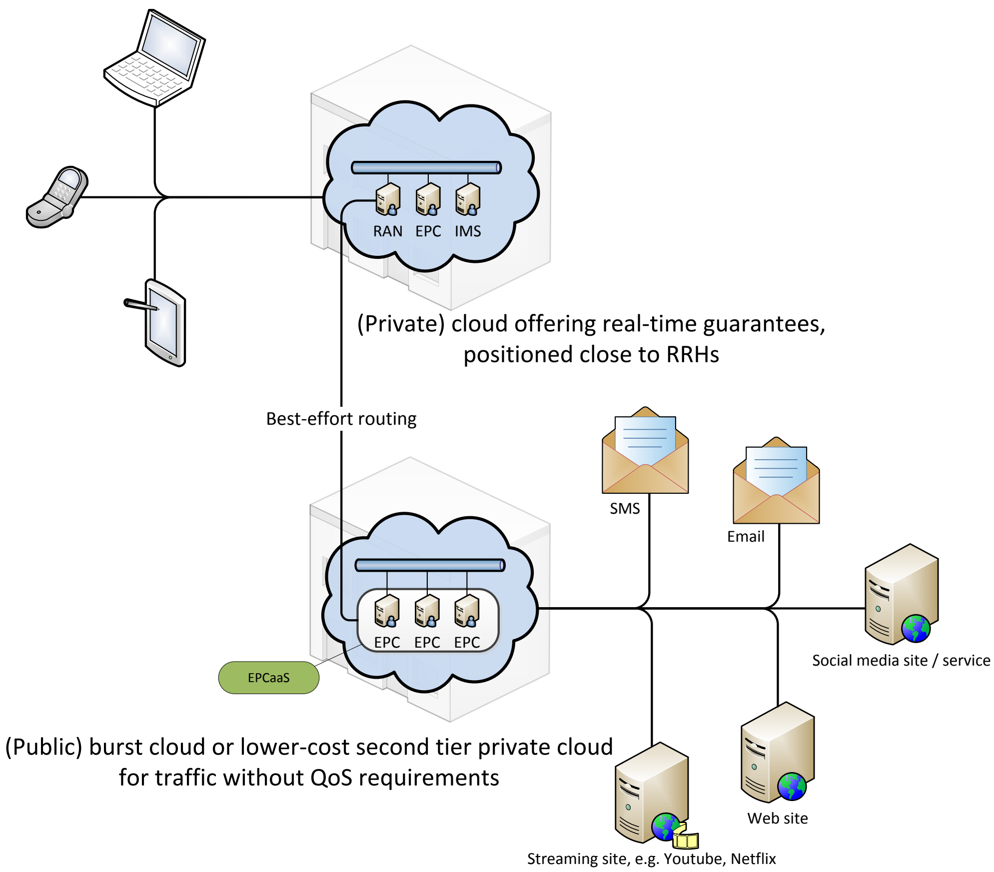

Figure 13 – MCN Services distributed over multiple clouds

Figure 13, illustrates an example deployment of EPCaaS, and all the EPC comprising components, on a remote data centre, offloading the processing of latency-insensitive packets to clouds outside of these operator data centres close to RRHs. This ensures that infrastructure is reserved for processing voice data, lowering investment costs per RRH data centre. It also offers a bursting strategy when data traffic grows beyond this RRH data centre's capacity.

3.5 DSSaaS Testbed Requirements

A classic Digital Signage Network (DSN) includes three basic components: Central Server (CS), to manage the user information, a MCR, to store the user's content, and player (screen, TV, etc.) to display desired information.

In this model DSS Players are directly connected to the CS and MCR for getting the contents. It is also needed to have Central Server and Main Content Repository connected for managing the contents uploaded by the End User (EU). Finally the connectivity between End User and Central Server should be provided making them able to access administration panel.

In summary, basic requirements from the hardware point of view is to have a high speed connection between the CS and MCR and a sufficient amount of storage space on MCR offered to the DSS EEU to ensure there is enough space for storing required media files. Also it is required to have a reliable connection from players to CS and MCR.

Moving to MCN cloudified approach, DSS as an over the top service in the MCN network requires permanent connectivity to players across the RAN to send multimedia content to be reproduced in players (mainly VoD files). But here it is not mandatory to have permanent low latency for ensuring content to be delivered. Because unlike in IMS, for DSS there is no live streaming media content to be manage.

For a DSS integration testbed, in addition to previous requirements, the following components need to be present:

- Cloud Controller: for providing PaaS functionality and making DSS instantiation possible.
- RANaaS: It would be mandatory to provide connectivity to the players.
- CDNaaS: The responsibility for the distribution of media contents will go to this service which results reducing network load, network latency and improving the availability of content, thanks to the web caching as the CDN end nodes will be deployed in the micro data centres to bring the contents closer to the players. To ensure the download times for the multimedia content respond to QoS parameters, times are set to be monitored which will also provide a continuous measure of the CDN performance.
- EPCaaS: Although DSS is not directly connected to EPC but an instance of EPC should be available for taking care of the data connectivity between DSS, CDN and RAN service.

3.6 Maintenance and Operation

Usage considerations and maintenance responsibilities of the updated testbed components are discussed in this section. Stress test and scalability plans are touched upon, but these will need more thorough attention as the MCN architecture and service requirements reach maturity. Finally, strategies for facilitating inter-cloud connectivity and possible performance penalties resulting from implementation approaches are discussed.

3.6.1 Usage/Deployment

Currently MCN service stacks making use of CloudSigma's Heat endpoint should not contain OpenStack resource types, but rather only CloudSigma resource types. OpenStack based VMs cannot be provisioned on CloudSigma's testbed efficiently. Service templates should at this time be deployed on service orchestrators at ZHAW's testbed, with the provisioning of resources to be done by the ZHAW design module. The provisioning of CloudSigma resources can be done by referencing the CloudSigma region and by passing the appropriate Heat template with CloudSigma-specific resource types. For development purposes, the CloudSigma Heat endpoint can be referenced directly and a suitable template can be passed directly to it.

3.6.2 Stress-test Plans and Scalability

The service orchestrator will be responsible for this based on MaaS metrics and identified QoS limits, which will trigger new service instances. OpenShift contains such functionality for gears and compute instances, as does Heat for VMs and containers. The service orchestrator can also perform scaling functions, should this be developed as part of its decision logic, to be linked to MaaS and possibly AaaS. Stress tests of services have been conducted in T3.2 and will continue to be conducted moving forward. Service owner cooperation is key in order to establish scaling triggers for services.

3.6.3 Maintenance

The ZHAW testbed is set to maintain its Havana based OpenStack cloud for the foreseeable future, with a secondary performance testing testbed being made available based on Icehouse. CloudSigma operates a continuous integration release schedule for its ever-evolving cloud and maintenance is inherent to this development pattern.

CloudSigma follows a test-driven-design ethos and as such the CloudSigma Heat plugin has been subjected to a range of tests to ensure feature completeness and specification conformity. It has been made available on Github under the Apache 2.0[3] license for community maintenance and development.

3.6.4 Connectivity

Direct fibre patching and Multi Protocol Label Switching (MPLS) tunnels were briefly explored in order to facilitate a low-latency connection between data centres hence clouds. The financial costs of going down this route have proved prohibitive and at this time this strategy will not be actively pursued.

VPN connectivity between services and testbeds is being developed in T3.1. This will be fundamental to the deployment of a (secure) distributed-cloud setup. CloudSigma has an OpenVPN software appliance developed that enables the rapid deployment of OpenVPN Access Server within CloudSigma's public cloud. This has been communicated with T3.1 as a possible VPN solution to ease the inter-cloud connectivity, which is currently being analysed.

3.7 Roadmap and Open Issues

Support services such as MaaS are being deployed and integrated on CloudSigma's cloud in order to make possible a full-featured service stack deployment. To facilitate MaaS, popular industry standard Zabbix is best suited for CloudSigma's cloud due to CloudSigma's independence from OpenStack native monitoring components, namely Ceilometer. To enable the reporting of monitoring metrics to MaaS, each MCN service drive image should have the appropriate Zabbix agent installed.

STGs for each service in the form of Heat templates need to be written with the CloudSigma resource types in mind. Once service owners become familiar with CloudSigma's available resource types, this should be quite straightforward. When combined with an analogous OpenStack specific template, this will allow for a service stack to be brought up on either cloud. The difference between cloud STGs, hence the difference between bringing up an MCN service stack on one cloud or another is primarily a matter of amending the resource types between clouds in each cloud-specific Heat template.

Distributed-cloud benchmarking will be further explored once services are running on both CS and ZHAW's testbeds to begin with, with other testbeds to follow suit pending successful service integration between these two testbeds. This could involve the benchmarking of inter-VM latency utilising a VPN connection and comparing the results to an unsecure direct connection between VMs over public Internet.

---

[3] https://github.com/cloudsigma/cs-heat-plugin

4 First Integration of Components

Following the plans laid down in D6.1 (D6.1, 2014), an integrated prototype has been under construction by the MobileCloud Networking (MCN) project.

4.1 Introduction

For simplicity reasons, regarding this first version of the integrated prototype, a single reference integration testbed was selected, simplifying the process of verification of dependencies and maintenance of all the needed and deployed MCN components. This testbed is located at ZHAW[4], which provided access to all MCN developers. Regarding the development of the Services and MCN modules, each team, co-ordinated by its respective Service Owner, is responsible for their own deployments and components' sanity checks, making use of the resources and functionalities provided by ZHAW's testbed.

An iterative process of integration was executed, currently including a large number of the MobileCloud Networking components, proving that a complex mobile cloud-networking environment is feasible and that inter-operability can be achieved within a limited time scope even though it is based on a highly heterogeneous set of components.

With more than 21 integration procedures executed, the current MCN prototype already presents a comprehensive view on the different components and their integration, as described in Section 4.3.

The integration results presented in this section do not aim at detailing implementation achievements; these are presented in the scope of work packages 3, 4 and 5 where development is performed. This section provides a single, concise mirror on the advancements of the overall end-to-end MCN prototype and its current integration. While different MCN Services and Components are at different stages of development, they are analysed as a whole, from the integration task perspective, due to the defined external interfaces and the integration tests between each. Additional information and defined solutions will be further provided whenever specific details are worthy of being mentioned, in particular regarding specific issues that were encountered.

The following subsections present the current integration status of MCN, including the different software components to be considered for the MCN Integrated Prototype System, due on M30, and motivating the upcoming integration activities to be reported in D6.3 due on M26.

4.2 Integration Status

In order to provide a single generic view of the integration status, a unified figure for the MCN Integrated Prototype System was defined, simplifying its overall understanding when compared to the previous, rather complex, version from D6.1 annexes. Further details on the specific implementation items are presented into each of the equivalent implementation deliverables and not detailed here, as they are not directly in the scope of the integration task. Figure 14 presents this new perspective on the integration status.

[4] Bart testbed, based on OpenStack Havana: http://bart.cloudcomplab.ch

In order to achieve the desired straightforwardness reasons, each MCN and support service is represented as a single box, including three different classes of components:

- The Service Manager, as a north-bound interface;
- The Service Orchestrator, as the main orchestration component;
- The Service Components from which the service consists of.

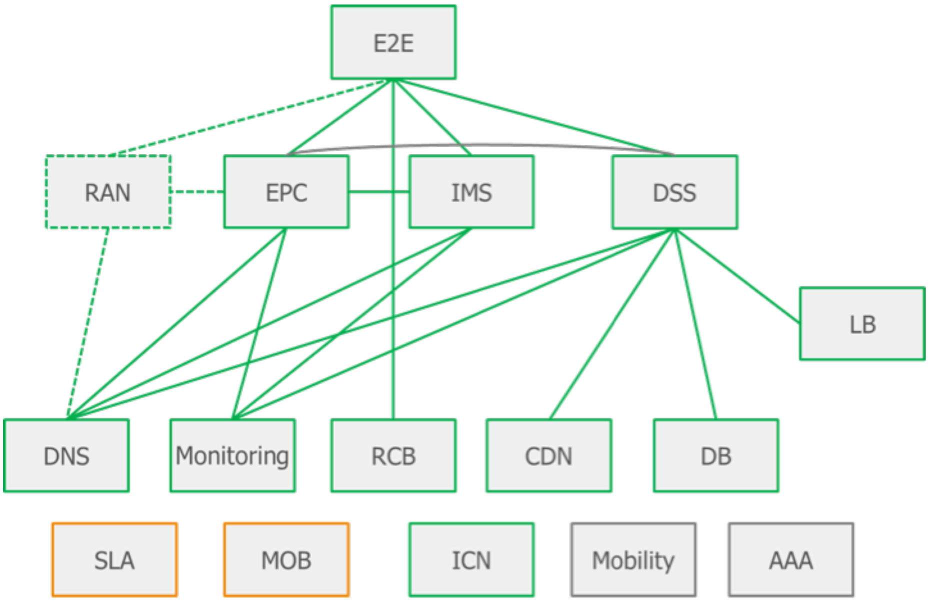

Figure 14 – Month 24 Mobile Cloud Networking Integrated Prototype

The relationship between the presented components and a closer analysis of Figure 14 is presented in the following sub-sections, describing the integration activities that already took place and those that are planned for upcoming iterations.

Specifically for the RAN part (dotted lines), currently the Fraunhofer OpenEPC eNB emulator is used, which does not present enough realistic scenarios. Even though an end-to-end prototype is already integrated, as depicted in Figure 14, the RAN part will be replaced at a later date with Eurecom's Open Air Interface.

The services in the bottom layer do not yet have any integration interface in the final integrated prototype system, mainly due to the gradual prototype development as well as to the gradual prototype integration. This list includes services such as SLA, MOB, ICN, Mobility and AAA, which will be handled in the future.

4.2.1 Deviations from the Prototype's Architecture

While executing and achieving the most efficient integrated prototype system that was possible, a set of variations from the functional architecture included in Deliverable 2.2 (D2.2, 2013) were observed and are further explained. These variations have been reported to Task 2.3 and will be included in the next iteration of the architecture deliverable (M30) and will enable the nuancing of the previous rather uniform architecture into a more realistic deployment environment.

During the first prototype system integration in month 18, the important role played by the E2E SM and SO was clearly observed. For this reason, the E2E Service Composition is depicted as a separate component, which aims to provide at a service level, the end-to-end virtual network infrastructure.

Additionally, during the prototype integration, it was observed that a specific set of support services, which are more part of the underlying virtualisation support platform than of the actual services, are controlled directly by the end-to-end service orchestrator specifically the RCB. The RCB comes as a support service for the end-to-end virtual service architecture, which uses it as a client, similarly to the compute and storage and networking resources reservation and thus becomes part of the platform fabric. As such, being an extension of the underlying prototype architecture, it has to be controlled in a similar manner. However, due to the end-to-end composition feature, it cannot be part of the underlying CC functionality and has to be controlled by the end-to-end orchestrator. Similar considerations are expected to have to be made when addressing the mobility support/DMM from a networking perspective. Regarding the other support and main services, no deviation has been considered, compared to the plan in D6.1.

4.2.2 MCN Services Integration

Currently, for the MCN services integration, there is a single system (as integrated and demonstrated in M18), which includes the RANaaS based on the Fraunhofer OpenEPC eNB emulation, the EPCaaS based on the Fraunhofer OpenEPC (with a modified architecture implementation) and the OpenIMSCore. This provides an all-together integrated Telco-oriented virtual network infrastructure. Through this integrated prototype system typical multimedia Telco services can be deployed such as voice calls, SMS, presence, among others. Additionally, there is the means for breaking out the data path directly from the EPCaaS towards the Internet, being able to provide basic Internet services to the subscribers without requiring a service platform. In particular, these two features are prepared to run in parallel without requiring further configurations.

In a separate prototype, the DSS Service has been integrated with CDN and other services, as described in the PoC (Section 2). Due to the nature of the integration in which the CDN is offering caches to the DSS service and does not provide a direct perceived service to the end user subscriber, the CDN in this case acts more as a support service and not as a standard MCN service. This feature will be further considered in the next architecture evolution.

For the time being there is no completed integration with Information Centric Network (ICN). Further details have to be considered in this area, as the EPC (S1 and S5 level), and especially the RAN (RRC level), are designed to carry only IP level messages, not being suitable for establishing an end-to-end layer-2 connection.

Additionally, due to different end-user subscriber devices used (differences in Operating System), there were no means currently to run the DSS service on top of the RAN-EPC infrastructure and in parallel with the IMS and with the internet service. A mitigation action was taken for creating also an integrated user device running all the counterpart software for RAN, EPC, IMS and DSS and thus to provide the means to run a single infrastructure with multiple application level platforms deployed in parallel.

In order to provide a more realistic RANaaS implementation, a decision was taken to shift the RAN implementation to the Eurecom OpenAirInterface platform. Due to tight connection with the EPC based on complex and standard interfaces, it is foreseen that a rather difficult integration procedure has to be executed. A careful attention will be given to this item in the next stage.

4.2.3 Support Services Integration

A major integration step for a large number of services was the integration with the **monitoring as a Service (MaaS)**. Currently, the MaaS is able to retrieve information from the infrastructure level, integrated supported services, as well as from the RAN, EPC, DSS and IMS, making it suitable to provide the dynamicity basis for the further advancements for elasticity of the virtual network services. At this stage MaaS information is used by Service Orchestrators to perform several operations such as elasticity-based decisions.

Rating Charging and Billing as a Service (RCBaaS) uses an implicit generic platform and has to be later integrated while the MOBaaS requires a more granular subscriber information from the system which may not be achieved in a scalable manner by the EPCaaS and should be acquired from external sources.

All the services that require DNS for their internal functionalities (e.g. RAN, EPC, IMS and DSS), resort to **DNSaaS** for performing the necessary queries and configurations. Although further integrations are envisaged, they are considered as of secondary importance, as the DNSaaS is already providing its supporting value for four MCN services.

Due to its limitation to HTTP-based platforms only, the **LBaaS** is integrated and is used only by the DSS Service and not by the other services. With this integration, there are no more other items to be executed for LBaaS.

Database as a Service (DBaaS) is currently integrated with the DSS. Although tests were executed for its integration with the EPC and the IMS Home Subscriber Server (HSS), for providing a distributed and elastic back-end for the subscriber information database, these features are not yet part of the overall MCN integrated prototype system. The suitability of their integration depends on the discovered bottlenecks while benchmarking a virtual EPC/IMS infrastructure. At this moment we assume that this integration is not necessary since the main factors that may deter scalability are foreseen in the data path and partially in the control plane.

SLAaaS and **MOBaaS** are currently still in development phase and their integration into the end-to-end prototype will occur until the end of M27 the latest. Similar considerations have to be made for the **Mobility/DMM** functionality. However, as this functionality integrates directly with the underlying infrastructure platform more considerations on the impact to the end-to-end service have to be made.

For the **AAAaaS** a decision on the specific integration interfaces was made. The service will integrate with the IMS/EPC HSS and through this means a centralised authentication and authorization mechanism will be provided for the Telco and for the IT-based services.

