# Data Centres Optimization for Energy-Efficient and EnvironmentalLy Friendly INternet

**Funding scheme:** Specific Targeted Research Projects – STREP

**Co-funded by the European Commission within the Seventh Framework Programme**

**Project no. 609140**

**Strategic objective:** FP7-SMARTCITIES-2013 (ICT-2013.6.2)

**Start date of project:** October 1st, 2013 (36 months duration)

SEVENTH FRAMEWORK PROGRAMME

# Deliverable D5.1

# Testbed description and testing scenarios

**Due date:** 30/09/2015

**Submission date:** 06/11/2015

**Deliverable leader:** GRNET

**Author list:** Vangelis Angelou (GRNET), Michal Balcerkiewicz (PSNC), Matteo Biancani (IRT), Stuart Clayman (UCL), Alex Galis (UCL), Domenico Gallico (IRT), Georgios Goumas (GRNET), Andrea Gronchi (NXW), Amaia Legarrea (i2CAT), Alina Nistor (SIEMENS), Septimiu Nechifor (SIEMENS), Ariel Oleksiak (PSNC), Roberto Spigolon (WIND), Artemis Voulkidis (SYNELIXIS), Theodore Zahariadis (SYNELIXIS), Tommaso Zini (NXW)

Dissemination Level

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

# List of Contributors

| Participant | Contributor |
|---|---|
| GRNET | Vangelis Angelou, Georgios Goumas |
| i2CAT | Amaia Legarrea |
| UCL | Stuart Clayman, Alex Galis |
| SIEMENS | Alina Nistor, Septimiu Nechifor |
| PSNC | Michał Balcerkiewicz, Ariel Oleksiak |
| IRT | Domenico Gallico, Matteo Biancani |
| WIND | Roberto Spigolon, Aniello Reale |
| SYN | Artemis Voulkidis, Theodore Zahariadis |
| NXW | Tommaso Zini, Andrea Gronchi |

# Amendment History

| Version | Date | Partners | Description/Comments |
|---|---|---|---|
| 0.1 | 28/01/2015 | GRNET | Initial draft |
| 0.2 | 17/06/2015 | i2CAT | Additional feedback to ToC |
| 0.3 | 7/7/2015 | GRNET | Example testing scenario |
| 0.4 | 30/7/2015 | GRNET | Initial introduction; Initial testbed descriptions; Added 2 more example scenario definitions. |
| 0.5 | 31/7/2015 | I2CAT | Initial testing plan and demo description |
| 0.6 | 23/8/2015 | GRNET | Initial testing metrics definition |
| 0.7 | 3/9/2015 | GRNET | Added GRNET testbed description |
| 0.8 | 14/9/2015 | GRNET | Added detailed testing objectives; updated GRNET testbed description |
| 0.9 | 14/9/2015 | UCL, SIEMENS | VLSP updates and SmartGrid testing scenario definition |
| 0.10 | 16/9/2015 | GRNET | Added preliminary Executive Summary; updated testing scenarios; changed document section order |
| 0.11 | 17/9/2015 | WIND, IRT, PSNC | Added/updated testbed descriptions |
| 0.12 | 23/9/2015<br>02/10/2015 | GRNET<br>SYN | Updated metric definitions, updated testbed section<br>Revised all document |
| 0.13 | 30/09/2015 | WIND | Updated test bed description, according to GRNET skeleton |
| 0.14 | 01/01/2015 | i2CAT | Updated i2CAT test bed description |
| 0.15 | 01/10/2015 | PSNC | Updated PSNC test bed description |
| 0.16 | 05/10/2015 | i2CAT | Merged version from 0.12v6, 0.13, 0.14 and 0.15.<br>Section 4, section 5.1.2. All changes accepted. |
| 0.17 | 07/10/2015 | NXW | Updated testing scenarios<br>Revised all document |
| 0.18 | 08/10/2015 | UCL | Document Review |
| 0.19 | 08/10/2015 | IRT | Updated testbed description |
| 0.20 | 09/10/2015 | SIEMENS | Added testing scenarios description |
| 0.21 | 12/10/2015 | GRNET | General input to all sections |
| 0.22 | 13/10/2015 | UCL | Testbed description update |
| 0.23 | 14/10/2015 | i2CAT | Merged version and general review of the whole document. Substantial input to section 4. |
| 0.24 | 19/10/2015 | GRNET | Small additions to the testing scenarios. |
| 0.25 | 19/10/2015 | SIEMENS | Updated smart grid testing scenario steps |
| 0.26 | 23/10/2015 | SYN | Review and update of the whole document |
| 0.27 | 27/10/2015 | SIE, PSNC | Updates |
| 0.28 | 27/10/2015 | UCL | Updates |
| 0.29 | 3/11/2015 | GRNET, SYN, NWX, i2CAT | Updates |
| RC1 | 4/11/2015 | GRNET | Review and update of the whole document |
| FF | 5/11/2015 | I2CAT, IRT | Final Review |