4.2.4 MCN SDK

As illustrated in Figure 15, the MCN SDK currently offers DNS, monitoring, CDN, DB and LB as supporting services for the MCN services. For RCB an integration procedure is also considered, however depending on the settlement whether RCB is part of the infrastructure – being controlled directly by the infrastructure manager (i.e. OpenStack) – or whether it is part of the services running on a top-level, independently. Bearing this in mind, the RCB has not yet been integrated into the MCN SDK.

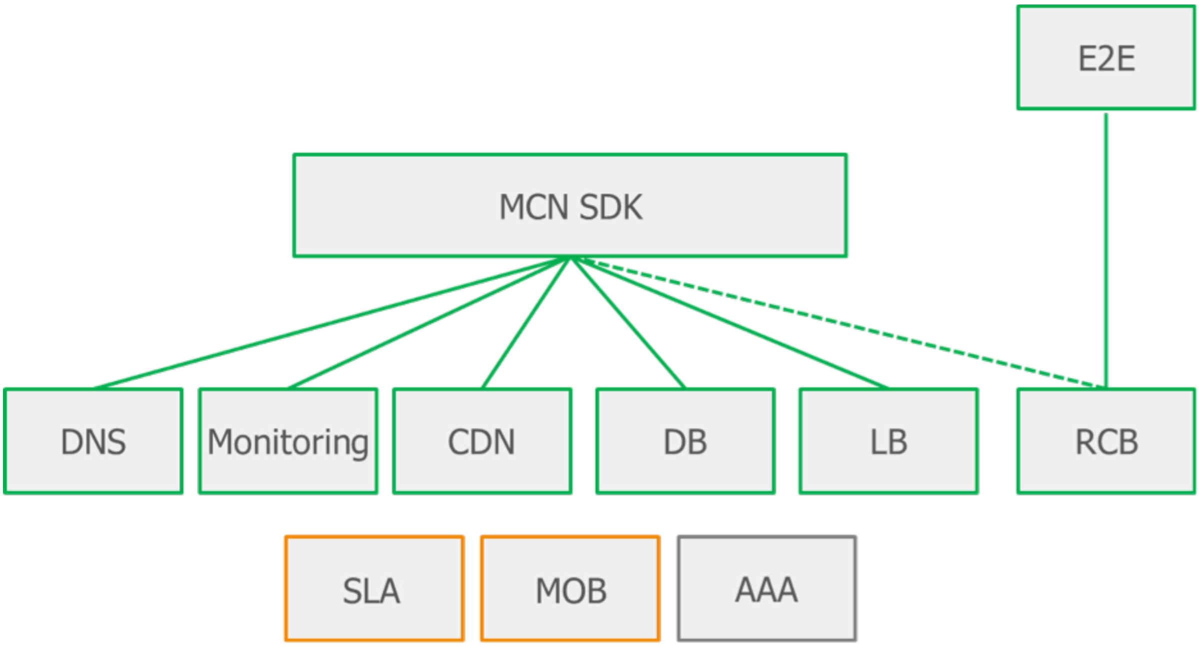

Figure 15 – MCN SDK

Further integration is considered with SLAaaS, MOBaaS and AAAaaS, being these activities postponed until these services reach an appropriate maturity stage in their development.

4.3 Defined Interfaces

In this section a mirror of the testing and integration interfaces is presented in a table form. A conservative approach was taken for the interfaces here included, accounting only for the specific ones, which are part of the integrated prototype. All other interfaces related with the MCN SM and SO will use the Open Cloud Computing Interface (OCCI).

Currently, bi-lateral interfaces integrated and interfaces still in integration (tests not completed in the com testbed) are not included. A further, more extended report will be available as part of D.6.3, including the different tests executed.

Regarding the testing and integration procedures, several simplifications were possible by using a single integration environment, compared to the highly formalised description from D6.1:

- Each of the software development teams was in charge of installing and testing their own components on the integrated testbed. A loose approach to unit tests was taken, in which only the interfaces exposed to the other components were tested. Each team was in charge for its own sanity checks.
- Regarding pre-conditions of testing, as the same testbed was used, there was no need for its description. Additionally, the API provided by the lower level components (as depicted in the previous subsections) is considered the API that has to be integrated, thus requiring a single presentation – available in the first implementation deliverables and further extended for the future second implementation deliverables. The people involved in the integration task were the same as the main responsible for the implementation part. This was possible due to a rather reduced size of the implementation teams.

- A single set of test conditions was considered in which the environmental conditions were the ones provided by the single integration testbed. A basic, functional set of input data was considered for each of the integration interfaces as well as for each of the expected results.
- For each service-specific interface its respective service owner will provide a client-side library, allowing third-party services to use this library while avoiding duplication of efforts in integration.

Table 7 – Defined interfaces and their status

Interface	Status	Notes
UE-RANaaS	Not started	OAI integration (needs EURECOM). OpenEPC version already integrated
RANaaS-EPCaaS	Not started	OAI Integration with EPC (needs EURECOM). OpenEPC version already integrated
EPCaaS-IMSaaS	DONE	
EPCaaS-DSSaaS	Started	Depends on the integration of the DSS player with the EPC client
EPCaaS-CDNaaS	N/A	CDNaaS is now considered a support service which is not used by EPCaaS
EPCaaS-ICNaaS	Not started	High incompatibility between the two systems has to be mitigated
RANaaS-DNSaaS	Not started	OpenEPC RANaaS was integrated, not the OAI version
RANaaS-MaaS	Started	A large part of the interfaces was developed (OAI version), not yet tested within the integrated prototype
RANaaS-SLAaaS	N/A	Integration through the Cloud Controller SDK
RANaaS-RCBaaS	N/A	Integration through the Cloud Controller SDK
RANaaS-MOBaaS	N/A	Prediction is considered not to offer fast enough results
EPCaaS-MaaS	DONE	
EPCaaS-RCBaaS	N/A	Integration through the Cloud Controller SDK
EPCaaS-SLAaaS	N/A	Integration through the Cloud Controller SDK
EPCaaS-MOBaaS	Not started	The integration provides limited evaluation results, apart from the functional testing, as a real mobility and QoS trace is not available to be played by the EPCaaS.
IMSaaS-MaaS	Started	
IMSaaS-DNSaaS	DONE	
IMSaaS-RCBaaS	N/A	Integration through the Cloud Controller SDK
IMSaaS-SLAaaS	N/A	Integration through the Cloud Controller SDK

IMSaaS-MOBaaS	N/A	No considerations from MOBaaS on the scenario
DSSaaS-MaaS	DONE	
DSSaaS-DNSaaS	DONE	
DSSaaS-RCBaaS	N/A	Integration through the Cloud Controller SDK
DSSaaS-SLAaaS	N/A	Integration through the Cloud Controller SDK
DSSaaS-CDNaaS	DONE	
DSSaaS-DBaaS	DONE	
DSSaaS-LBaaS	DONE	
DSSaaS-MOBaaS	N/A	No considerations from MOBaaS on the scenario
ICNaaS-MaaS	Started	
ICNaaS-DNSaaS	Not started	
ICNaaS-RCBaaS	N/A	Integration through the Cloud Controller SDK
ICNaaS-SLAaaS	N/A	Integration through the Cloud Controller SDK
ICNaaS-CDNaaS	N/A	Divergent technologies for similar goal
ICNaaS-MOBaaS	N/A	No considerations from MOBaaS on the scenario
CDNaaS-MaaS	Not started	
CDNaaS-SDK	DONE	
CDNaaS-DNSaaS	Not started	
DNSaaS-SDK	DONE	
MaaS-SDK	DONE	
DBaaS-SDK	DONE	
LBaaS-SDK	DONE	
SLAaaS-SDK	Started	
MOBaaS-SDK	Not started	Specific target service, the opportunity of its integration has to be further assessed
AAAaaS-SDK	Not started	Specific target service, the opportunity of its integration has to be further assessed
AAAaaS-EPCaaS	Started	
RCBaaS-SDK	Started	
RCBaaS-MaaS	Not started	
RANaaS-E2E	Not started	Eurecom joined late the project
EPCaaS-E2E	DONE	

IMSaaS-E2E	DONE	
DSSaaS-E2E	DONE	
ICNaaS-E2E	Not started	
CDNaaS-E2E	N/A	CDN is a support service

Summing up, out of 36 integration tasks 16 have been completed (44%), with 7 other tasks already being conducted and 13 yet to be started.

4.4 Roadmap and Open Issues

The following list describes the current open issues both from a technical and an integration perspective. Different features concern several development teams and have to be executed in parallel; therefore no specific priority or importance was considered, knowing that at the end all of them have to be completed.

1. Integration with OpenAirInterface (OAI) – a major integration has to be executed for providing an integrated OAI with the framework (SM, SO and communication with the end-to-end SO), with the EPC (conformant to the 3GPP S1 interfaces) and with the overall underlying infrastructure (connectivity to devices, IP addresses allocation, etc.)
2. Integration with Mobility/DMM – in order to be able to provide dynamic scalable data paths for the elasticity, there is a strong need from the underlying platform to support dynamic routing mechanism and flexible networking transparent to the network functions themselves.
3. Integration with currently in development support services – SLAaaS, MOBaaS and AAAaaS – for providing a comprehensive architecture, integration with the remaining support services is needed. Additionally, there is a need for refining and completing the operations for the other services.
4. OpenStack issues – the underlying virtual infrastructure manager (OpenStack) has major limitations in regard to networking capabilities, which are not in the scope of MCN to solve especially in the area of routing and NAT functionality. For example, OpenStack cannot route packets that do not have as source address the IP address allocated to the specific component, thus severely limiting the capabilities of deploying virtual routers or gateways. Additionally, a standard deployment of OpenStack introduces a specific NAT for being able to reach any of the components from an external network, which was not considered when the Telco network was developed e.g. between the RAN and the EPC or within the EPC. Several actions were already taken to overcome these issues through providing end-to-end virtual tunnels between different components, however these solutions are only able to provide a proof-of-concept prototype and do not adhere to a graceful scalable architecture.

Apart from the integration tasks, which have to be further explored and executed, a major concern remains in what regards to the runtime elastic scaling of components and their impact on the other components of the end-to-end system. With month 24 a decision was taken, in which components are expected to elastically scale and, based on the elasticity implementation, further integration activities will have to be executed in another integration phase, due on M30.

5 Evaluation Plans

Following the plans previously described in D6.1 (D6.1 2014), evaluation plans have been detailed for the MCN services. This section refers to the overall definition of this evaluation planning and Appendix A describes them in more detail, presenting also some preliminary results obtained for some services. This process has been executed in the scope of Task 6.5, which started its activity in M22.

The evaluation work done under the scope of WP6 is mainly experimental using testbeds, prototypes (basic and final) and proof-of-concept applications (IMS and DSS) co-ordinated by the remaining tasks of this work package. Technical WPs (WP3, WP4 and WP5) have made an initial experimentation and evaluation work, providing a valuable input for WP6 evaluation activities.

5.1 Introduction

The initial evaluation planning and results described in this document follow the methodology described D6.1 (D6.1 2014). This methodology is recapped in the following subsection.

Here we defined the strategic evaluation of objectives, according to the requirements defined in WP2 and presented in (D2.1 2013) as well as those defined in WP6. These strategic objectives include the use of the appropriate selection of Key Performance Indicators (KPIs) (qualitative and quantitative) suitable for evaluation. Along with this selection of indicators, it is important to set the qualitative targets and expected results to be achieved per each of the indicators. These targets and expected results provide a clear identification of "success", otherwise would mean that there is no real notion of whether the effort and investment will bring the desired benefits. Moreover, only with a recognized success there can be a reasonable potential for the adoption of the proposed solution.

The selection of indicators followed the following criteria:

- *Relevance*: an indicator should represent an assessment criterion, i.e. have a significant importance for the evaluation process.
- *Completeness*: the set of selected indicators should consider all aspects of the solution under evaluation.
- *Measurability*: the identified indicators should be capable to be measured objectively or subjectively.
- *Familiarity*: the indicators should be easy to understand.
- *Non-redundancy*: indicators should not measure the same aspect of an assessment criterion.
- *Independence*: small changes in the measurements of an indicator should not impact preferences assigned to other indicators of the evaluation model.

Furthermore, it is necessary to select and test a set of use-cases according to their characteristics and requirements. Having accomplished the abovementioned objectives, the impact evaluation and process evaluation is carried out, by comparing the measured outcomes with the initial definition of success. Finally, this process should then lead to a more deep analysis to the possible adoption of MCN solutions by the different stakeholders.

The following subsections present the relevant methodologies, areas, impacts and indicators considered for each service.

5.2 Methodology and Evaluation Technique

The methodology followed by each service owner, in order to obtain the complete evaluation assessment and respective results is described in detail in (D6.1 2016). Figure 16 depicts graphically the evaluation process.

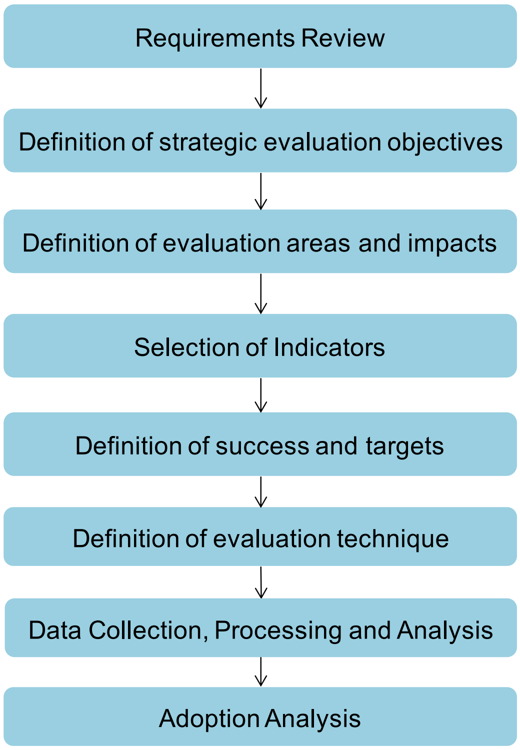

Figure 16 – Evaluation process

The evaluation process comprises the following stages/activities:

1. **Requirements review** – Revision of the stakeholders involved in the prototypes along with their needs and requirements.
2. **Definition of strategic evaluation objectives** – From the revision process the strategic evaluation objectives will be drawn.
3. **Definition of evaluation areas and impacts** – Define the areas to be evaluated and the expected impacts.
4. **Selection of indicators** – Identification of the KPIs required to measure the impact (KPI name, description and data units).
5. **Definition of success and targets** – Define target values for indicators to conclude the success of not of a certain requirement.
6. **Definition of evaluation technique** – Identify the most appropriate technique in order to perform the assessments; in other words, defines the method of measurement, frequency, etc. This stage might have an impact on the testbed setup (introduction of new requirements to allow the assessment).
7. **Data Collection, Processing and Analysis**

   a. Set up of data source – define Datasheets and/or databases for collecting the required data.
   b. Data collection – process of collecting data and storing it.
   c. Data analysis / Identification of success / Impact analysis – process of analysing the collected data.
8. **Adoption analysis** – analyse the potential for the adoption of the MCN solution.

This process will be iterative in order to cope with revisions and further calibrations. For the sake of clarity, the following subsection elaborates on examples for the different phases of the evaluation process. These examples, although in line with the project scope, should not be interpreted as certainties since they will still be subject to further analysis in the work to be undertaken in upcoming tasks. For more details refer to (D6.1 2016), section 5.2.

5.3 Evaluation Areas, Impacts and Indicators

The execution of the methodology resulted in the creation of six tables per service, which compile the most relevant data to be considered in the evaluation. As an example the different tables are presented, with information ranging from the planning of requirements, to the gathering of results and the analysis. Appendix A and its subsections detail all of these tables for the existing services and gathered information.

5.3.1 Requirements and Strategic Evaluation Objectives

This table describes the list high-level requirements and the respective strategic evaluation objective.

Nº	Requirement	Strategic Evaluation Objective

Here is an example:

1	On-Demand Instantiation	Support the on-demand ANDSF service instantiation. The customer can request an ANDSF service instantiation, through a web portal.

5.3.2 Definition of Evaluation Areas, Impacts and Selection of Indicators

This table describes the evaluation area, the impact and the indicators (name and id) that will evaluate that impact.

Strategic Evaluation Objective	Evaluation Area	Impact	Nº	Indicator's name

Here is an example:

Support the on-demand ANDSF service instantiation.	IaaS/MCN Service	Instance Works Properly	1	Works Properly

The customer can request an ANDSF service instantiation, through a web portal.		Time elapsed for complete Instantiation	2	Instantiation Time
		Number of Instances per tier	3	Number Instances/Tier

5.3.3 Definition of Success and Targets

This table determines success or failure according to the defined targets and quantifies them.

Nº	Indicator's name	Target quantification

Here is an example:

2	Instantiation Time	Few Minutes (< ~10 min.)

5.3.4 Definition of Evaluation Technique

This table describes the evaluation techniques used to obtain results.

Nº	Indicator's name	Indicator's description	Data/Unit	Data Source	Method of Measurement	Frequency

Here is an example:

2	Instantiation Time	Time elapsed until full service instantiation	Seconds (s), quantitative	-	Manual/Script Measurement	On every new service Instantiation

5.3.5 Set-up of Data Source and Data Collection

This table defines the results (values) obtained for the different metrics using the evaluation techniques referred above. For the time being, and considering the early stage of the experimentation and evaluation task, only a few services provide preliminary results. The entirety of these results are planned to appear on D6.4.

5.3.6 Data Analysis / Identification of Success / Impact Analysis

This table defines the results (values) obtained for the different metrics using the evaluation techniques referred above. As explained before, the actual assessment to be accompanied with each service is planned for a later stage of the project.

5.4 Roadmap and Open Issues

The presented evaluation plans revisit the initial considerations and roadmap introduced in (D6.1, 2014), which still stands accurate. The following table summarizes the roadmap, identifying the activities to be conducted in order to achieve results for two different phases: finalisation of the experimentation and evaluation plans (presented in this document) and complete assessment and retrieval of results (to be included in D6.4). Each phase comprises all evaluation process activities except for the latter one, the adoption analysis. Due to the proximity of this deliverable (D6.2), with D6.3 (M26), no significant updates are expected by then. However, should important developments occur, an intermediate phase will also consider contributions for D6.3. The adoption analysis activity will only be considered in the end of Phase 2 due to the more consistent set of information that is expected to be available at the time.

Table 8 – Evaluation Activities (according to adopted methodology) Gantt Chart

<table>
<tr><th></th><th>14</th><th>15</th><th>16</th><th>17</th><th>18</th><th>19</th><th>20</th><th>21</th><th>22</th><th>23</th><th>24</th><th>25</th><th>26</th><th>27</th><th>28</th><th>29</th><th>30</th><th>31</th><th>32</th><th>33</th><th>34</th><th>35</th><th>36</th></tr>
<tr><td>Methodology definition</td><td colspan="4"></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Review Requirements</td><td></td><td></td><td colspan="6">Phase 1 (D6.2)</td><td></td><td></td><td></td><td colspan="5">Phase 2 (D6.4)</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Definition of strategic evaluation objectives</td><td></td><td></td><td colspan="6">Phase 1 (D6.2)</td><td></td><td></td><td></td><td colspan="5">Phase 2 (D6.4)</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Definition of evaluation areas and impacts</td><td></td><td></td><td colspan="6">Phase 1 (D6.2)</td><td></td><td></td><td></td><td colspan="5">Phase 2 (D6.4)</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Selection of indicators</td><td></td><td></td><td></td><td></td><td></td><td colspan="4">Phase 1 (D6.2)</td><td></td><td></td><td colspan="6">Phase 2 (D6.4)</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Definition of success and targets</td><td></td><td></td><td></td><td></td><td></td><td colspan="4">Phase 1 (D6.2)</td><td></td><td></td><td colspan="6">Phase 2 (D6.4)</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Definition of evaluation technique</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="3">Phase 1 (D6.2)</td><td></td><td></td><td></td><td></td><td></td><td colspan="6">Phase 2 (D6.4)</td><td></td></tr>
<tr><td>Data Collection, Processing and Analysis</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="3">Phase 1 (D6.2)</td><td></td><td></td><td></td><td></td><td></td><td colspan="6">Phase 2 (D6.4)</td><td></td></tr>
<tr><td>Adoption analysis</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="4"></td><td></td></tr>
</table>

6 Summary and Outlook

This second deliverable of work package 6 completes the planning of all the activities (T6.1) to be conducted until its end. Furthermore, detailed updates regarding the considered proof-of-concept applications and services are provided (T6.2).

The presented PoCs concerning DSS and IMS reveal that the expected activities to be achieved by M24 were successfully completed, setting the path the detailing the upcoming activities for the PoCs until M27.

While integration activities (T6.3) are still undergoing, due to the active development of the MCN Services and Modules, several interfaces have already been integrated, validating the overall MCN architecture. The succeeding endeavours consider the first finalised integrated version of the MCN Prototype System to be presented in the upcoming D6.3.

The established approaches for testbed management and for the MCN distributed-cloud have also been presented in this deliverable, providing the necessary infrastructure and the roadmap for additional testbed support (T6.4) for, upon completion of the integrated prototype, conducting the defined experimentation and evaluation activities (T6.5), measuring service-specific KPIs and overall performance of the MCN framework.

A Appendix A: Updated Testbed Descriptions

The following sections incorporate the tables containing the updated testbed information provided by each partner via the online form entitled Testbed Characterization and Capacity. The descriptions have been updated only in some cases; otherwise the original description from the original proposal has been used, as it remains relevant to date.

A.1 Fraunhofer/TUB

Fraunhofer provides an OpenStack based testbed for the deployment of EPC and IMS. Currently a single OpenStack instance is used. For network functions placement, three OpenStack small size data centres are further considered. A more definitive description update will be provided in the final version of the deliverable.

Table 9 – Fraunhofer/TUB testbed characterization and capacity

Testbed characterization and Capacity	
General testbed configuration	
Hypervisor	Kernel-based Virtual Machine (KVM)
How is VM monitoring made possible on your IaaS stack?	Intra-VM testing tools, at the discretion of the VM user
Access methods (upload, download)	Bilateral communication
Connectivity	License based access
Cloud Interface	
Provisioning	API, Command line Interface (CLI), Web console
API protocol / middleware	OpenStack API
Networking	API
IaaS	
What IaaS stack does your cloud employ?	OpenStack
If OpenStack, what version will be made available for the testbeds?	Grizzly, Havana
If Open Stack, will OpenStack Heat be deployed on your cloud and made available to the testbeds?	Planned
PaaS	
What PaaS can be deployed and made available on the testbeds?	TBD
What version is the PaaS?	N/A
Storage	
Available storage interfaces	-
Image format	Quick Emulator (QEMU) Copy On Write (QCOW)
Solid State Drive (SSD) capacity (GB/per VM)	-
Hard Disk Drive **(**HDD) capacity (GB/per VM)	-
Combined storage capacity	-
What are the maximum real-world IOPS attainable from your storage solution?	-
Compute capacity	
CPU (GHz/core)	-
RAM (GB / VM)	-
Number of VMs	-
Speed (Linux)	-
Speed (Windows)	-
Networking	
What switches do you employ within your cloud?	External hardware switches (not under the testbed control).
What novel and relevant to MCN functionality do	Currently no functionality is considered.

your switches offer users of your cloud?	
Maximum internal network bandwidth made available per VM (in Gb)	-
Maximum external network bandwidth made available per VM (in Gb)	-
Maximum inter-VM latency in *ms*	-
Total cloud external network bandwidth (in Gb)	-
Software Defined Networking	
Is SDN available on your cloud infrastructure?	Yes
What SDN protocols does your infrastructure support?	OpenFlow
What SDN hardware switches does your infrastructure support?	Under acquisition process
What version of the SDN protocol does your infrastructure support?	OpenFlow 1.1
Other relevant components/hardware	
Does your testbed provide other relevant components?	

A.2 PTInS

PTIN will provide a testing environment composed of:

a) OpenStack platform (2 OpenStack instances – to have a scenario with 2 CSPs/DCs); Five cloud nodes currently available (more nodes will be added in the future if needed);

b) Network Connectivity Provider (NCP) IP/ Multi Protocol Label Switching (MPLS) network - Nine Cisco routers integrated in the testbed, in order to reproduce typical NCP conditions.

NCP OpenFlow-based Network – OpenFlow network in order to reproduce the case in which the NCP has provides new features.

Table 10 – PTIN testbed characterization and capacity

Testbed characterization and Capacity	
General testbed configuration	
Hypervisor	KVM
How is VM monitoring made possible on your IaaS stack?	Intra-VM testing tools, at the discretion of the VM user
Access methods (upload, download)	API via Hypertext Transfer Protocol Secure (https)
Connectivity	Internet VPN
Cloud Interface	
Provisioning	API, Web Console
API protocol / middleware	OpenStack API
Networking	API, Web Console
IaaS	
What IaaS stack does your cloud employ?	OpenStack
If OpenStack, what version will be made available for the testbeds?	Havana
If Open Stack, will OpenStack Heat be deployed on your cloud and made available to the testbeds?	Planned
PaaS	
What PaaS can be deployed and made available on the testbeds?	OpenShift
What version is the PaaS?	-

Storage	
Available storage interfaces	Block device storage, Volume storage, Object storage
Image format	QCOW2
SSD capacity (GB/per VM)	-
HDD capacity (GB/per VM)	-
Combined storage capacity	-
What are the maximum real-world IOPS attainable from your storage solution?	-
Compute capacity	
CPU (GHz/core)	-
RAM (GB / VM)	-
Number of VMs	-
Speed (Linux)	-
Speed (Windows)	-
Networking	
What switches do you employ within your cloud?	-
What novel and relevant to MCN functionality do your switches offer users of your cloud?	-
Maximum internal network bandwidth made available per VM (in Gb)	-
Maximum external network bandwidth made available per VM (in Gb)	-
Maximum inter-VM latency in *ms*	-
Total cloud external network bandwidth (in Gb)	-
Software Defined Networking	
Is SDN available on your cloud infrastructure?	Yes
What SDN protocols does your infrastructure support?	OpenFlow
What SDN hardware switches does your infrastructure support?	Open vSwitch
What version of the SDN protocol does your infrastructure support?	-
Other relevant components/hardware	
Does your testbed provide other relevant components?	-

A.3 FT/Orange

Orange makes available as an MCN testbed some commercial eNBs from different Ericsson and ALU. They support the Long Term Evolution (LTE) Rel. 8 with 20MHz 2x2 MIMO at 2.6GHz and 10MHz 2x2 MIMO at 800 MHz. Different fronthaul transport options can be implemented between RRH and Base Band Unit (BBU) (max 15km distance for Alcatel Lucent and 25km for Ericsson). CPRI and radio metrology equipment are also available to generate and evaluate CPRI and radio performances. Both eNBs are connected to Orange EPC (Paris). SynchE provides synchronization to BBUs.

Due to the fact that FT/Orange testbed is not based on cloud infrastructure only a limited number of fields from the online form were deemed relevant and therefore left blank. However the description above was provided which includes the current capacity and characteristics. A scaled down table follows:

Table 11 – FT/Orange testbed characterization and capacity

Testbed characterization and Capacity	
Connectivity	Secure remote user access
Availability	The testbed is shared among Orange projects and must be reserved in advance for use in MCN
Capacity	Fronthaul: up to 3 CPRI links at 2.5Gbit/s
Is SDN available on your cloud infrastructure?	The testbed is not a cloud infrastructure

A.4 CloudSigma

CloudSigma offers its commercial Infrastructure-as-a-Service (IaaS) platform to the Mobile Cloud Project, through its Zurich based Cloud Infrastructure. Whenever convenient, other CloudSigma Cloud Infrastructures may also be used (e.g. its Las Vegas infrastructure). The platform combines a proprietary stack with the use of state of the art open source technologies to provide a utility approach to IaaS provisioning. The platform offers high level of control and flexibility in the provision of computational power, RAM, storage (SSD as well as conventional magnetic), and networking. As a commercially proven public cloud provider, CloudSigma is in a position to share experience and know-how with other project partners. CloudSigma provides a Restful API to the full feature set of its cloud, to enable the automation and deployment of multi-tenant environments optimised for interoperability, scalability and energy efficiency, purposing the idea for the creation of a Cloud Controller.

A series of how-to videos demonstrating the various functionalities available via the CloudSigma Web App has been recorded and distributed among MCN partners. The subjects include the following:

- How to create a server
- How to create a new drive
- How to attach a drive
- How to start a server and connect to it using a VNC tunnel
- How to clone a drive image
- How to purchase a static IP address and attach it to a VM
- How to purchase a VLAN and attach it to a VM

Table 12 – CloudSigma testbed characterization and capacity

Testbed characterization and Capacity	
General testbed configuration	
Hypervisor	KVM
How is VM monitoring made possible on your IaaS stack?	Intra-VM testing tools, at the discretion of the VM user, NewRelic third party integration
Access methods (upload, download)	FTP (including Secure Shell, SSH File Transport Protocol, SFTP), API via Https
Connectivity	Internet, VPN, Secure Remote User Access, Direct private patch to local switch
Cloud Interface	
Provisioning	API, API middleware, Web console, Python

	library (Pycloudsigma)
API protocol / middleware	JClouds, Fog
Networking	API, Web console
IaaS	
What IaaS stack does your cloud employ?	Proprietary CloudSigma Stack
If OpenStack, what version will be made available for the testbeds?	N/A
If Open Stack, will OpenStack Heat be deployed on your cloud and made available to the testbeds?	N/A
PaaS	
What PaaS can be deployed and made available on the testbeds?	None of the above
What version is the PaaS?	N/A
Storage	
Available storage interfaces	Volume storage
Image format	RAW
SSD capacity (GB/per VM)	8TB per image, subject to availability. More than 1 image can be mounted per VM
HDD capacity (GB/per VM)	N/A
Combined storage capacity	
What are the maximum real-world IOPS attainable from your storage solution?	No limit, subject to cloud capacity and bearing in mind the drive limit of up to 8TB
Compute capacity	
CPU (GHz/core)	80
RAM (GB / VM)	128
Number of VMs	Subject to negotiation
Speed (Linux)	2s
Speed (Windows)	2s
Networking	
What switches do you employ within your cloud?	Vyatta OS based routers and Plexxi routers
What novel and relevant to MCN functionality do your switches offer users of your cloud?	A commercial grade bleeding edge implementation of SDN
Maximum internal network bandwidth made available per VM (in Gb)	20
Maximum external network bandwidth made available per VM (in Gb)	10
Maximum inter-VM latency in *ms*	1
Total cloud external network bandwidth (in Gb)	10+
Software Defined Networking	
Is SDN available on your cloud infrastructure?	Yes
What SDN protocols does your infrastructure support?	Plexxi Affinity networking
What SDN hardware switches does your infrastructure support?	QoS via Affinity networking.
What version of the SDN protocol does your infrastructure support?	Latest Plexxi protocol
Other relevant components/hardware	
Does your testbed provide other relevant components?	We facilitate the full use of the underlying CPU's instruction set, along with any and all optimisations this offers. NUMA, Single Instruction Multiple Data (SIMD), Streaming SIMD Extensions (SSE) etc. technologies are made available to the VMs. Any x86 compatible OS can be deployed on our infrastructure and full root rights are afforded to

	users. Compute, RAM and Storage are offered independently and are not bundled in any way. Drive images can be attached to more than one VM or none at all. The images are persistent, even when not attached to a VM. We do not place upper limits on internal and external data throughput.

A.5 NEC

The testbed information we have to date has not been updated since the original proposal. The following description is current as of the writing of this deliverable.

NEC has different OpenFlow-oriented test facilities, which will be used on-site for different tests. First the more test oriented development centre with two racks of servers and 2 OpenFlow switches as Top of the Rack switches a node reservation system storage. For demo purposes NEC will also provide an OpenFlow set-up with smaller scale hardware (but nevertheless able to emulate more networks through VLAN).

A.6 INTEL

Intel currently has a testbed of 17 compute nodes for validating research concepts. These are available for experimental cloud-related work. The systems are a collection of different hardware specifications, which are useful for testing various types of admission and capacity oriented configurations. The testbeds backbone network is currently 1Gbps Ethernet. The testbeds capacity is constantly under review with plans in the coming year to add two Storage Area Networks (SANs) and significantly grow the compute capacity.

Table 13 – INTEL characterization and capacity

Testbed characterization and Capacity	
General testbed configuration	
Hypervisor	KVM
How is VM monitoring made possible on your IaaS stack?	-
Access methods (upload, download)	API via Https
Connectivity	Secure remote user access
Cloud Interface	
Provisioning	API, Web console
API protocol / middleware	OpenStack API, OCCI, Cloud Data Management Interface (CDMI)
Networking	API
IaaS	
What IaaS stack does your cloud employ?	OpenStack
If OpenStack, what version will be made available for the testbeds?	Havana
If Open Stack, will OpenStack Heat be deployed on your cloud and made available to the testbeds?	Planned
PaaS	
What PaaS can be deployed and made available on the testbeds?	OpenShift

What version is the PaaS?	-
Storage	
Available storage interfaces	Block device storage, Volume storage
Image format	QCOW2
SSD capacity (GB/per VM)	10
HDD capacity (GB/per VM)	10
Combined storage capacity	500
What are the maximum real-world IOPS attainable from your storage solution?	N/A
Compute capacity	
CPU (GHz/core)	2.4
RAM (GB / VM)	16 max
Number of VMs	100
Speed (Linux)	240
Speed (Windows)	240
Networking	
What switches do you employ within your cloud?	-
What novel and relevant to MCN functionality do your switches offer users of your cloud?	-
Maximum internal network bandwidth made available per VM (in Gb)	-
Maximum external network bandwidth made available per VM (in Gb)	-
Maximum inter-VM latency in *ms*	-
Total cloud external network bandwidth (in Gb)	-
Software Defined Networking	
Is SDN available on your cloud infrastructure?	Yes
What SDN protocols does your infrastructure support?	None of the above
What SDN hardware switches does your infrastructure support?	-
What version of the SDN protocol does your infrastructure support?	-
Other relevant components/hardware	
Does your testbed provide other relevant components?	-

A.7 TI

TI provides an OpenStack based testbed with Cisco UCS Compute nodes connected through a Catalyst switch.

Telecom Italia currently provides the largest cloud-computing infrastructure by means of the brand “Nuvola Italiana”. Telecom Italy leverages on his strategic assets to ensure to the Market an end-to-end cloud services (IaaS, SaaS and PaaS) offering with high standard in terms of quality, reliability and security.

The Telecom Italia actual infrastructure (IaaS and PaaS) cloud offering is based on 8 Next Generation Data Centres enabled by shared and virtualised architectures, connected by high capacity broadband and dedicated networks.

Telecom Italia Lab (TILAB), located in Turin, provides the research activities in this domain. This department is a part of the Innovation and R&D Departments of Telecom Italia that involves around 1600 researchers and technicians.

TILAB runs multiple labs equipped with several IT and Network technologies, which mimic the commercial "Nuvola Italiana" platform. In order to demonstrate the use cases related to this project Telecom Italia will provide the following platforms and technologies:

- IT virtualised resources based on our shared infrastructure (VMware, XenServer).
- Networking:
    - Standard TCP/IP router and Ethernet switches
    - OpenFlow-capable switches
    - Network delay emulators
    - Virtual network appliances (load balancer, firewalls, ...)
- Storage Area Network (SAN), Internet Small Computer System Interface (iSCSI), Fibre Channel, Network File System (NFS).

Telecom Italia will provide remote access to other partners and the necessary operational support to contribute to the experimentation and evaluation phase.

Table 14 – TI testbed characterization and capacity

Testbed characterization and Capacity	
General testbed configuration	
Hypervisor	KVM
How is VM monitoring made possible on your IaaS stack?	Intra-VM testing tools, at the discretion of the VM user
Access methods (upload, download)	API via Https
Connectivity	Internet, VPN, Secure remote user access
Cloud Interface	
Provisioning	API
API protocol / middleware	OpenStack API
Networking	API
IaaS	
What IaaS stack does your cloud employ?	OpenStack
If OpenStack, what version will be made available for the testbeds?	Grizzly, Havana
If Open Stack, will OpenStack Heat be deployed on your cloud and made available to the testbeds?	Yes
PaaS	
What PaaS can be deployed and made available on the testbeds?	OpenShift
What version is the PaaS?	-
Storage	
Available storage interfaces	Block device storage, Volume storage, Object storage
Image format	QCOW2
SSD capacity (GB/per VM)	0
HDD capacity (GB/per VM)	20
Combined storage capacity	20
What are the maximum real-world IOPS attainable from your storage solution?	-
Compute capacity	
CPU (GHz/core)	1.8
RAM (GB / VM)	16
Number of VMs	50

Speed (Linux)	60
Speed (Windows)	60
Networking	
What switches do you employ within your cloud?	Cisco Catalyst 3560G
What novel and relevant to MCN functionality do your switches offer users of your cloud?	-
Maximum internal network bandwidth made available per VM (in Gb)	-
Maximum external network bandwidth made available per VM (in Gb)	-
Maximum inter-VM latency in *ms*	-
Total cloud external network bandwidth (in Gb)	-
Software Defined Networking	
Is SDN available on your cloud infrastructure?	Yes
What SDN protocols does your infrastructure support?	OpenFlow
What SDN hardware switches does your infrastructure support?	-
What version of the SDN protocol does your infrastructure support?	1
Other relevant components/hardware	
Does your testbed provide other relevant components?	-

A.8 UTWENTE

UTWENTE intends to provide an OpenStack virtualization testbed to implement a DMM architecture prototype based on SDN/OpenFlow technology. However, DMM is required for demonstration by month 21 and therefore not all the parameters have been finalised. The online form has been partially completed but we expect more input over the months following the delivery of D6.1. The spreadsheet will be manually updated as required.