# Table of Contents

# Figures Summary

# Tables Summary

# Abbreviations

| | |
|---|---|
| AC | Alternating Current |
| ADR | Automated Demand Response |
| AM | Aggregate Managers |
| AP | Action Point |
| API | Application Programming Interface |
| BMS | Benchmarking Monitoring Statistics |
| BoD | Bandwidth on Demand |
| CLP | Command Line Protocol |
| CPU | Central Processing Unit |
| DC | Data Centre |
| DLC | Direct Liquid Cooling |
| DoW | Description of Work |
| EC | European Commission |
| EP | Energy Provider |
| EU | European Union |
| FP7 | Seventh Framework Programme |
| FTP | File Transfer Protocol |
| GENI | Global Environment for Network Innovations |
| GPU | Graphics Processing Unit |
| GRE | Generic Routing Encapsulation |
| GUI | Graphical User Interface |
| HEN | Heterogeneous Experimental Network |
| HPC | High Performance Computing |

| | |
|---|---|
| HTTP | HyperText Transfer Protocol |
| IaaS | Infrastructure-as-a-Service |
| iLO | Integrated Lights-Out |
| IM | Island Managers |
| IPMI | Intelligent Platform Management Interface |
| IPR | Intellectual Properties Rights |
| LAN | Local Area Network |
| LDAP | Lightweight Directory Access Protocol |
| LEAs | Linux Embedded Applications Systems |
| LGPL | Lesser General Public License |
| LIMS | Lightweight IP Measurement System |
| LNH | Lightweight Network Hypervisor |
| MPLS | Multiprotocol Label Switching |
| NaaS | Network as a Service |
| OCF | OFELIA Control Framework |
| OSI | Open Systems Interconnection |
| PM | Person Months |
| PMO | Project Management Office |
| PUE | Power Usage Effectiveness |
| PV | Photovoltaic |
| QCT | Quality Check Team |
| QoE | Quality of Experience |
| QoS | Quality of Service |
| ROADM | Reconfigurable Optical Add-Drop Multiplexer |
| SDN | Software-Defined Networking |

| | |
|---|---|
| SGC | Smart Grid Controller |
| SLA | Service Level Agreement |
| SLARC | Service Level Agreement Renegotiation Controller |
| SMASH | Systems Management Architecture for Server Hardware |
| SNMP | Simple Network Management Protocol |
| TCP/IP | Transmission Control Protocol/Internet Protocol |
| ToC | Table of Content |
| TMC | Technical Management Committee |
| TO | Test Objective |
| UC | Use Case |
| UI | User Interface |
| UMF | Unified Management Framework |
| UPS | Uninterruptible Power Supply |
| VGA | Video Graphics Adapter |
| VIM | Virtual Infrastructure Management |
| VIS | Virtual Infrastructure Information Service |
| VLAN | Virtual Local Area Network |
| VLSP | Very Lightweight Software Driven Network and Service Platform |
| VMC | Virtual Machine Container |
| VR | Virtual Router |
| VT-AM | Virtualisation Aggregate Managers |
| WAN | Wide Area Network |
| WP | Work package |

# Executive Summary

This document is the deliverable D5.1 “Testbed description and testing scenarios” of Work Package 5 “System integration, trials and performance evaluation” of the FP7-SMARTCITIES-2013 DOLFIN project. It is based on and it is a continuation of the project work from:

- D2.2 “DOLFIN requirements, system architecture and testbeds”
- D3.1 “Data Centre energy consumption optimization platform (eCOP) (Design)”
- D4.1 “Synergetic Data Centres for energy efficiency (Design)”

DOLFIN aims to significantly contribute to the energy efficiency of Data Centres (DCs) and the simultaneous stabilization of Smart Grids, by coordinating federated DCs and Smart Grids through a holistic approach. DOLFIN will model and monitor energy consumption at various levels (DC, room, rack, server) and enable seamless, automatic migration of Virtual Machines (VMs) between servers of the same DC or across a group of energy-conscious, Synergetic DCs, aiming to:

- optimize the overall DC energy consumption by dynamically changing the percentage of active versus stand-by servers and the load per active server in a DC, and
- provide balancing/stabilization services to the Smart Grid, particularly under peak load, by dynamically adjusting the energy consumption characteristics of the local DCs.

This deliverable provides a detailed description of the testing methodology to be employed during integration, verification and demonstration of DOLFIN as a system and of the testbeds provided for such tasks by the participating partners.

The document is organized so as to provide:

- A general description of DOLFIN’s business objectives and identification of key features that are central to the integration process and are to be demonstrated in the testing phase.
- A definition of testing measurements that serve as quantifiable indicators of the satisfiability of testing scenario outcomes.
- A specification of the organization of testing measurements into testing objectives, high-level aggregation of measurements that are designed in line with the DOLFIN use cases to unambiguously verify DOLFIN’s operation.
- The design of testing scenarios, based on DOLFIN use cases and specific testing objectives that will be employed during the integration process and that can also be used for DOLFIN demonstration.
- The outline of the testing plan for DOLFIN, containing the different testing phases for DOLFIN and the specific modules that are to be integrated in each phase.
- The detailed setup and operation of the testbeds that will be used for integration and demonstration purposes.

# 1. Integration and testing methodology

DOLFIN is a determined project that aims at introducing a novel paradigm in coordinated DC energy consumption optimisation, by considering that DCs need to interface with Smart Grids, while supporting multi-objective Service Level Agreements (SLAs). In this deliverable we present DOLFIN's integration approach coupled with an extensive set of functionality and performance tests to evaluate the individual DOLFIN components and the overall system. The testbeds and test cases defined and implemented in DOLFIN are in line with the relevant business scenarios and use cases detailed in [1].

The validation and evaluation activities foreseen in DOLFIN have the following objectives:

- Validate that all DOLFIN components a) are free of software errors (bugs), b) are functioning according to their detailed design in [2] and [3], c) meet the relevant system requirements and d) are properly integrated to form the DOLFIN platform, described in [1].
- Perform an extensive, multi-level functional evaluation that will verify the successful implementation of the DOLFIN system in accordance to the relevant strategic objectives that include:
    - The injection of energy awareness to modern DCs by providing a) a dynamic, service-effective and energy-efficient monitoring framework for all resources in a DC and b) a policy enforcement framework for the (re)allocation of resources which will be used to optimise the aggregate demand across a group of synergetic DCs.
    - The incorporation of a set of mechanisms to enable interaction and collaboration with Smart Grid networks in order to contribute to the energy stabilisation of the power distribution system.
    - The incorporation of a set of mechanisms to enable online SLA negotiation, considering the contractual quality of service, the energy required to be consumed for service delivery and the overall cost of the service.
- Carry out a quantitative performance evaluation of the DOLFIN system in line with the business scenarios described in [1] that designate the following key goals: a) energy reduction, b) energy stabilisation and c) improvement in the quality of the energy mix.