Further updates indicate that UTWENTE:

- Had problems installing Folsom + Quantum so installed Grizzly + Quantum instead.
- Bought an OpenFlow enabled switch, Pronto P-3295 to install and work with OpenFlow. This had not been received at the time of delivering this report.
- Waiting for confirmation from FHG on whether the OpenEPC module needs to realise the DMM and service migration testbeds.
- Working on the implementation of a DMM solution on ns3 LENA. This simulation will also include an ns3 model of an OpenFlow enabled switch.
- Working on the implementation of an ICN based service migration solution on the ns3-dev-ndnsim simulation environment.

Table 15 – UTWENTE testbed characterization and capacity

Testbed characterization and Capacity	
General testbed configuration	
Hypervisor	OpenStack - OpenFlow Controller
How is VM monitoring made possible on your IaaS stack?	-
Access methods (upload, download)	Via Custom Virtualization Service

Connectivity	Internet VPN
Cloud Interface	
Provisioning	Web Console
API protocol / middleware	OpenStack API
Networking	Web console, custom software
IaaS	
What IaaS stack does your cloud employ?	OpenStack
If OpenStack, what version will be made available for the testbeds?	Grizzly, Havana
If Open Stack, will OpenStack Heat be deployed on your cloud and made available to the testbeds?	Planned
PaaS	
What PaaS can be deployed and made available on the testbeds?	TBD or unknown
What version is the PaaS?	TBD or unknown
Storage	
Available storage interfaces	Block storage
Image format	TBD or unknown
SSD capacity (GB/per VM)	TBD or unknown
HDD capacity (GB/per VM)	TBD or unknown
Combined storage capacity	TBD or unknown
What are the maximum real-world IOPS attainable from your storage solution?	TBD or unknown
Compute capacity	
CPU (GHz/core)	TBD or unknown
RAM (GB / VM)	TBD or unknown
Number of VMs	TBD or unknown
Speed (Linux)	-
Speed (Windows)	-
Networking	
What switches do you employ within your cloud?	TBD or unknown
What novel and relevant to MCN functionality do your switches offer users of your cloud?	TBD or unknown
Maximum internal network bandwidth made available per VM (in Gb)	TBD or unknown
Maximum external network bandwidth made available per VM (in Gb)	TBD or unknown
Maximum inter-VM latency in *ms*	TBD or unknown
Total cloud external network bandwidth (in Gb)	TBD or unknown
Software Defined Networking	
Is SDN available on your cloud infrastructure?	Yes
What SDN protocols does your infrastructure support?	OpenFlow
What SDN hardware switches does your infrastructure support?	TBD or unknown
What version of the SDN protocol does your infrastructure support?	TBD or unknown
Other relevant components/hardware	
Does your testbed provide other relevant components?	-

A.9 UBERN

Originally UBERN was going to provide a wireless testbed to emulate the RAN. Due to changes in the approach in T3.5 this is no longer the case. UBERN currently has a testbed made available at the UBERN lab for the purpose of testing a VM with OpenAirInterface. More VMs will be set up on CloudSigma's infrastructure. The VMs will be used to run an LTE radio part emulator.

Table 16 – UBERN testbed characterization and capacity

Testbed characterization and Capacity	
General testbed configuration	
Hypervisor	XEN
How is VM monitoring made possible on your IaaS stack?	Hypervisor hooks per VM Hypervisor VM load logs or related functionality
Access methods (upload, download)	API via Https, Peer-to-Peer (P2P), Secure Copy Protocol (SCP)
Connectivity	Internet, VPN
Cloud Interface	
Provisioning	Web console
API protocol / middleware	No cloud
Networking	-
IaaS	
What IaaS stack does your cloud employ?	It is not IaaS
If OpenStack, what version will be made available for the testbeds?	-
If Open Stack, will OpenStack Heat be deployed on your cloud and made available to the testbeds?	No
PaaS	
What PaaS can be deployed and made available on the testbeds?	OpenAirInterface
What version is the PaaS?	-
Storage	
Available storage interfaces	Volume storage
Image format	TBD
SSD capacity (GB/per VM)	0
HDD capacity (GB/per VM)	32
Combined storage capacity	1000
What are the maximum real-world IOPS attainable from your storage solution?	-
Compute capacity	
CPU (GHz/core)	3.33
RAM (GB / VM)	4
Number of VMs	1
Speed (Linux)	-
Speed (Windows)	-
Networking	
What switches do you employ within your cloud?	-
What novel and relevant to MCN functionality do your switches offer users of your cloud?	-
Maximum internal network bandwidth made available per VM (in Gb)	-
Maximum external network bandwidth made available per VM (in Gb)	-
Maximum inter-VM latency in *ms*	-
Total cloud external network bandwidth (in Gb)	-
Software Defined Networking	
Is SDN available on your cloud infrastructure?	No
What SDN protocols does your infrastructure support?	None of the above
What SDN hardware switches does your	-

infrastructure support?	
What version of the SDN protocol does your infrastructure support?	-
Other relevant components/hardware	
Does your testbed provide other relevant components?	-

A.10 BT

BT will be no longer providing a testbed due to them exiting the project.

A.11 ZHAW

Within the InIT Cloud Computing Laboratory runs the "ICCLab project". It is the foundational seed project of our research lab. It designs, deploys, and operates the ICCLab cloud computing hard and software infrastructure and addresses important research aspects around automation and interoperability of cloud computing infrastructure and frameworks.

The framework of choice is OpenStack for IaaS, which enjoys significant industry and academic support and is reaching good levels of maturity. The lab will support pre-production usage scenarios on top of OpenStack services as well as experimental research on OpenStack technology and extensions. Currently the actively deployed OpenStack services are the OpenStack compute service (including keystone, glance and nova), and Swift, an object storage service. From the PaaS perspective, the framework of choice is CloudFoundry.

The lab is equipped with 20 Commercial of the shelf (COTS) computing units, each running on 8×2.4 GHz Cores, 64GB RAM and 4×1TB local storage per unit. To store templates and other data we run an additional 12TB NFS or iSCSI Storage that is connected to a switch with a 10Gbit Ethernet interface. The Computing Units are connected to a 1Gbit network for data and another 1Gbit net for control traffic. At the heart of the ICCLab is the Management Server, which provides an easy way to stage different setups for different OpenStack instances (productive, experimental, etc.). The network is a mix of both traditional and contemporary networking hardware. For traditional networking HP ProCurve switches and for research related to SDN, Pica8 OpenFlow (v1.2) switches are used. The Management Server provides a Dynamic Host Configuration Protocol (DHCP), Pre-Execution Environment (PXE) service through Foreman all of which allow a bare metal computing unit to be provisioned automatically, using Foreman, and then have pre-assigned roles installed, using a combination of Foreman and Puppet. This provides a great deal of flexibility and support for different usage scenarios.

Table 17 – ZHAW testbed characterization and capacity

Testbed characterization and Capacity	
General testbed configuration	
Hypervisor	KVM
How is VM monitoring made possible on your IaaS stack?	NewRelic
Access methods (upload, download)	FTP (including SFTP), API via Https
Connectivity	Internet
Cloud Interface	
Provisioning	API, CLI, Web console
API protocol / middleware	OpenStack API, OCCI
Networking	OpenStack Web console

IaaS	
What IaaS stack does your cloud employ?	OpenStack
If OpenStack, what version will be made available for the testbeds?	Havana
If Open Stack, will OpenStack Heat be deployed on your cloud and made available to the testbeds?	Yes
PaaS	
What PaaS can be deployed and made available on the testbeds?	OpenShift, CloudFoundry
What version is the PaaS?	-
Storage	
Available storage interfaces	Block storage, Volume storage
Image format	QCOW, QCOW2, RAW
SSD capacity (GB/per VM)	-
HDD capacity (GB/per VM)	-
Combined storage capacity	-
What are the maximum real-world IOPS attainable from your storage solution?	-
Compute capacity	
CPU (GHz/core)	-
RAM (GB / VM)	-
Number of VMs	-
Speed (Linux)	-
Speed (Windows)	-
Networking	
What switches do you employ within your cloud?	pica8
What novel and relevant to MCN functionality do your switches offer users of your cloud?	-
Maximum internal network bandwidth made available per VM (in Gb)	-
Maximum external network bandwidth made available per VM (in Gb)	-
Maximum inter-VM latency in *ms*	-
Total cloud external network bandwidth (in Gb)	-
Software Defined Networking	
Is SDN available on your cloud infrastructure?	SDN
What SDN protocols does your infrastructure support?	OpenFlow
What SDN hardware switches does your infrastructure support?	-
What version of the SDN protocol does your infrastructure support?	-
Other relevant components/hardware	
Does your testbed provide other relevant components?	Ceilometer access (API)

A.12 ONE

OneSource will provide a simple testbed with running OpenStack installation (Havana), suitable for small experiments, using Open vSwitch and Neutron for networking. This will support the development, integration and functional testing of the MobileCloud platform, composed of 6 computing units which can be decomposed in multiple virtual machines to emulate typical topologies of the MobileCloud platform. Additionally, for larger tests (during limited periods of time), OneSource may also provide access to the TITAN Cluster, a shared HPC facility composed of 30 real machines (currently configured to run 80 virtual machines) and adequate management, storage and networking resources. OneSource may also provide networking support services, such as a VPN service, in order to connect its testbed with other MobileCloud testbeds to support the integration and testing of different MobileCloud components.

Table 18 – ONE's testbed characterization and capacity

Testbed characterization and Capacity	
General testbed configuration	
Hypervisor	KVM
How is VM monitoring made possible on your IaaS stack?	Intra-VM testing tools, at the discretion of the VM user
Access methods (upload, download)	API via Https
Connectivity	Internet, VPN, Secure Remote User Access
Cloud Interface	
Provisioning	API
API protocol / middleware	OpenStack API
Networking	API
IaaS	
What IaaS stack does your cloud employ?	OpenStack
If OpenStack, what version will be made available for the testbeds?	Havana
If Open Stack, will OpenStack Heat be deployed on your cloud and made available to the testbeds?	yes
PaaS	
What PaaS can be deployed and made available on the testbeds?	OpenShift
What version is the PaaS?	latest
Storage	
Available storage interfaces	Block device storage, Volume storage
Image format	QCOW2
SSD capacity (GB/per VM)	-
HDD capacity (GB/per VM)	Variable (max. 2Tb)
Combined storage capacity	2Tb
What are the maximum real-world IOPS attainable from your storage solution?	-
Compute capacity	
CPU (GHz/core)	2,27
RAM (GB / VM)	Variable (max. 8Gb)
Number of VMs	Variable
Speed (Linux)	-
Speed (Windows)	-
Networking	
What switches do you employ within your cloud?	Virtual switch (Open vSwitch)
What novel and relevant to MCN functionality do your switches offer users of your cloud?	-
Maximum internal network bandwidth made available per VM (in Gb)	0,1
Maximum external network bandwidth made available per VM (in Gb)	1
Maximum inter-VM latency in *ms*	1
Total cloud external network bandwidth (in Gb)	1
Software Defined Networking	
Is SDN available on your cloud infrastructure?	Yes
What SDN protocols does your infrastructure support?	OpenFlow

What SDN hardware switches does your infrastructure support?	Open vSwitch (virtually supports anything)
What version of the SDN protocol does your infrastructure support?	1.1
Other relevant components/hardware	
Does your testbed provide other relevant components?	

A.13 STT

The DSS testbed is for internal use, to have an instance of DSS deployed using the Cloud Controller through the Service Manager and Service Orchestrator with simple functionality, and performing unit testing of DSS service components. The main features of this testbed are presented on the following table:

Table 19 – DSS testbed characterization and capacity

Testbed characterization and Capacity	
General testbed configuration	
Hypervisor	Hypervisor type 2. VirtualBox
How is VM monitoring made possible on your IaaS stack?	-
Access methods (upload, download)	API via Https
Connectivity	Internet, VPN
Cloud Interface	
Provisioning	API, CLI, Web console
API protocol / middleware	MCN Cloud Controller API
Networking	-
IaaS	
What IaaS stack does your cloud employ?	OpenStack
If OpenStack, what version will be made available for the testbeds?	Havana
If Open Stack, will OpenStack Heat be deployed on your cloud and made available to the testbeds?	Planned
PaaS	
What PaaS can be deployed and made available on the testbeds?	OpenShift
What version is the PaaS?	-
Storage	
Available storage interfaces	-
Image format	-
SSD capacity (GB/per VM)	-
HDD capacity (GB/per VM)	-
Combined storage capacity	-
What are the maximum real-world IOPS attainable from your storage solution?	-
Compute capacity	
CPU (GHz/core)	-
RAM (GB / VM)	-
Number of VMs	-
Speed (Linux)	-
Speed (Windows)	-
Networking	
What switches do you employ within your cloud?	-
What novel and relevant to MCN functionality do	-

your switches offer users of your cloud?	
Maximum internal network bandwidth made available per VM (in Gb)	-
Maximum external network bandwidth made available per VM (in Gb)	-
Maximum inter-VM latency in *ms*	-
Total cloud external network bandwidth (in Gb)	-
Software Defined Networking	
Is SDN available on your cloud infrastructure?	-
What SDN protocols does your infrastructure support?	-
What SDN hardware switches does your infrastructure support?	-
What version of the SDN protocol does your infrastructure support?	-
Other relevant components/hardware	
Does your testbed provide other relevant components?	-

B Appendix B: Detailed Evaluation Plans

B.1 DSS

The DSS Service is being led by STT, which conducts its development, integration and assessment activities. The following subsections follow the previously presented evaluation guidelines adapted for this service.

B.1.1 Requirements and Strategic Evaluation Objectives

Table 20 – Requirements and Evaluation Objectives

Nº	Requirement	Strategic Evaluation Objective
1	On-Demand Service Instantiation	DSS can be instantiated on demand, according to EEU business needs
2	Automated Service Scaling	DSS can be elastic scaled according to the load of the system to provide a reliable service
3	Flexibility & Dynamic Contents	DSS is able to show and modify in time different types of information simultaneously
4	Service Performance	Improvement based on CDN integration and RAN/EPC architecture
		DSS Central Management Server should be fast internal managed

B.1.2 Definition of Evaluation Areas, Impacts and Selection of Indicators

Table 21 – Evaluation Area, Impact and Indicators definition

Strategic Evaluation Objective	Evaluation Area	Impact	Nº	Indicator's name
DSS can be instantiated on demand, according to EEU business needs	IaaS/MCN Service	Instance Works Properly	1	Works Properly
		Time elapsed for complete Instantiation	2	Instantiation Time
		Number of Instances (VMs)	3	Number Instances
DSS can be elastic scaled according to the load of the system to provide a reliable service	IaaS/MCN Service	Elastic Scaling time and number of instances	4	Elastic Scaling Availability
DSS is able to show and modify in time different types of	IaaS/MCN Service	Flexibility and Dynamic Contents of	5	Flexibility: Dynamic Contents & Different content types

information simultaneously		different types		
Improvement based on CDN integration and RAN/EPC architecture	IaaS/MCN Service	Faster Content loading/downloading	6	Loading/downloading Time
DSS Central Management Server should be fast internal managed	IaaS/MCN Service	Response Time	7	Response Time

B.1.3 Definition of success and targets

Table 22 – Impacts Target

Nº	Indicator's name	Target quantification
1	Works Properly	Yes
2	Instantiation Time	Few Minutes (< ~10 min.)
3	Number Instances	As Defined/Expected (1, 2, 3, etc.)
4	Elastic Scaling Availability	Auto-scaling Availability
5	Flexibility: Dynamic Contents & Different content types	Automatic content change and flexible HTML5 layouts
6	Faster content loading/downloading	20% loading and downloading time reduction
7	Response Time	< 4s

B.1.4 Definition of evaluation technique

Table 23 – Indicators assessment

Nº	Indicator's name	Indicator's description	Data/Unit	Data Source	Method of Measurement	Frequency
1	Works Properly	Indicates whether the service works as expected	Boolean (OK, NOK), qualitative	-	Manual Observation	On every deployment change
2	Instantiation Time	Time elapsed until full service instantiation	Seconds (s), quantitative	-	Manual/Script Measurement	On every new service Instantiation
3	Number Instances/Tier	Number of instances per tier	#, Quantitative	-	Manual Observation	On every deployment change

						(scaling)
4	Elastic Scaling Availability	Number of present instances on the system	#, Quantitative	Datasheet	Manual Observation	Per proof
5	Flexibility: Dynamic Contents & Different content types	Validate the different kinds of files are showed on the players	#, Quantitative	Datasheet	Manual Observation	Per proof
6	Faster content loading/downloading	Elapsed time from start and finish loads or downloads	Millisecond (ms), quantitative	Datasheet	Script Measurement	Each time a load or download is done
7	Response Time	Elapsed time to receive a response from Central Management Server (CMS)	Seconds (s), quantitative	Datasheet	Script Measurement	On Every Request

B.2 EPC - PGW, SGW, MME, HSS

The EPC Service is being led by FHG, which conducts its development, integration and assessment activities. The following subsections follow the previously presented evaluation guidelines adapted for this service.

B.2.1 Requirements and Strategic Evaluation Objectives

Table 24 – Requirements and Evaluation Objectives

Nº	Requirement	Strategic Evaluation Objective
1	On-Demand Service Instantiation	The service should be deployed on demand with at least two different software architectures
2	On-Demand/Automated Service Scaling	The data path components and the HSS will scale on demand/automated based on policies. Depending on the integration with OAI, also control plane entities will be scaling. Different scaling algorithms will be tested.
3	Seamless Scaling	Support of seamless data path scaling. The end user should not perceive any disruption/downtime.
4	Efficient Scaling	The scaling procedures will be evaluated not to consume a

		large amount of resources
5	Multi-tenancy	Running multiple parallel EPC infrastructures of different tenants
6	Benchmarking Application Performance	Providing a relevant set of measurements on the capacity and on the capabilities of a virtualised EPC (with different architectures, with different user traces, with different eNB placements, with different services on the data path). It includes: • Performance – resources consumed/100 users • Quality – delay of operations perceived at eNB • Robustness – recovering from a component sudden shut-down It includes support for: attachment/detachment, active handover procedure and generic IP data traffic exchange.
7	RAN-EPC performance	Providing a relevant set of measurements for the RAN-EPC integrated system specifically in relationship with the quality of the EPS service.

B.2.2 Definition of Evaluation Areas, Impacts and Selection of Indicators

Table 25 – Evaluation Area, Impact and Indicators definition

Strategic Evaluation Objective	Evaluation Area	Impact	Nº	Indicator's name
On-Demand Service Instantiation	IaaS/MCN Service	Instance Works Properly	1	Works Properly
		Time elapsed for complete Instantiation	2	Instantiation Time
On-Demand/Automated Service Scaling	IaaS/MCN Service	Instance Works Properly	1	Works Properly
		Time elapsed for complete Instantiation	2	Instantiation Time
		Time elapsed for scale-in	4	Instance starting
		Time elapsed for scale-down	3	Instance removal time

Seamless Scaling	MCN Service	Delay/Packet Loss/Bandwidth fluctuation	5	QoS measurement at end-device side
Efficient Scaling	IaaS/MCN Service	Overhead produced by scaling	6	Compute/Storage/Network resources consumed
Multi-tenancy	IaaS/MCN Service	Number of Tenants instantiated	7	Number of Tenants
		Multiple Isolated Tenants	8	Tenant Isolation
Benchmarking – Application performance	IaaS&MCN service / MCN service	Performance	12	Infrastructure resources consumed
		Quality	10	Requests/second
			11	Response time
			5	Simplified QoS
		Robustness	9	High availability at data path level
RAN-EPC performance	IaaS/MCN service/MCN service	Quality	10	Requests/second
			11	Response time
			5	Simplified QoS

B.2.3 Definition of Success and Targets

Table 26 – Impacts Target

Nº	Indicator's name	Target quantification
1	Works Properly	Yes ~~(No / Partially).~~ Functional test executed (attachment/detachment/handover/default bearer data traffic)
2	Instantiation Time	Few Minutes (< ~10 min.) – split into EPC and platform related procedures
3	Scale-Out Time	Few Minutes (< ~10 min.) – split into EPC and platform related procedures
4	Scaling-In Time	Few Minutes (< ~10 min.) – split into EPC and platform related procedures

5	Seamless End-to-end service quality	Packet loss/Delay/QoS as perceived at end-devices – Minimal changes
6	Performance Overhead	Compute/Storage/Network overhead created by scaling
7	Number of Tenants	As Defined/Expected (1, 2, 3)
8	Tenant Isolation	No Impact Noticeable (see 12, 13)
9	Service Availability	Downtime < 99.999%
10	Requests per Second	[~10, ~1000]
11	Response Time	< 200 ms
12	Performance	IaaS measured Compute/Storage/Network resources consumed

B.2.4 Definition of Evaluation Technique

Table 27 – Indicators assessment

Nº	Indicator's name	Indicator's description	Data/Unit	Data Source	Method of Measurement	Frequency
1	Works Properly	Functional	Yes/No	-	Functional test of basic procedures	Every deployment
2	Instantiation Time	Procedure Duration	Time	monitoring	Script measurement	Some deployments
3	Scale-Out Time	Procedure Duration	Time	monitoring	Script measurement	Elastic scaling deployments
4	Scaling-In Time	Procedure Duration	Time	monitoring	Script measurement	Elastic scaling deployments
5	Seamless End-to-end service quality	QoS parameters	Time, Bandwidth Packet Loss	Benchmarking tool	Benchmarking	Benchmark
6	Performance Overhead	Infrastructure Parameters (DELTA)	Compute/Storage/Network	monitoring	Benchmarking	Benchmark
7	Number of Tenants	Functional	Number	-	Functional test of basic procedures	Selected deployments
8	Tenant	Functional	Yes/No	-	Functional test of	Selected

	Isolation				basic procedures	deployments
9	Service Availability	Downtime	Time	-	Computation based on the scale-out and of the seamless end-to-end service quality parameters	Computation
10	Requests per Second	[~100, ~1000]	Number	monitoring	Benchmarking tool	Benchmark
11	Response Time	Procedure delay in load conditions	Time	Benchmarking tool	Benchmarking tool	Benchmark
12	Performance	Infrastructure Parameters	Compute/Storage/Network	monitoring	Benchmarking tool	Benchmark

B.3 EPC - ANDSF

The ANDSF Service is being led by PTInS, which conducts its development, integration and assessment activities. The following subsections follow the previously presented evaluation guidelines adapted for this service.

B.3.1 Requirements and Strategic Evaluation Objectives

Table 28 – Requirements and Evaluation Objectives

Nº	Requirement	Strategic Evaluation Objective
1	On-Demand Instantiation	Support the on-demand ANDSF service instantiation. The customer can request an ANDSF service instantiation, through a web portal.
2	Automated Scaling	Support the automated ANDSF service (horizontal) scaling (in/out) based on information such as monitoring data (MaaS), prediction (MOBaaS), etc.
3	Seamless Scaling	Support of seamless ANDSF service scaling. The ANDSF user should not perceive any disruption/downtime.
4	Multi-tenancy	Support of multiple independent ANDSF tenants. Each tenant is assigned with an isolated ANDSF service environment.
5	monitoring	Support ANDSF Service monitoring, including both infrastructural and service level indicators.
6	Charging	Support ANDSF Service charging, based on global ANDSF relevant indicators, using multiple business models.
7	High Availability	Assure carrier-grade availability levels (e.g. 5 9's), even when

		regular scaling operations are considered.
8	Performance	The ANDSF service should support requests in a range of tens to thousands per second, assuring service response times under a pre-defined threshold.

B.3.2 Definition of Evaluation Areas, Impacts and Selection of Indicators

Table 29 – Evaluation Area, Impact and Indicators definition

Strategic Evaluation Objective	Evaluation Area	Impact	Nº	Indicator's name
Support the on-demand ANDSF service instantiation. The customer can request an ANDSF service instantiation, through a web portal.	IaaS/MCN Service	Instance Works Properly	1	Works Properly
		Time elapsed for complete Instantiation	2	Instantiation Time
		Number of Instances per tier	3	Number Instances/Tier
Support the automated ANDSF service (horizontal) scaling (in/out) based on information such as monitoring data (MaaS), prediction (MOBaaS), etc.	IaaS/MCN Service	Instance Works Properly	1	Works Properly
		Time elapsed for complete Scaling	4	Scaling Time
		Number of Instances per tier	3	Number Instances/Tier
Support of seamless ANDSF service scaling. The ANDSF user should not perceive any disruption/downtime.	MCN Service	Downtime during a Scaling operation	5	Scaling Downtime
Support of multiple independent ANDSF tenants. Each tenant	IaaS/MCN Service	Number of Tenants instantiated	6	Number of Tenants

is assigned with an isolated ANDSF service environment.		Multiple Isolated Tenants	7	Tenant Isolation
Support ANDSF Service monitoring, including both infrastructural and service level indicators.	IaaS/MCN Service	Availability of monitoring infrastructure metrics	8	Infrastructure Metrics
		Availability of monitoring ANDSF service metrics	9	ANDSF Service Metrics
Support ANDSF Service charging, based on global ANDSF relevant indicators, using multiple business models.	MCN Service	Availability of Charging capabilities	10	Charging Capabilities
Assure carrier-grade availability levels (e.g. 5 9's), even when regular scaling operations are considered.	MCN Service	Carrier-grade Availability	11	Service Availability
The ANDSF service should support requests in a range of tens to thousands per second, assuring service response times under a pre-defined threshold.	MCN Service	Number of requests per second	12	Requests per Second
		Response Time	13	Response Time

B.3.3 Definition of Success and Targets

Table 30 – Impacts Target

Nº	Indicator's name	Target quantification
1	Works Properly	Yes ~~(No / Partially)~~
2	Instantiation Time	Few Minutes (< ~10 min.)
3	Number Instances/Tier	As Defined/Expected (1, 2, 3, etc.)

4	Scaling Time	Few Minutes (< ~5 min.)
5	Scaling Downtime	Few Milliseconds (< ~100 ms)
6	Number of Tenants	As Defined/Expected (1, 2, 3)
7	Tenant Isolation	No Impact Noticeable (see 12, 13)
8	Infrastructure Metrics	Yes (Infrastructure Metrics Available)
9	ANDSF Service Metrics	Yes (ANDSF Service Metrics Available)
10	Charging Capabilities	Yes (Charging Models Can Be Applied)
11	Service Availability	Downtime < 99.999%
12	Requests per Second	[~10, ~1000]
13	Response Time	< 200 ms

B.3.4 Definition of Evaluation Technique

Table 31 – Indicators assessment

Nº	Indicator's name	Indicator's description	Data/Unit	Data Source	Method of Measurement	Frequency
1	Works Properly	Indicates whether the service works as expected	Boolean (yes, no), qualitative	-	Manual Observation	On every deployment change
2	Instantiation Time	Time elapsed until full service instantiation	Seconds (s), quantitative	-	Manual/Script Measurement	On every new service Instantiation
3	Number Instances/Tier	Number of instances per tier	#, Quantitative	-	Manual Observation (OpenStack)	On every deployment change (scaling)
4	Scaling Time	Time elapsed until full scaling	Seconds (s), quantitative	-	Script Measurement	On every scaling
5	Scaling Downtime	Downtime (outage) period during the scaling process	Millisecond (ms) or seconds (s), quantitative	-	Script Measurement	On every scaling
6	Number of Tenants	Number of active tenants	#, Quantitative	-	Manual/Script Measurement	Anytime needed

7	Tenant Isolation	Whether tenants are isolated	Boolean (yes, no), qualitative	-	Indirect Observation (see 12, 13)	Anytime needed
8	Infrastructure Metrics	Infrastructure metrics (CPU, memory, etc.) are available	Boolean (yes, no), qualitative	-	Manual Observation	Anytime needed
9	ANDSF Service Metrics	ANDSF service metrics (# reqs, etc.) are available	Boolean (yes, no), qualitative	-	Manual Observation	Anytime needed
10	Charging Capabilities	The ANDSF service can be charged	Boolean (yes, no), qualitative	-	Manual Observation	Anytime needed
11	Service Availability	Service is up and running	Boolean (yes, no), qualitative	-	Script Measurement	Every second (s)
12	Requests per Second	Number of requests, in average, per second	Req./second (#/s), quantitative	-	Script Measurement	Every N seconds (s),
13	Response Time	Time elapsed until the client obtains a valid response	Millisecond (ms), quantitative	-	Script Measurement	Every testing request (N second (s))

B.4 IMS

The IMS Service is being led by TUB, which conducts its development, integration and assessment activities. The following subsections follow the previously presented evaluation guidelines adapted for this service.

B.4.1 Requirements and Strategic Evaluation Objectives

Table 32 – Requirements and Evaluation Objectives

Nº	Requirement	Strategic Evaluation Objective
1	Energy efficiency	Energy consumption reduction
2	Cost Efficiency	CAPEX & OPEX cost reduction
3	Time efficiency	Faster availability
4	On-Demand Instantiation	Instantiation on demand, based on EEU business requests
5	Automated Scaling	Automated scaling without any downtime

6	Support pay as you go service	Charging based on usage
7	Multi-tenancy	Support of multiple independent tenants. Each tenant is assigned with an isolated service environment
8	Service continuity	Provide the service also in case of failures, scaling or upgrade
9	Compatibility with Legacy system supporting a smooth migration	Integration with legacy systems as well as green field architecture
10	Exploit MCN framework	Integration with other MCN services and support services
11	Application Performance	Support requests in a specific range per second, assuring service response times under a pre-defined threshold.

B.4.2 Definition of Evaluation Areas, Impacts and Selection of Indicators

Table 33 – Evaluation Area, Impact and Indicators definition

Strategic Evaluation Objective	Evaluation Area	Impact	Nº	Indicator's name
Energy consumption reduction	Energy	Energy consumption	1	Energy consumption
CAPEX & OPEX cost reduction	Economics	Capital Expenditures (CAPEX)	2	CAPEX
		Operational Expenses (OPEX)	3	OPEX
Faster availability	IaaS/MCN service Architecture	Deployment time	4	Deployment time
		Provisioning time	5	Provisioning time
		Scaling time	6	Scaling time
Instantiation on demand, based on EEU business requests	IaaS/MCN service Architecture	Deployment time	4	Deployment time
		Provisioning time	5	Provisioning time
Automated scaling without any downtime	IaaS/MCN Service	Scaling time	6	Scaling time
Charging based on usage	IaaS/MCN Service	RCBaaS integration	10	RCB integration
Support of multiple	IaaS/MCN Service	Number of	7	Number of tenants

independent tenants. Each tenant is assigned with an isolated service environment		tenants		
Provide the service also in case of failures, scaling or upgrade	IaaS/MCN Service	Service uptime	8	Service uptime
Integration with legacy systems as well as MCN systems	MCN Service	Charging legacy support	9	Legacy charging system
		RCBaaS support	10	RCB integration
Integration with other MCN services and support services	MCN Service	EPCaaS integration	11	EPC integration
		DBaaS integration	12	DB integration
		MaaS integration	13	monitoring integration
		RCBaaS integration	10	RCB integration
Support requests in a specific range per second, assuring service response times under a pre-defined threshold.	IaaS/MCN Service	Number of requests per second	14	Requests per second
		Response Time	15	Response Time

B.4.3 Definition of Success and Targets

Table 34 – Impacts Target

Nº	Indicator's name	Target quantification
1	Energy consumption	> 30% physical infrastructure energy consumption reduction
2	CAPEX	> 50% CAPEX reduction
3	OPEX	> 60% OPEX reduction
4	Deployment time	> 80% deployment time reduction
5	Provisioning time	> 50% deployment time reduction
6	Scaling time	< 5 mins

7	Number of tenants	> 10
8	Service uptime	> 99,99%
9	Legacy charging system	> 3
10	RCB integration	Yes
11	EPC integration	Yes
12	DB integration	Yes (HSS improvement)
13	monitoring integration	Yes
14	Requests per second	> 20 cps (for a small instance type)
15	Response Time	< 500 ms

B.4.4 Definition of Evaluation Technique

Table 35 – Indicators assessment

Nº	Indicator's name	Indicator's description	Data/Unit	Data Source	Method of Measurement	Frequency
1	Energy consumption		KW	Data sheet	At the beginning At the end	1 hour
2	CAPEX		€	Data sheet	Simulation	NA
3	OPEX		€	Data sheet	Simulation	NA
4	Deployment Time		sec	Data sheet	On Request Creation On end of creation process	On Every request for deployment (multiple runs)
5	Provisioning time		sec	Data sheet	On request for configuration On end of configuration process	On every request for configuration (multiple runs)
6	Scaling time		sec	Data sheet	At the beginning At the end	Per experiment
7	Number of Tenants		Number	Data sheet	Verification of distinction and partition of data and configuration	Per experiment

8	Service Uptime		Percentage	Data sheet	Percentage of time the service was working during the experiment	Per experiment
9	Legacy charging system		Number	Code execution	Interworking	Per experiment
10	RCB integration		Yes/No	Code execution	Interworking	Per experiment
12	EPC integration		Yes/No	Code execution	Interworking	Per experiment
12	DB integration		Yes/No	Code execution	Interworking	Per experiment
13	monitoring integration		Yes/No	Code execution	Interworking	Per experiment
14	Requests per second		Num/sec	Data sheet	Number of requested managed per second	Per experiment
15	Response Time		ms	Data sheet	Elapsed time to receive a response from service	On Every Request

B.5 RAN

The RAN Service is being led by Orange, which conducts its development, integration and assessment activities. The following subsections follow the previously presented evaluation guidelines adapted for this service.

B.5.1 Requirements and Strategic Evaluation Objectives

Table 36 – Requirements and Evaluation Objectives

Nº	Requirement	Strategic Evaluation Objective
1	On-Demand Service Instantiation	Support service creation on-demand of RAN base station connecting User Equipment to EPC
2	On-Demand Service Scaling	Support the on-demand per-tenant scaling in and out, up and down
3	Automated Service Scaling	Support the per-tenant automated scaling in and out, up and down, as requested by the Service Orchestrator
4	Migration	Enable the migration of base station with service continuity requirements from SLA

5	monitoring	Support RAN Service monitoring, including both infrastructural and service level indicators
6	Multi-tenancy	Support of multiple mobile network operators
7	Resiliency	RANaaS shall automatically recreate failed base stations.
8	Charging	Support RANaaS Service charging, based on relevant indicators, using multiple business models.

B.5.2 Definition of Evaluation Areas, Impacts and Selection of Indicators

Table 37 – Evaluation Area, Impact and Indicators definition

Strategic Evaluation Objective	Evaluation Area	Impact	Nº	Indicator's name
Support service creation on-demand of RAN base station connecting User Equipment to EPC	IaaS/MCN service	Time for providing resources to the base station	1	Deployment time
		Time for configuring the base station	2	Provisioning time
		Ability for eNB to connect to the service platforms	3	S1-U latency
		Ability for eNB to connect to the MME	4	S1-AP latency
Support the on-demand per-tenant scaling in and out, up and down	IaaS/MCN service	Time for the base station to be scaled	5	Scaling time
		Ability for eNB to connect to the service platforms	3	S1-U latency
		Ability for eNB to connect to the MME	4	S1-AP latency
Automated Service Scaling	IaaS/MCN service	Time for the base station to be scaled	5	Scaling time
		Ability for eNB to connect to the service platforms	3	S1-U latency
		Ability for eNB to connect to the	4	S1-AP latency

		MME		
Enable the migration of base station with service continuity requirements from SLA	IaaS/MCN service	Downtime during migration	6	Service downtime time
		Migration time for deploying the base station	1	Deployment time
		Migration time for configuring the base station	2	Provisioning time
Collection of application metrics	MCN service	Number of attached UE	7	Attached UE
		Volume of traffic per operator	8	Traffic volume
		Ability for eNB to connect to the service platforms	3	S1-U latency
		Ability for eNB to connect to the MME	4	S1-AP latency
Collection of infrastructure metrics	IaaS/MaaS	Usage of CPU for one RANaaS instance	9	CPU usage
Support of multiple independent operators. Each operator is assigned with an isolated RAN service environment.	IaaS/MCN Service	Number of Operators instantiated	10	Number of Operators
RANaaS shall automatically determine erroneous states and recreate failed base stations.	IaaS/MCN Service	Base station virtual machine is off	11	VM outage duration
		Base station software core is off	12	Service outage duration
		Base station software core is in erroneous state (malfunctioning)	13	SLA malfunctioning duration
		Time for providing resources to the	1	Deployment time

		base station		
		Time for configuring the base station	2	Provisioning time
		Ability for eNB to connect to the service platforms	3	S1-U latency
		Ability for eNB to connect to the MME	4	S1-AP latency
Support RANaaS Service charging, based on relevant indicators, using multiple business models.	RCBaaS/MCN Service	Ability to connect to RCB	15	RCB connection

B.5.3 Definition of Success and Targets

Table 38 – Impacts Target

Nº	Indicator's name	Target quantification
1	Deployment time	Few Minutes (< ~10 min.)
2	Provisioning time	Few Minutes (< ~5 min.)
3	S1-U latency	Few Milliseconds (< ~100 ms)
4	S1-AP latency	Few Milliseconds (< ~100 ms)
5	Scaling time	Few Minutes (< ~5 min.)
6	Service downtime	Few Milliseconds (< ~100 ms)
7	Attached UE	As Defined/Expected (1, 2, 3)
8	Traffic volume	Yes (traffic volume Available)
9	CPU usage	Yes (traffic volume Available)
10	Number of Operators	As Defined/Expected (1, 2, 3)
11	VM outage duration	Uptime > 99%
12	Service outage duration	Uptime > 99%
13	SLA malfunctioning duration	Uptime > 99%
14	RCB connection	Yes (Charging Models Can Be Applied)

B.5.4 Definition of Evaluation Technique

Table 39 – Indicators assessment

Nº	Indicator's name	Indicator's description	Data/Unit	Data Source	Method of Measurement	Frequency
1	Deployment time	Time for instantiating the base station	sec	Datasheet	On Request Creation On end of creation process	On Every request for deployment (multiple runs)
2	Provisioning time	Time for it activating the configuration of the base station	sec	Datasheet	On request for configuration On end of configuration process	On every request for configuration (multiple runs)
3	S1-U latency	Mean time of packet between the base station and the core network GW	sec	Datasheet	Periodic measurement	Every X seconds (e.g., 10)
4	S1-AP latency	Mean time of packet between the base station and the core network MME	sec	Datasheet	Periodic measurement	Every X seconds (e.g., 10)
5	Scaling time	Time for executing the scaling decision	sec	Datasheet	At the beginning At the end	Every minute (s)
6	Service downtime	Time when the base station does not handle UE' traffic	sec	Datasheet	At each migration	Every minute (s)
7	Attached UE	UE shall be able to attach to the base station	Number	Datasheet	Per UE RRC request Per UE RRC connection closing	Every minute (s)
8	Traffic	UEs shall be able to	Number	Datasheet	At each default bearer	Every

	volume	send/receive traffic			set up At each default bearer closing	minute (s)
9	CPU usage	CPU overhead due to the base stations running	Percentage of CPU usage	Datasheet	Average of CPU usage	Every 5m
10	Number of Operators	Number of simultaneous operators of one RANaaS instance	Natural number (greater or equal to one)	Datasheet	Verification of distinction and partition of data and configuration	Per experiment
11	VM outage duration	When the VM is not active anymore	sec	Datasheet	Script Measurement	Every second (s)
12	Service outage duration	When the base station software crashes	sec	Datasheet	Script Measurement	Every second (s)
13	SLA malfunctioning duration	When base station software performance are not matching KPIs	sec	Datasheet	Script Measurement	Every second (s)
14	RCB connection	If RCB connection is OK	Boolean (yes, no), qualitative	Datasheet	Manual Observation	Anytime needed

B.5.5 Set up of Data Source and Data Collection

RANaaS will have two data sources. The cloud controller will provide the RANaaS with the status of deployment commands, while MaaS is required to provide CPU usage, free RAM, and the number of attached UEs to a given eNB. All these information sources are required and used by RANaaS to deploy new and scale existing VMs. Another function is to improve the overall RAN QoS.