The Integration and testing methodology of the DOLFIN project is based on and it is a continuation of the system work from:

- D2.2 “DOLFIN requirements, system architecture and testbeds”
- D3.1 “Data Centre energy consumption optimization platform (eCOP) (Design)”
- D4.1 “Synergetic Data Centres for energy efficiency (Design)”

The validation and evaluation strategy in DOLFIN is heavily based on the seven core Use Cases defined in [1] which directly derive from the aforementioned business scenarios. These core Use Cases are briefly summarized in Table 1 – DOLFIN Use Cases

The core Use Cases (UCs) are further expanded to achieve optimization at three operation levels: a) the energy-conscious single DC level where energy consumption optimization is applied to a single DC, based on system virtualization and optimal VMs distribution and proper HVAC/Lighting control, b) the group of energy-conscious synergetic DCs level, where the cumulative energy consumption optimization in a group of DCs is targeted, based on optimal VMs distribution across the whole set of servers belonging to the group of DCs using policy-based methods and c) the Smart City level, where energy consumption optimization at the Smart City level is targeted, providing stabilization of the local Smart Grid, based on VMs distribution across the servers that are part of a group of DCs, following an electricity demand-response approach.

| Core Use Case ID | Core Use Case Name | Use Case ID | Use Case Name |
|---|---|---|---|
| **UC1** | Energy-efficient workload redistribution | **UC.1.1** | Energy-efficient workload redistribution using a single DC |
| | | **UC.1.2** | Energy efficient workload redistribution using synergetic DC of the same administrative domain |
| | | **UC.1.3** | Energy efficient workload redistribution using synergetic DC of alternative administrative domain |
| **UC2** | Multi tariffs from the Utility companies | **UC.2.1.1** | Multi tariffs from the Utility companies in the case of a single DC |
| | | **UC.2.1.2** | Multi tariffs from the Utility companies using a synergetic DC of the same administrative domain |
| | | **UC.2.1.3** | Multi tariffs from the Utility companies using synergetic DC of alternative administrative domain |
| **UC3** | Flexible contract between the Utility, the Smart Grid | **UC.2.2.1** | Flexible contract between the Utility, the Smart Grid and the DC owner in the case of a single DC |
| | | **UC.2.2.2** | Flexible contract between the Utility, the Smart Grid and the DC owner using a synergetic DC of the same |

| | | | |
|---|---|---|---|
| | and the DC owner | | administrative domain |
| | | **UC.2.2.3** | Flexible contract between the Utility, the Smart Grid and the DC owner using synergetic DC of alternative administrative domain |
| **UC4** | SLA renegotiation with end customers | **UC.2.3.1** | SLA renegotiation with end customers in the case of a single DC |
| | | **UC.2.3.2** | SLA renegotiation with end customers using a synergetic DC of the same administrative domain |
| | | **UC.2.3.3** | SLA renegotiation with your customers using a synergetic DC of alternative administrative domain |
| **UC5** | Green-powered service to end-customers | **UC.3.1.1** | Green-powered service to end-customers in case of a single DC |
| | | **UC.3.1.2** | Green-powered service to end customers, using a synergetic DC of the same administrative domain |
| | | **UC.3.1.3** | Green-powered service to end customers using a synergetic DC of alternative administrative domain |
| **UC6** | Optimize benefits/incentives from national/European authorities | **UC.3.2.1** | Optimize benefits/incentives from national/European authorities in the case of a single DC |
| | | **UC.3.2.2** | Optimize benefits/incentives from national/European authorities, using a synergetic DC of the same administrative domain |
| | | **UC.3.2.3** | Optimize benefits/incentives from national/European authorities, using a synergetic DC of alternative administrative domain |
| **UC7** | Smart City | **UC.3.3.1** | Smart City using a single DC |
| | | **UC.3.3.2** | Smart City using a synergetic DC of the same administrative domain |
| | | **UC.7.3** | Smart City using a Synergetic DC of alternative |

| | | | |
|---|---|---|---|
| | | | administrative domain |

Table 1 – DOLFIN Use Cases

# 2. Testing objectives

As a general principle to test the DOLFIN system, the main objectives addressing the needs of both users and DC operators, need to be made clear and quantifiable. In this section, we elaborate on the main features provided by DOLFIN and we set tangible indicators to measure their effectiveness. These measures are high-level aggregations of quantifiable data that can be monitored during DC operation and can demonstrate that DOLFIN meets the demands of the use cases previously described.