B.5.6 Data Analysis / Identification of Success / Impact analysis

The currently foreseen scaling algorithm of RANaaS will depend of several parameters such as the CPU usage and number of attached UE to an eNB. Besides these parameters, our profiling analysis shall be adjusted to the characteristics of VMs and the traffic profiles of users.

B.6 CDN

The CDN Service is being led by ZHAW, which conducts its development, integration and assessment activities. The following subsections follow the previously presented evaluation guidelines adapted for this service.

B.6.1 Requirements and Strategic Evaluation Objectives

Table 40 – Requirements and Evaluation Objectives

Nº	Requirement	Strategic Evaluation Objective
1	On-Demand Service Instantiation	Support service creation on-demand with all the required components.
2	On-Demand Service Scaling	Support scaling to deliver efficient usage of resources and no service disruptions.
3	monitoring	Support CDN components monitoring for scaling decisions.
4	Charging	Support charging based on metrics collected by monitoring.
6	Performance	CDN should handle a large amount of requests without perceived impact.
7	High Availability	CDN service should have a high uptime.

B.6.2 Definition of Evaluation Areas, Impacts and Selection of Indicators

Table 41 – Evaluation Area, Impact and Indicators definition

Strategic Evaluation Objective	Evaluation Area	Impact	Nº	Indicator's name
Support service creation on-demand with all the required components	IaaS/MCN Service	Instance Works Properly	1	Works Properly
		Time elapsed for complete Instantiation	2	Instantiation Time
		Number of Instances (VMs)	3	Number Instances
Support scaling to deliver efficient usage of resources and no service disruptions	IaaS/MCN Service	Instance Works Properly	1	Works Properly
		Time elapsed for complete Instantiation	2	Instantiation Time
		Number of Instances (VMs)	3	Number Instances
	MCN Service	Decision Time	4	Decision Time
Support CDN components	MCN Service	Event Reaction Time	5	Event Reaction Time

monitoring for scaling decisions.		Availability of CDN Service monitoring Metrics	6	CDN Service Metrics
		Availability of Infrastructure monitoring Metrics	7	Infrastructure Metrics
Support charging based on metrics collected by monitoring	MCN Service	Availability of Charging Capabilities	8	Charging Capabilities
All ICN components should handle a large amount of requests without perceived impact	MCN Service	Number of Requests per Second	9	Number of Requests per Second
		Content Delivery Time	10	Content Delivery Time
CDN service should have a high uptime	MCN Service	Service Uptime	11	Service Uptime

B.6.3 Definition of Success and Targets

Table 42 – Impacts Target

Nº	Indicator's name	Target quantification
1	Instance Works Properly	Yes
2	Time elapsed for complete Instantiation	Time scale of seconds
3	Number of Instances (VMs)	As Defined/Expected (1, 2, 3, etc.)
4	Decision Time	Time scale of milliseconds
5	Event Reaction Time	Time scale of seconds
6	CDN Service Metrics	Yes (CDN service metrics available)
7	Infrastructure Metrics	Yes (Infrastructure metrics available)
8	Charging Capabilities	Yes (Charging models can be applied)
9	Number of Requests per Second	Thousands

10	Content Delivery Time	Time scale of minutes for HD video and seconds for images
11	Service Uptime	> 99% availability

B.6.4 Definition of Evaluation Technique

Table 43 – Indicators assessment

Nº	Indicator's name	Indicator's description	Data/Unit	Data Source	Method of Measurement	Frequency
1	Works Properly	Indicates whether the service works as expected	Boolean (OK, NOK), qualitative	-	Manual Observation	On every deployment change
2	Instantiation Time	Time elapsed until full service instantiation	Seconds (s), quantitative	-	Manual/Script Measurement	On every new service Instantiation
3	Number Instances/Tier	Number of instances per tier	#, Quantitative	-	Manual Observation	On every deployment change (scaling)
4	Decision Time		Millisecond (ms), quantitative	Datasheet	Script Measurement	On every decision (multiple runs)
5	Event Reaction Time		Millisecond (ms), quantitative	Datasheet	Script Measurement	On every monitoring event (multiple runs)
6	CDN Service Metrics	Metrics are present at monitoring service	Boolean (yes, no), qualitative	Code Execution	Manual/Script Measurement	Per experiment
7	Infrastructure Metrics	Metrics are present at monitoring service	Boolean (yes, no), qualitative	Code Execution	Manual/Script Measurement	Per experiment
8	Charging Capabilities	Charging service is	Boolean (yes, no),	Code Execution	Manual Observation	Per experiment

		running without issues	qualitative			
9	Number of Requests per Second	Counter incremented each time a request is received, and after the average per time is calculated	Natural number (greater or equal to one)	Datasheet	Script Measurement	Every 5 minutes
10	Content Delivery Time		Millisecond (ms), quantitative	Datasheet	Script Measurement	Every 5 minutes
11	Service Uptime	Percentage of time the service was working during the experiment	Percentage, quantitative	Datasheet	Script Measurement	Per experiment

B.7 ICN

The ICN Service is being led by UBERN, which conducts its development, integration and assessment activities. The following subsections follow the previously presented evaluation guidelines adapted for this service.

B.7.1 Requirements and Strategic Evaluation Objectives

Table 44 – Requirements and Evaluation Objectives

Nº	Requirement	Strategic Evaluation Objective
1	On-Demand Service Instantiation	Support service creation on-demand with all the required components.
2	On-Demand Service Scaling	Support scaling to deliver efficient usage of resources and no service disruptions.
3	monitoring	Support ICN components monitoring for scaling decisions.
4	Charging	Support charging based on metrics collected by monitoring.
6	Performance	All ICN components should handle a large amount of requests without perceived impact.
7	Compatibility	ICN should support legacy requests (HTTP).

8	High Availability	ICN service should have a high uptime.
9	Cost Savings	ICN should reduce costs in terms of network connectivity.

B.7.2 Definition of Evaluation Areas, Impacts and Selection of Indicators

Table 45 – Evaluation Area, Impact and Indicators definition

Strategic Evaluation Objective	Evaluation Area	Impact	Nº	Indicator's name
Support service creation on-demand with all the required components	IaaS/MCN Service	Instance Works Properly	1	Works Properly
		Time elapsed for complete Instantiation	2	Instantiation Time
		Number of Instances (VMs)	3	Number of Instances
Support scaling to deliver efficient usage of resources and no service disruptions	IaaS/MCN Service	Instance Works Properly	1	Works Properly
		Time elapsed for complete Instantiation	2	Instantiation Time
		Number of Instances (VMs)	3	Number Instances
	MCN Service	Decision Time	4	Decision Time
Support ICN components monitoring	IaaS/MCN Service	Event Reaction Time	5	Event Reaction Time
		Availability of ICN Service monitoring Metrics	6	ICN Service Metrics
		Availability of Infrastructure monitoring Metrics	7	Infrastructure Metrics
Support charging based on metrics collected by monitoring	MCN Service	Availability of Charging Capabilities	8	Charging Capabilities
All ICN components should handle a large amount of requests without perceived	MCN Service	Number of Requests per Second	9	Number of Requests per Second
		Content Delivery Time	10	Content Delivery Time

impact				
ICN should support legacy requests (HTTP)	MCN Service	Support of Legacy Requests	11	Support of Legacy Requests
		Number of Legacy Requests per Second	12	Number of Legacy Requests per Second
ICN service should have a high uptime	MCN Service	Service Uptime	13	Service Uptime
ICN should reduce costs in terms of network connectivity	MCN Service	Bandwidth Savings at Core Networks	14	Bandwidth Savings at Core Networks

B.7.3 Definition of Success and Targets

Table 46 – Impacts Target

Nº	Indicator's name	Target quantification
1	Works Properly	Yes
2	Instantiation Time	Time scale of minutes
3	Number Instances/Tier	As Defined/Expected (1, 2, 3, etc.)
4	Decision Time	Time scale of milliseconds
5	Event Reaction Time	Time scale of seconds
6	ICN Service Metrics	Yes (ICN service metrics available)
7	Infrastructure Metrics	Yes (Infrastructure metrics available)
8	Charging Capabilities	Yes (Charging models can be applied)
9	Number of Requests per Second	Thousands
10	Content Delivery Time	Time scale of minutes for HD video and seconds for images
11	Support of Legacy Requests	Yes (HTTP requests are translated and processed)
12	Number of Legacy Requests per Second	Thousands
13	Service Uptime	> 99% availability
14	Bandwidth Savings at Core Networks	> 20% bandwidth savings

B.7.4 Definition of Evaluation Technique

Table 47 – Indicators assessment

Nº	Indicator's name	Indicator's description	Data/Unit	Data Source	Method of Measurement	Frequency
1	Works Properly	Indicates whether the service works as expected	Boolean (OK, NOK), qualitative	-	Manual Observation	On every deployment change
2	Instantiation Time	Time elapsed until full service instantiation	Seconds (s), quantitative	-	Manual/Script Measurement	On every new service Instantiation
3	Number Instances/Tier	Number of instances per tier	#, Quantitative	-	Manual Observation	On every deployment change (scaling)
4	Decision Time	Time elapsed since an event is received until a decision is made	Millisecond (ms), quantitative	Datasheet	Script Measurement	On every decision (multiple runs)
5	Event Reaction Time	Time elapsed since an event is first detected until an action is taken	Millisecond (ms), quantitative	Datasheet	Script Measurement	On every monitoring event (multiple runs)
6	ICN Service Metrics	Metrics are present at monitoring service	Boolean (yes, no), qualitative	Code Execution	Script Measurement	Per experiment
7	Infrastructure Metrics	Metrics are present at monitoring service	Boolean (yes, no), qualitative	Code Execution	Script Measurement	Per experiment
8	Charging Capabilities	Charging service is running without	Boolean (yes, no), qualitative	Code Execution	Script Measurement	Per experiment

		issues				
9	Number of Requests per Second	Counter incremented each time a request is received, and after the average per time is calculated	Natural number (greater or equal to one)	Datasheet	Script Measurement	Every 5 minutes
10	Content Delivery Time	Time elapsed since a content request is sent until content is received	Millisecond (ms), quantitative	Datasheet	Script Measurement	Every 5 minutes
11	Support of Legacy Requests	HTTP requests handled and answered	Boolean (yes, no), qualitative	Code Execution	Script Measurement	Per experiment
12	Number of Legacy Requests per Second	Counter incremented each time a request is received, and after the average per time is calculated	#, Quantitative	Datasheet	Script Measurement	Every 5 minutes
13	Service Uptime	Percentage of time the service was working during the experiment	Percentage, quantitative	Datasheet	Script Measurement	Per experiment
14	Bandwidth Savings at Core Networks	Bandwidth usage average at core with and without caching	Percentage, quantitative	Datasheet	Script Measurement	Per experiment

B.7.5 Set up of Data Source and Data Collection

Two data sources will exist. The first one is code execution, which enables the visualization of metrics being gathered and components working. The second is a datasheet, with columns representing the multiple indicators and containing the values obtained per measurement frequency. Such datasheet is filled with data gathered directly from code execution (via debug messages/statistics files) or by visualization at appropriate tools. With such datasheet, statistics analysis of indicators evolution over time can be performed and valuable lessons can be taken to improve the service.

B.7.6 Data Analysis / Identification of Success / Impact Analysis

A first set of experiments was performed in order to obtain data for analysis. Not all the indicators were considered, as some are still not available in the current version of the service. Also, only a set of small-scale scenarios was considered, while larger and more complex scenarios are left for a later stage when all the indicators are available and all the platform services are running. These scenarios are described in the table below.

Table 48 – Scenarios

Nº	Number of Routers	Number of Clients	Image Flavour	Requests per Client	Content Object Size
1	1	1	m1.medium	Sequential content requests with pipeline of interests (5 simultaneous queries)	10 Megabytes
2	2	6	m1.medium	Sequential content requests with pipeline of interests (5 simultaneous queries)	10 Megabytes
3	2+1 (after event)	6	m1.medium	Sequential content requests with pipeline of interests (5 simultaneous queries)	10 Megabytes

From this set of scenarios, data for analysis was obtained through multiple runs and processed to gather final average values for the evaluated indicators. These results are presented below.

Table 49 – Indicators results

Nº	Indicator's name	Data/Unit	Scenario 1	Scenario 2	Scenario 3
1	Deployment Time	ms	71300	84817	85683
2	Provisioning Time	ms	10430	10485	10363
3	Decision Time	ms	N/A	N/A	305
4	Event Reaction Time	ms	N/A	N/A	82084
9	Content Delivery Time	ms	2765	5047	3991
10	Support of Legacy Requests	N/A	YES	YES	YES

Considering the results obtained in this first set of experiments with the service, the first general conclusion is that the service is capable of running in a cloud environment with the MCN framework. Moreover, it is quickly deployed and provisioned and promptly delivers the expected performance to end-users. In fact, even when the load increases, if triggers are configured (as in scenario number 3), decisions are readily made and a fast scaling operation is performed to deal with the extra load. Still, as integration with other services such as MaaS will progress, better mechanisms to deal with service scalability will be adopted and results will tend to improve in the long run.

Also noticeable is the existing compatibility with legacy requests, which are not ICN-compatible and need to be translated before processing. These requests were tested and the service was capable of delivering an answer to end-users (the requested content object).

B.8 Infrastructure

The IaaS is being led by Intel, which conducts its integration and assessment activities. The following subsections follow the previously presented evaluation guidelines adapted for this service.

Note that in the section we focus around the components actually being advanced in the MCN project. Services components already available are considered out of scope – e.g. compute service (OpenStack/CloudSigma Compute service) and storage service (OpenStack/CloudSigma Block Storage service).

B.8.1 Requirements and Strategic Evaluation Objectives

Table 50 – Requirements and Evaluation Objectives

Nº	Requirement	Strategic Evaluation Objective
1	Network connectivity QoS	Support of QoS (e.g. bandwidth) in network connectivity services.
2	Inter-DC network connectivity	Support of inter-DC network connectivity.
3	Increased network flexibility	Support traffic steering functionality

B.8.2 Definition of Evaluation Areas, Impacts and Selection of Indicators

Table 51 – Evaluation Area, Impact and Indicators definition

Strategic Evaluation Objective	Evaluation Area	Impact	Nº	Indicator's name
Support of QoS (e.g. bandwidth) in network connectivity services.	IaaS	Network connectivity QoS feature	1	Network QoS
Support of inter-DC network connectivity.	IaaS	Inter-DC network connectivity feature	2	Inter-DC connectivity
Support traffic steering functionality	IaaS	Traffic steering feature	3	Traffic steering

B.8.3 Definition of Success and Targets

Table 52 – Impacts Target

Nº	Indicator's name	Target quantification
1	Network QoS	Yes
2	Inter-DC connectivity	Yes
3	Traffic steering	Yes

B.8.4 Definition of Evaluation Technique

Table 53 – Indicators assessment

Nº	Indicator's name	Indicator's description	Data/Unit	Data Source	Method of Measurement	Frequency
1	Network QoS	-	Boolean (yes, no), qualitative	-	Script Measurement	Per experiment
2	Inter-DC connectivity	-	Boolean (yes, no), qualitative	-	Manual Observation	Per experiment
3	Traffic steering	-	Boolean (yes, no), qualitative	-	Manual Observation	Per experiment

B.9 Monitoring

The monitoring Service is being led by TUB, which conducts its development, integration and assessment activities. The following subsections follow the previously presented evaluation guidelines adapted for this service.

B.9.1 Requirements and Strategic Evaluation Objectives

Table 54 – Requirements and Evaluation Objectives

Nº	Requirement	Strategic Evaluation Objective
1	On-Demand Service instantiation	Support service instantiation and provisioning of parameters.
2	On-Demand Service disposal	Support disposal of service at the end of the MC service lifecycle.
4	Multi-tenancy	Support for multi tenancy
5	monitor resources	Support monitoring of heterogeneous resources.
6	Unified interface	Provide a unified interface for exporting monitoring data towards other MCN services.
7	Multiple consumers	The MCN monitoring service must be usable from a variety of consumers, without any major degradation of performance.
8	Polling mechanisms	The MCN monitoring service must be able to provide monitoring data following a polling mechanism (queries/replies).
9	Asynchronous notification mechanisms	The MCN monitoring service must be able to provide monitoring data following a subscription/notification mechanism, where the consumer can specify the type of data to

		be received, the time interval for periodical notifications or the threshold(s) for alert notifications.
10	Extendibility	The MCN monitoring service must be easily extendable to further monitoring sources, without any major change in the architecture.
11	Integration with MCN services	The MCN monitoring service must be easily integrated with MCN services as a support service. It must be able to interact with the other components of an MCN service, both in terms of monitoring sources and consumers.

B.9.2 Definition of Evaluation Areas, Impacts and Selection of Indicators

Table 55 – Evaluation Area, Impact and Indicators definition

Strategic Evaluation Objective	Evaluation Area	Impact	Nº	Indicator's name
On-Demand Service instantiation and provisioning	IaaS/MCN service	Instance Works Properly	1	Works Properly
		Time elapsed for complete Instantiation	2	Instantiation Time
		Number of Instances (VMs)	3	Number Instances
On-Demand Service disposal	IaaS/MCN service	Time for disposing MaaS	4	Disposal time
Time Efficiency	MCN service	Time elapsed for complete Instantiation	2	Instantiation Time
Multi-tenancy	MCN service	Availability to support multi-tenancy	5	Multi-tenancy
monitor resources	MCN service	Support for monitoring of heterogeneous resources	6	monitoring
Unified interface	MCN service	Provide a unified interface for	7	Unified interface

		monitoring		
Multiple consumers	MCN service	Support for multiple monitoring consumer	8	Multiple consumers
Polling mechanisms	MCN service	Support for polling of monitoring data	9	Polling mechanisms
Asynchronous notification mechanisms	MCN service	Support for asynchronous event notification	10	Asynchronous notification mechanisms
Extendibility	MCN service	Ability to extend monitoring metrics for additional MCN services	11	Extendibility
Integration in MCN services	MCN service	EPCaaS integration	12	EPCaaS integration
		IMSaaS integration	13	IMSaaS integration
		RCBaaS integration	14	RCBaaS integration
		SLAaaS integration	15	SLAaaS integration
		RANaaS integration	16	RANaaS integration

B.9.3 Definition of Success and Targets

Table 56 – Impacts Target

Nº	Indicator's name	Target quantification
1	Works Properly	Yes
1	Instantiation time	< 5 min
3	Number Instances	As Defined/Expected (1, 2, 3, etc.)
4	Disposal time	< 5 min

5	Multi-tenancy	Yes
6	monitoring	Yes
7	Unified interface	Yes
8	Multiple consumers	Yes
9	Polling mechanisms	Yes
10	Asynchronous notification mechanisms	Yes
11	Extendibility	Yes
12	EPCaaS integration	Yes
13	IMSaaS integration	Yes
14	RCBaaS integration	Yes
15	SLAaaS integration	Yes
16	RANaaS integration	Yes

B.9.4 Definition of Evaluation Technique

Table 57 – Indicators assessment

Nº	Indicator's name	Indicator's description	Data/Unit	Data Source	Method of Measurement	Frequency
1	Works Properly	Indicates whether the service works as expected	Boolean (yes, no), qualitative	-	Manual Observation	On every deployment change
	Instantiation Time	Time elapsed until full service instantiation	Seconds (s), quantitative	-	Script Measurement	On every new service Instantiation
3	Number Instances/Tier	Number of instances per tier	#, Quantitative	-	Manual Observation (OpenStack)	On every deployment change (scaling)
4	Disposal time	Time elapsed until full service disposal	Seconds (s), quantitative	-	Script Measurement	On every service disposal
5	Multi-		Boolean (yes, no),	Code	Manual Observation	Per

	tenancy		qualitative	execution		experiment
6	monitoring		Boolean (yes, no), qualitative	Code execution	Manual Observation	Per experiment
7	Unified interface		Boolean (yes, no), qualitative	Code execution	Manual Observation	Per experiment
8	Multiple consumers		Boolean (yes, no), qualitative	Code execution	Manual Observation	Per experiment
9	Polling mechanisms		Boolean (yes, no), qualitative	Code execution	Manual Observation	Per experiment
10	Asynchronous notification mechanisms		Boolean (yes, no), qualitative	Code execution	Manual Observation	Per experiment
11	Extendibility		Boolean (yes, no), qualitative	Code execution	Manual Observation	Per experiment
12	EPCaaS integration		Boolean (yes, no), qualitative	Code execution	Manual Observation	Per experiment
13	IMSaaS integration		Boolean (yes, no), qualitative	Code execution	Manual Observation	Per experiment
14	RCBaaS integration		Boolean (yes, no), qualitative	Code execution	Manual Observation	Per experiment
15	SLAaaS integration		Boolean (yes, no), qualitative	Code execution	Manual Observation	Per experiment
16	RANaaS integration		Boolean (yes, no), qualitative	Code execution	Manual Observation	Per experiment

B.10 Analytics

The Analytics Service is being led by Intel, which conducts its development, integration and assessment activities. The following subsections follow the previously presented evaluation guidelines adapted for this service.

B.10.1 Requirements and Strategic Evaluation Objectives

Table 58 – Requirements and Evaluation Objectives

Nº	Requirement	Strategic Evaluation Objective
1	Multi-tenancy	Support multi-tenancy
2	High Uptime	High uptime and when needed elasticity in the deployment.

B.10.2 Definition of Evaluation Areas, Impacts and Selection of Indicators

Table 59 – Evaluation Area, Impact and Indicators definition

Strategic Evaluation Objective	Evaluation Area	Impact	Nº	Indicator's name
Support multi-tenancy	PaaS	Number of tenants	1	Number of tenants
Support High uptime	PaaS	Number of tenants	1	Number of tenants

B.10.3 Definition of Success and Targets

Table 60 – Impacts Target

Nº	Indicator's name	Target quantification
1	Number of tenants	Support up to 10 parallel working tenants

B.10.4 Definition of Evaluation Technique

Table 61 – Indicators assessment

Nº	Indicator's name	Indicator's description	Data/Unit	Data Source	Method of Measurement	Frequency
1	Number of Tenants	-	#, Quantitative	-	Manual Observation	Per experiment

B.11 MOB

The Mobility Prediction Service is being led by UT, which conducts its development, integration and assessment activities. The following subsections follow the previously presented evaluation guidelines adapted for this service.

B.11.1 Requirements and Strategic Evaluation Objectives

Table 62 – Requirements and Evaluation Objectives

Nº	Requirement	Strategic Evaluation Objective
1	On-Demand Service Instantiation	Support the on-demand MOB service instantiation.
2	Automated Scaling	Support the automated MOB service scaling (in/out) based on system load.
3	Precision	Prediction's precision, based on the received information from MaaS
4	Performance	Prediction's performance, of both the Mobility and Bandwidth, based on the requested time scale

B.11.2 Definition of Evaluation Areas, Impacts and Selection of Indicators

Table 63 – Evaluation Area, Impact and Indicators definition

Strategic Evaluation Objective	Evaluation Area	Impact	Nº	Indicator's name
MOB can be instantiated on demand	MCN Service (ICNaaS/EPCaaS)	On-Demand Deployment	**1**	Instantiation time
MOB can be elastic scaled based on system load	MCN Service (ICNaaS/EPCaaS)	Scaling time and number of instances	**2**	Scaling time and number of instances
Prediction precision will be based on the received information from MaaS	MCN Service (ICNaaS/EPCaaS)	Precision of mobility and bandwidth prediction algorithms	**3**	Prediction precision
Prediction performance will be based on the request's time scale	MCN Service (ICNaaS/EPCaaS)	Performance of mobility and bandwidth prediction algorithms	**4**	Prediction performance

B.11.3 Definition of Success and Targets

Table 64 – Impacts Target

Nº	Indicator's name	Target quantification

1	Deployment time	Deployment in a predefined proper time
2	Scaling time and number of instances	Scaling in a predefined proper time
3	Prediction precision	Prediction in a predefined proper confidence interval
4	Prediction performance	Response in a predefined proper time

B.11.4 Definition of Evaluation Technique

Table 65 – Indicators assessment

Nº	Indicator's name	Indicator's description	Data/Unit	Data Source	Method of Measurement	Frequency
1	Deployment time	Required time to instantiate a MOBaaS on demand	Second		Measurements of time consumption of instantiating/deploying the MOBaaS instance.	
2	Scaling time and number of instances	Required time in order to scaling (in/ out) a MOBaaS based on the system load	Second		Measurements of delay introduced by the scaling operation, and also the system performance before and after the scaling.	
3	Prediction precision	A parameter to measure the precision of prediction	Percentage /Probability/ Prediction granularity (to predict the locations of users in the next X time unit, such as next 1 minutes/hour /day.)	Data source of prediction precision is in general the inputs of MOBaaS, which is MaaS. MaaS should give certain amount of user information reactively/reactively. Prediction precision depends on the granularity of	Simulation	

				the inputs from MaaS.		
4	Prediction performance	A parameter to measure the performance of prediction algorithms	Second/ Successful prediction Percentage (compare the predicted locations with the actual locations, and see the percentage of correctness).		Simulation	

B.11.5 Set-up of Data Source and Data Collection

When MOBaaS is connected with MaaS, all the necessary information used for prediction should be provided on-demand. However, at this moment when the interface between MOBaaS and MaaS is not ready, the inputs to MOBaaS are artificially provided.

The performance of MOBaaS, in the mobility prediction algorithms, depends on the quality of the mobility traces used for training the system. To this end we choose the data provided by Nokia[5] for academic research.

The data collected a wide range of contextual data, including GPS and Cell IDs, from approximately 200 users over 2 years. This provides us several advantages such as:

- Very rich data
- Big variety of user
- Real mobility data (compared to simulated or modelled)
- Location can be taken from GPS
- Another set of location data can be taken from Cell IDs
- The Cell ID location can be further improved using the signal strengths seen by the mobile devices

The data is granted upon signing legal agreements, mainly aiming at protecting the privacy of the participants, and illegal copying. We extracted the mobility traces and trained/ran our mobility prediction algorithms taking these aspects into consideration.

The bandwidth prediction element of MOBaaS is not trained in the way the mobility prediction element is, but it needs fifteen minutes of historical data (prior to the moment the bandwidth

[5] https://www.idiap.ch/dataset/mdc

prediction is evaluated for) to produce accurate results. The algorithm for bandwidth prediction works with any data in flow-record format, where a flow-record typically consists of (at least) the following elements:

- Source (IP + port)
- Destination (IP + port)
- Number of bytes transferred in the flow
- Number of packets transferred in the flow
- Duration of the flow in seconds

Ideally, the algorithm should be fed traces from a mobile provider, but such datasets are not available due to privacy constraints or reasons alike. The algorithm is already validated in the literature, but finding the ideal time scale and defining an appropriate error margin (the two main variables in the algorithm) should be done using real data.
Until real data is available, the algorithm will be tested and developed into a prototype using datasets obtained from university campus networks, for the following reasons:

- Readily available, in the correct data format
- Similar data to provider traffic, as a university is an ISP on its own
- Campus networks typically feature a great number of users
- A large share of the traffic comes from mobile users

The study about how to retrieve the relevant information through MaaS will be carried in the upcoming developments of MOBaaS.

B.11.6 Data Analysis / Identification of Success / Impact Analysis

The performance of prediction algorithm will depend on several issues, which basically can be divided into two types: the granularity of prediction requests and the granularity of inputs. The request granularity means from the MOBaaS consumer point of view, how long the prediction algorithm should provide.

As an example in the mobility prediction, ICNaaS asks MOBaaS to predict the user location in next 1 minute, 10 minutes, 1 hour, or 1 day. Depending on different application scenarios, a high granularity prediction request of course is more difficulty to achieve. The input granularity means the granularity of the data provided from MaaS. Due to huge number of user information MaaS monitors, it might be difficulty to ask MaaS to give a very rich input data as the current off-line Nokia data we're using. For example, MaaS might only be able to provide the user location information every 30 seconds, or even every 10 seconds, instead of every second. Therefore, the reduced input granularity will of course lead to degraded prediction performance.

In the mobility prediction, the performance will be evaluated by comparing the predicted user locations with the actual user locations. In the current experiments, which are based on the Nokia data set, the full input data trace will be equally split into two parts. The first part will be used as the training set of the prediction algorithm. Using the first part of the data trace, prediction algorithm will generate certain amount of predicted user locations. These locations will be compared to the rest part of the location trace, which can give a successful percentage of the prediction algorithm.

For the bandwidth prediction, accuracy is determined by calculating the time-share of the estimated bandwidth requirements being insufficient with regards to the actual demand of bandwidth. Again, prediction performance is highly affected by the amount of information that MaaS is able to provide, but defining a larger error margin can overcome this issue partially, leading to possible over-provisioning of network links. Over-provisioning is not a big issue, although it should be avoided whenever possible. Under-provisioning leads directly to impaired end-user experiences, and therefore considered a 'fail' during evaluation, as expressed in the aforementioned time-share.

B.12 DNS

The DNS Service is being led by ONE, which conducts its development, integration and assessment activities. The following subsections follow the previously presented evaluation guidelines adapted for this service.

B.12.1 Requirements and Strategic Evaluation Objectives

Table 66 – Requirements and Evaluation Objectives

Nº	Requirement	Strategic Evaluation Objective
1	On-Demand Service Instantiation	Support service creation on-demand
2	On-Demand Seamless Service Scaling	Support seamless service scaling
3	Elasticity	Support service scaling
4	Multi-tenancy	Support multi-tenancy
5	High Availability	High uptime and low recovery time
6	Monitoring of service	Support service monitoring
8	Application Performance	The service will meet load conditions

B.12.2 Definition of Evaluation Areas, Impacts and Selection of Indicators

Table 67 – Evaluation Area, Impact and Indicators definition

Strategic Evaluation Objective	Evaluation Area	Impact	Nº	Indicator's name
Support service creation on-demand	IaaS	Deployment time	1	Deployment time
		Provisioning time	2	Provisioning time
Imperceptible service disruption/downtime	IaaS	Provisioning time	1	Provisioning time
		Deployment time	2	Deployment time

Support both horizontal and vertical scaling	IaaS	Provisioning time	2	Scaling time
Support multi-tenancy	IaaS	Number of tenants	3	Number of tenants
High uptime and low recovery time	MCN Service	Service runtime	4	Service runtime
		Deployment time	1	Deployment time
		Provisioning time	2	Provisioning time
Support service monitoring	MCN Service	Availability of monitoring metrics	5	monitoring metrics
The service will meet load conditions	MCN Service	Number of queries per second	6	Number of queries per second
		Service CPU overhead	7	Service CPU overhead
		Service Memory overhead	8	Service Memory overhead
		Response Time	9	Response time

B.12.3 Definition of Success and Targets

Table 68 – Impacts Target

Nº	Indicator's name	Target quantification
1	Works Properly	Yes
2	Instantiation Time	Few Minutes (< ~10 min.)
3	Number Instances/Tier	As Defined/Expected (1, 2, 3, etc.)
4	Service runtime	99% of availability
5	Scaling time	Few Minutes (< ~5 min.)
6	Number of tenants	Unlimited support of tenants
7	Monitoring metrics	Up to date metrics
8	Number of queries per second	Order of thousands

9	Service CPU overhead	Bellow 80%

B.12.4 Definition of Evaluation Technique

Table 69 – Indicators assessment

Nº	Indicator's name	Indicator's description	Data/Unit	Data Source	Method of Measurement	Frequency
1	Deployment Time		ms	Datasheet	On Request Creation On end of creation process	On Every request for deployment (multiple runs)
2	Provisioning time		ms	Datasheet	On request for configuration On end of configuration process	On every request for configuration (multiple runs)
3	Number of Tenants		Natural number (greater or equal to one)	Datasheet	Verification of distinction and partition of data and configuration	Per experiment
4	Service runtime		Percentage of availability	Datasheet	MUT / MUT + MDT	Every 5m
5	Monitoring metrics		Percentage of up to date values	Datasheet	Comparison between Metrics in MaaS and actual values	Every 5m
6	Number of Queries per second		qps	Datasheet	Average queries per second	Every 5m
7	Service CPU overhead		Percentage of CPU usage	Datasheet	Average of CPU usage	Every 5m
8	Service Memory overhead		Percentage of Memory usage	Datasheet	Average of memory usage	Every 5m
9	Response Time		ms	Datasheet	Elapsed time to receive a response from service	On Every Request

B.12.5 Set-up of Data Source and Data Collection

The data source will consist on a datasheet with different columns, each representing an independent variable that corresponds to a specific indicator. Each row will contain the obtained value per measurement frequency.

The aim of this datasheet is to highlight the variations over time.

B.12.6 Data Analysis / Identification of Success / Impact Analysis

The collected data demonstrated a service runtime over 99%, in which the DNSaaS was available to answer to requests. Table 70 summarizes the obtained results for the evaluation techniques of DNSaaS.

Table 70 – Indicators Values

Nº	Indicator's name	Value
1,2	Deployment Time + Provisioning Time	± 28000 ms
3	Number of Tenants	10, 50
4	Service Runtime	99,5%
6	Number of Queries per second (qps)	± 2040 to ±8500
7	Service CPU overhead	±30% to ±90%
8	Service Memory Overhead	±20% to ±50%
9	Response Time	±130ms to ±480ms

Multiple tenants were considered during the evaluation (between 10 and 50). This indicator assessed to which extent the DNSaaS supports multiple clients performing queries for naming resolution. Within the 50 tenants, DNSaaS was able to provide acceptable response time. The number of clients and the number of servers holding the naming resolution affected the number of queries. For instance, with 10 tenants and only one server the number of supported queries relied in ±2040, while with three servers DNSaaS supported around 3500. Also DNSaaS supports a high number of queries per second within 50 tenants, with around 8500 qps.

The Service CPU overhead includes the average of the different servers composing the DNSaaS architecture. For instance, some servers presented a high CPU overhead when there was only a single server for the naming resolution and a high number of tenants. This is the expected behaviour as the number of requests is higher. The memory overhead is not as significant as the CPU overhead, since the data for the service is hold in a database, and each query for naming resolution leads to database query. Moreover, the service to function does not hold too much information on memory. The overhead is mainly in terms of CPU, as a lot of concurrency can be found within higher number of tenants.

An important metric regarding the DNSaaS performance includes the service response time, which was also affected by the number of servers and the number of simultaneous tenants. For instance, within three servers the response time was lower, since the requests were distributed.

B.13 RCB

The RCB Service is being led by Italtel, which conducts its development, integration and assessment activities. The following subsections follow the previously presented evaluation guidelines adapted for this service.

B.13.1 Requirements and Strategic Evaluation Objectives

Table 71 – Requirements and Evaluation Objectives

Nº	Requirement	Strategic Evaluation Objective
1	On-Demand Instantiation	Support the on-demand RCB (Rating, Charging and Billing) instantiation. An MCN service can request an RCB instantiation as support service.
2	Automated Scaling	Support the automated RCB service scaling (in/out) based on information such as (mainly) monitoring (MaaS), but also others.
3	Seamless Scaling	Support of seamless RCB scaling. The MCN Service should not perceive any disruption/downtime.
4	Multi-tenancy	Support of multiple independent RCB instances (allocated to multiple MCN Services). Each tenant is assigned to an isolated RCB environment.
5	Monitoring	Support RCB service monitoring, including both infrastructural and support service level indicators.
6	High Availability	Assure carrier-grade availability levels (e.g. 5 9's), even when regular scaling operations are considered.
7	Performance	The RCB service should support rating, charging and billing operations in the range of thousands per second.

B.13.2 Definition of Evaluation Areas, Impacts and Selection of Indicators

Table 72 – Evaluation Area, Impact and Indicators definition

Strategic Evaluation Objective	Evaluation Area	Impact	Nº	Indicator's name
Support the on-demand RCB (Rating, Charging and Billing) instantiation. An MCN service can request an RCB instantiation as support	IaaS/MCN Support Service	Instance Works Properly	1	Works Properly
		Time elapsed for complete	2	Instantiation Time

service.		Instantiation		
		Number of Instances (VMs)	3	Number Instances
Support the automated RCB service scaling (in/out) based on information such as (mainly) monitoring (MaaS), but also others.	IaaS/MCN Support Service	Instance Works Properly	1	Works Properly
		Time elapsed for complete Scaling	4	Scaling Time
		Number of Instances	3	Number Instances
Support of seamless RCB scaling. The MCN Service should not perceive any disruption/downtime.	MCN Service/MCN Support Service	Downtime during a Scaling operation	5	Scaling Downtime
Support of multiple independent RCB instances (allocated to multiple MCN Services). Each tenant is assigned to an isolated RCB environment.	IaaS/MCN Support Service	Number of Tenants instantiated	6	Number of Tenants
		Multiple Isolated Tenants	7	Tenant Isolation
Support RCB service monitoring, including both infrastructural and support service level indicators.	IaaS/MCN Support Service	Availability of monitoring infrastructure metrics	8	Infrastructure Metrics
		Availability of monitoring RCB service metrics	9	RCB Service Metrics
Assure carrier-grade availability levels (e.g. 5 9's), even when regular scaling operations are considered.	MCN Support Service	Carrier-grade Availability	10	Service Availability
The RCB service should support rating, charging and billing operations in the	MCN Support Service	Number of RCB operations per second	11	Operations per Second

range of thousands per second.		Response Time	12	Response Time

B.13.3 Definition of Success and Targets

Table 73 – Impacts Target

Nº	Indicator's name	Target quantification
1	Works Properly	Yes
2	Instantiation Time	Few Minutes (< ~10 min.)
3	Number Instances	As Defined/Expected (1, 2, 3, etc.)
4	Scaling Time	Few Minutes (< ~5 min.)
5	Scaling Downtime	Few Milliseconds (< ~100 ms)
6	Number of Tenants	As Defined/Expected (1, 2, 3)
7	Tenant Isolation	No Impact Noticeable (see 11, 12)
8	Infrastructure Metrics	Yes (Infrastructure Metrics Available)
9	RCB Service Metrics	Yes (RCB Service Metrics Available)
10	Service Availability	Downtime < 99.999%
11	Operations per Second	[~10, ~1000]
12	Response Time	< 200 ms

B.13.4 Definition of Evaluation Technique

Table 74 – Indicators assessment

Nº	Indicator's name	Indicator's description	Data/Unit	Data Source	Method of Measurement	Frequency
1	Works Properly	Indicates whether the service works as expected	Boolean (Yes, No), qualitative	-	Manual Observation	On every deployment change
2	Instantiation Time	Time elapsed until full service instantiation	Seconds (s), quantitative	-	Manual/Script Measurement	On every new service Instantiation
3	Number	Number of instances	#,	-	Manual Observation	On every deployment

	Instances	(VMs)	Quantitative		(OpenStack)	change (scaling)
4	Scaling Time	Time elapsed until full scaling	Seconds (s), quantitative	-	Script Measurement	On every scaling
5	Scaling Downtime	Downtime (outage) period during the scaling process	Millisecond (ms) or seconds (s), quantitative	-	Script Measurement	On every scaling
6	Number of Tenants	Number of active tenants	#, Quantitative	-	Manual/Script Measurement	Anytime needed
7	Tenant Isolation	Whether tenants are isolated	Boolean (yes, no), qualitative	-	Indirect Observation (see 11, 12)	Anytime needed
8	Infrastructure Metrics	Infrastructure metrics (CPU, memory, etc.) are available	Boolean (yes, no), qualitative	-	Manual Observation	Anytime needed
9	RCB Service Metrics	RCB service metrics (# opr, etc.) are available	Boolean (yes, no), qualitative	-	Manual Observation	Anytime needed
10	Service Availability	Service is up and running	Boolean (yes, no), qualitative	-	Script Measurement	Every second (s)
11	Operations per Second	Number of operations, in average, per second	Opr./second (#/s), quantitative	-	Script Measurement	Every N seconds (s),
12	Response Time	Time elapsed until the client obtains a valid response	Millisecond (ms), quantitative	-	Script Measurement	Every testing request (N second (s))

B.14 AAA

The AAA Service is being led by Italtel, which conducts its development, integration and assessment activities. The following subsections follow the previously presented evaluation guidelines adapted for this service.

B.14.1 Requirements and Strategic Evaluation Objectives

Table 75 – Requirements and Evaluation Objectives

Nº	Requirement	Strategic Evaluation Objective
1	On-Demand Service Instantiation	Support the on-demand AAA (Authentication, Authorization and Accounting) instantiation. An MCN service can request an AAA instantiation as support service.
2	Multi-tenancy	Support of multiple independent AAA instances. Each AAA instance will serve one tenant.
3	Monitoring	Support AAA service monitoring, including both infrastructural and support service level indicators.
4	High Availability	Assure carrier-grade availability levels (e.g. 5 9's), even when regular scaling operations are considered.
5	Performance	The AAA service should authenticate and authorize operations in the range of thousands per second.

B.14.2 Definition of Evaluation Areas, Impacts and Selection of Indicators

Table 76 – Evaluation Area, Impact and Indicators definition

Strategic Evaluation Objective	Evaluation Area	Impact	Nº	Indicator's name
Support the on-demand AAA (Authentication, Authorization and Accounting) instantiation. An MCN service can request an AAA instantiation as support service.	MCN service / MCN composed service	Instance Works Properly	1	Works Properly
		Time elapsed for complete Instantiation	2	Instantiation Time
		Number of Instances (VMs)	3	Number Instances
Support of multiple independent AAA instances.	Many independent MCN services	Number of Tenants instantiated	4	Number of Tenants
		Multiple Isolated Tenants	5	Tenant Isolation

Support AAA service monitoring, including both infrastructural and support service level indicators.	MCN service / MCN composed service	Availability of monitoring infrastructure metrics	6	Infrastructure Metrics
		Availability of monitoring AAA service metrics	7	AAA Service Metrics
Assure carrier-grade availability levels (e.g. 5 9's), even when regular scaling operations are considered.	MCN service / MCN composed service	Carrier-grade Availability	8	Service Availability
The AAA service should authenticate and authorize operations in the range of thousands per second.	MCN service / MCN composed service	Number of AAA operations per second	9	Operations per Second
		Response Time	10	Response Time

B.14.3 Definition of Success and Targets

Table 77 – Impacts Target

Nº	Indicator's name	Target quantification
1	Works Properly	Yes
2	Instantiation Time	Few Minutes (< ~10 min.)
3	Number Instances	As Defined/Expected (1, 2, 3, etc.)
4	Number of Tenants	As Defined/Expected (1, 2, 3)
5	Tenant Isolation	No Impact Noticeable (see 11, 12)
6	Infrastructure Metrics	Yes (Infrastructure Metrics Available)
7	AAA Service Metrics	Yes (AAA Service Metrics Available)
8	Service Availability	Downtime < 99.999%
9	Operations per Second	[~10, ~1000]
10	Response Time	< 200 ms

B.14.4 Definition of Evaluation Technique

Table 78 – Indicators assessment

Nº	Indicator's name	Indicator's description	Data/Unit	Data Source	Method of Measurement	Frequency
1	Works Properly	Indicates whether the service works as expected	Boolean (OK, NOK), qualitative	-	Manual Observation	On every deployment change
2	Instantiation Time	Time elapsed until full service instantiation	Seconds (s), quantitative	-	Manual/Script Measurement	On every new service Instantiation
3	Number Instances	Number of instances (VMs)	#, Quantitative	-	Manual Observation (OpenStack)	On every deployment change (scaling)
4	Number of Tenants	Number of active tenants	#, Quantitative	-	Manual/Script Measurement	Anytime needed
5	Tenant Isolation	Whether tenants are isolated	Boolean (yes, no), qualitative	-	Indirect Observation (see 11, 12)	Anytime needed
6	Infrastructure Metrics	Infrastructure metrics (CPU, memory, etc.) are available	Boolean (yes, no), qualitative	-	Manual Observation	Anytime needed
7	RCB Service Metrics	AAA service metrics (# opr, etc.) are available	Boolean (yes, no), qualitative	-	Manual Observation	Anytime needed
8	Service Availability	Service is up and running	Boolean (yes, no), qualitative	-	Script Measurement	Every second (s)
9	Operations per Second	Number of operations, in average, per second	Opr./second (#/s), quantitative	-	Script Measurement	Every N seconds (s),
10	Response Time	Time elapsed until the client obtains a valid response	Millisecond (ms), quantitative	-	Script Measurement	Every testing request (N second (s))

B.15 SLA

The SLA Service is being led by Intel, which conducts its development, integration and assessment activities. The following subsections follow the previously presented evaluation guidelines adapted for this service.

B.15.1 Requirements and Strategic Evaluation Objectives

Table 79 – Requirements and Evaluation Objectives

Nº	Requirement	Strategic Evaluation Objective
1	On-Demand Service Scaling	Horizontal Scaling
2	monitor services	monitors services with respect to SLA violations
3	Multi-tenancy	Supports multi-tenancy
4	High Availability	High uptime and low recovery time
5	Multiple consumers	Is available to other services for SLA monitoring
6	Asynchronous Event notifications	Alerts service to SLA Violations.

B.15.2 Definition of Evaluation Areas, Impacts and Selection of Indicators

Table 80 – Evaluation Area, Impact and Indicators definition

Strategic Evaluation Objective	Evaluation Area	Impact	Nº	Indicator's name
On-Demand Service Scaling	PaaS	Instance Works Properly	1	Works Properly
		Time elapsed for complete Instantiation	2	Instantiation Time
monitors services with respect to SLA violations	PaaS	monitor service violation	3	monitor service violation
Supports multi-tenancy	PaaS	Multiple tenants	4	Multiple tenants
High uptime and low recovery time	PaaS	Service runtime	5	Service runtime
		Time elapsed for complete Instantiation	2	Instantiation Time
Is available to other services for SLA	PaaS	Multiple consumers	6	Multiple consumers

monitoring				
Alerts service to SLA Violations.	PaaS	Violation notifications	7	Violation notifications

B.15.3 Definition of Success and Targets

Table 81 – Impacts Target

Nº	Indicator's name	Target quantification
1	Works Properly	Yes
2	Instantiation time	< 10 minutes
3	monitor service violation	Yes
4	Multiple tenants	Yes
5	Service runtime	> 95%
6	Multiple consumers	Yes
7	Violation notifications	Yes

B.15.4 Definition of Evaluation Technique

Table 82 – Indicators assessment

Nº	Indicator's name	Data/Unit	Data Source	Method of Measurement	Frequency
1	Works Properly	Boolean (yes, no), qualitative		Manual Observation	Per experiment
1	Deployment time	Millisecond (ms), quantitative	Code execution	Script Measurement	Per experiment
2	Provisioning time	Millisecond (ms), quantitative	Code execution	Script Measurement	Per experiment
3	monitor service violation	Boolean (yes, no), qualitative	Code execution	Script Measurement	Per experiment
4	Multiple tenants	Boolean (yes, no), qualitative	Code execution	Manual/Script Measurement	Per experiment

5	Service runtime	Percentage of availability	Code execution	Script Measurement	Per experiment
6	Multiple consumers	Boolean (yes, no), qualitative	Code execution	Manual/Script Measurement	Per experiment
7	Violation notifications	Boolean (yes, no), qualitative	Code execution	Manual/Script Measurement	Per experiment

B.16 IMS plus RAN/EPC

The integrated prototype system between IMS, EPC and RAN is being led by TUB, which conducts its development, integration and assessment activities. The following subsections follow the previously presented evaluation guidelines adapted for this service.

B.16.1 Requirements and Strategic Evaluation Objectives

Table 83 – Requirements and Evaluation Objectives

Nº	Requirement	Strategic Evaluation Objective
1	Translation time between signalling.	Low translation time.
2	Good interoperability	Interoperability between user equipment on RAN and IMS side.
3	Time efficiency	Fast availability of services in IMS and EPC environment.
4	Service threshold for IMS and EPC	Threshold base on hardware limitations of not virtualized systems.
5	On-Demand Service Scaling	Scaling without any downtime or interruption of procedures.
6	Carrier grade	High availability and reliability for predefined user load.
7	Compatibility with circuit switched call infrastructure.	Translation between signalling protocols.
8	Compatibility with carrier RAN hardware.	Compliance with 3GPP standards concerning interoperability with carrier RAN equipment.
9	User integration in EPC and IMS.	Integrated user identities and handling in IMS and EPC over HSS.
10	Application Performance	Support requests in a specific range per second, assuring service response times under a pre-defined threshold.

B.16.2 Definition of Evaluation Areas, Impacts and Selection of Indicators

Table 84 – Evaluation Area, Impact and Indicators definition

Strategic Evaluation Objective	Evaluation Area	Impact	Nº	Indicator's name
Low translation time.	MCN service	Service request procedure time.	1	Translation time
Interoperability between user equipment in EPC and IMS.	MCN service	Service Interoperability	2	Signalling Interoperability
Fast availability of services in IMS and EPC environment.	MCN service	Service request response time	3	Availability Time
Threshold base on hardware limitations of not virtualized systems.	RAN/MCN service	Call drop rate. Service availability.	4	Availability
Scaling without any downtime or interruption of procedures.	MCN service	Scaling time	5	Scaling time
High availability and reliability for predefined user load.	MCN service	Service reliability Call drop rate.	6	Carrier grade
Translation between signalling protocols	MCN service	Compatibility of services	7	Compatibility of service
Compliance with 3GPP standards concerning interoperability with carrier RAN equipment.	RAN/MCN service	Compatibility of hardware	8	Compatibility of hardware
Integrated user identities and	EPC/MCN service	User identities	9	User Integration.

handling in IMS and EPC over HSS.				
Support requests in a specific range per second, assuring service response times under a pre-defined threshold.	IaaS/MCN Service	Number of requests per second	10	Requests per second
		Response Time	11	Response Time
			12	Ground load

B.16.3 Definition of Success and Targets

Table 85 – Impacts Target

Nº	Indicator's name	Target quantification
1	Translation time	< 500 ms
2	Signalling Interoperability	Support of mandatory basic call procedures.
3	Availability Time	< 4 sec for call procedure round trip.
4	Availability	8 user per cell
		100 per MME
5	Scaling time	< 5 min
6	Carrier grade	99,6% under no load 99% for ground load
7	Compatibility of service	Support of mandatory basic procedures.
8	Compatibility of hardware	Support of mandatory basic procedures.
9	User Integration	1:1 mapping
10	Requests per second	> 20 cps (for a small instance type)
11	Response Time	First response < 500 ms
12	Ground load	6 user per cell
		50 user per MME

B.16.4 Definition of Evaluation Technique

Table 86 – Indicators assessment

Nº	Indicator's name	Data/Unit	Data Source	Method of Measurement	Frequency

1	Translation time	Millisecond (ms), quantitative	MSC	Message input, translated message output.	N/A
2	Signalling Interoperability	Boolean (yes, no), qualitative	UA/UE	Message procedure input, output on a call success.	N/A
3	Availability Time	Seconds (s), quantitative	IMS Bench	At start of procedure until procedure success.	10 / second
4	Availability	User	MSC	Success calls on time.	N/A
5	Scaling time	Seconds (s), quantitative			
6	Carrier grade	Success/ attempt, quantitative	IMS Bench	Start with procedure beginning and end with success or timeout	10 / second
7	Compatibility of service	success/ attempt, quantitative	MSC/MME/EPC	Start with procedure beginning and with success of procedure failure.	N/A
8	Compatibility of hardware	Success/ attempt, quantitative	MSC/MME/EPC	Start with procedure beginning and with success of procedure failure.	N/A
9	User Integration	Boolean (yes, no), qualitative	HSS	Interworking	Per register and call attempt
10	Requests per second	Attempt/second, quantitative	IMS Bench	Elapsed time to receive a response from service	10 / sec in IMS N/A EPC
11	Response Time	Millisecond (ms), quantitative	IMS Bench MSC/MME/EPC	Time to first response.	10 / second 10 / second for MME
12	Ground load	User per cell, quantitative	MSC/MME/EPC	MSC output/ MME output	N/A

B.17 MCN-Enabled Digital Signage plus RAN/EPC

The integrated prototype system between DSS, EPC and RAN is being led by STT, which conducts its development, integration and assessment activities. The following subsections follow the previously presented evaluation guidelines adapted for this service.

B.17.1 Requirements and Strategic Evaluation Objectives

Table 87 – Requirements and Evaluation Objectives

Nº	Requirement	Strategic Evaluation Objective
1	Smart Content Location and Follow-Me-Cloud	DSS, along with EPC and RAN, will be able to display location related media
2	Remote Control	DSS, thanks to EPC and RAN, will be accessed and used from any location and in any moment
3	General performance improvement	DSS, thanks to EPC and RAN will decrease the downloading times for contents

B.17.2 Definition of Evaluation Areas, Impacts and Selection of Indicators

Table 88 – Evaluation Area, Impact and Indicators definition

Strategic Evaluation Objective	Evaluation Area	Impact	Nº	Indicator's name
DSS, along with EPC and RAN, will be able to display location related media	IaaS/MCN Service	Faster location related content	1	Faster location related content
DSS, thanks to EPC and RAN, will be accessed and used from any location and in any moment	IaaS/MCN Service	Higher availability (location and moment)	2	Higher availability
DSS, thanks to EPC and RAN will decrease the downloading times for contents	IaaS/MCN Service	Faster content downloading	3	Faster content downloading

B.17.3 Definition of Success and Targets

Table 89 – Impacts Target

Nº	Indicator's name	Target quantification
1	Faster location related content	Faster location of the user, allowing almost instant related location info
2	Higher availability	-
3	Faster content downloading	Higher speed downloading

B.17.4 Definition of Evaluation Technique

Table 90 – Indicators assessment

Nº	Indicator's name	Data/Unit	Data Source	Method of Measurement	Frequency
1	Faster location related content	Millisecond (ms), quantitative	Datasheet	Average time to display located contents	Each content downloading request
2	Higher availability	#, Quantitative	Datasheet	Average number of accesses	Each access
3	Faster content downloadin g	Millisecond (ms), quantitative	Datasheet	Elapsed time from start and finish downloads	Each time a download is done

References

D2.1. (2013) Reference Scenarios and Technical System Requirements Definition, MobileCloud Networking Project

D2.2. (2013) Overall Architecture Definition, MobileCloud Networking Project

D3.1. (2013) Infrastructure Management Foundations – Specifications & Design for MobileCloud framework, MobileCloud Networking Project

D3.2. (2014) Infrastructure Management Foundations – Components First Release, MobileCloud Networking Project

D3.3. (2014) Infrastructure Management Foundations – Components Final Release, MobileCloud Networking Project (to appear)

D5.1. (2013) Design of Mobile Platform Architecture and Services, MobileCloud Networking Project

D5.2. (2014) Implementation of IMSaaS, DSN and Mobile Platform, MobileCloud Networking Project

D6.1. (2014) Initial Report on Integration and Evaluation Plans, MobileCloud Networking Project

DoW. (2012) Description of Work, Mobile-Cloud Networking