In short, DOLFIN aims to:

- Reduce the amount of energy consumed by DCs in order to provide direct and indirect benefits to DC operators.
- Manipulate the amount of energy consumed by the DC in order to stabilize energy consumption in geographical and temporal terms.
- Work with energy providers to provide computing services that take advantage of a "green energy mix" and are, thus, environmentally friendly.

These general business objectives can be translated into the following technical objectives that form the core targets of the DOLFIN demonstrations:

A. **Optimization of energy consumption:** The optimization of the energy consumption is the most important aspect in terms of broad project impact. Reduced DC energy consumption alleviates the environmental impact of modern infrastructures, while improved efficiency benefits both DC operators in terms of reduced operational costs and consumers, as operational savings will trickle down to reduce the price of virtual infrastructures.

   DOLFIN uses an online optimization approach to reducing energy consumption that takes into account monitoring information about the DC environment from internal and external sources.

B. **Temporal and geographical shifting of computer processing:** Shifting DC computing load in time and space may result in optimal placement of computing workload across a given set of physical machines. In this way, reduced energy consumption through server shutdown, as well as greater exploitation of locally generated green power may be achieved.

   In more detail, time shifting workloads is made possible in DOLFIN-enabled DCs through exploitation of different SLA tariffs, implying more flexible large scale data processing. As the process of shifting time-critical load requires agreement between the customer and the DC operator, the extra flexibility provided from customers is rewarded with lower tariffs by DC operators for such workloads.

Space or geographical shifting of computation is made possible by modern Cloud infrastructures that use virtualization. Thus, virtualized workloads can be migrated between DCs in different geographical areas, benefitting from local advantages (e.g. lower tariffs from energy providers (EPs), greener energy mix, regional incentives, etc.).

C. **Responsiveness to energy grid price and power fluctuations:** DOLFIN's capability to respond to Smart Grid information enables DCs to contribute to stabilization of local energy consumption. Moreover, DC operators are given the opportunity to reduce their operating costs by varying their DC's consumption based on price fluctuations.

   In addition, a "green compute service" is made possible in DOLFIN-enabled DCS, in the cases where the energy mix provided to DCs across a DOLFIN system is taken into account. In this context, users wishing to operate on green energy can sign up for a green SLA and have their computational workloads run only on renewable energy powered servers or on DCs with the greenest energy mix currently available.

D. **Ensuring DC user experience by defining and re-negotiating SLAs:** Computing performance is often difficult to quantify, especially in terms of user satisfaction. A formal SLA defines key expectations of performance as perceived by the user, being able to renegotiate these SLAs between DC users and operators is a key feature of DOLFIN that ensures minimal performance degradation during energy consumption optimizations. Also, users can benefit from price incentives offered by DCs as a result of reduced anticipated power consumption.

Each of these key features can be measured using a set of measurements that are combined to form quantifiable testing objectives. The achievement of these testing objectives measurements will verify the DOLFIN conditions in a specific test scenario outcome. The set of quantifiable measurements is presented in detail in Section 3 of this document. The measurements that will be obtained from the set of the DOLFIN use cases include:

- Energy consumption in different tiers of the DOLFIN architecture (e.g. DC federation, DC, area, rack, server) and in different types of infrastructure (e.g. computing, network or cooling infrastructure). Along with raw energy consumption measurements, we have to take into account the energy costs, which are a different but also important indicator for DC operators.
- The number of time/space shifting operations performed by DOLFIN as a result of DC optimization attempts. These are indicators of the aggressiveness of the optimization and of the variety of the operating conditions in DOLFIN-enabled DCs.
- The amount of $CO_2$-equivalent saved by reducing energy consumption in the DC and the corresponding user-base recognition of the green nature of DOLFIN-enabled DCs and consequently, an increased number of environmentally-aware customers that could potentially benefit from direct incentives provided by government or local authorities when provable energy consumption or green energy mix targets are met.
- The number of SLA violations incurred and the penalties levied during a time period can be indicative of the DOLFIN-enabled DC performance and the trade-offs between energy reduction and user experience.

| Measure ID | Measure Description |
|---|---|
| **A.1** | DOLFIN system energy consumption |
| **A.1.1** | DOLFIN-enabled DC energy consumption |

| | |
|---|---|
| **A.1.1.1** | DC computing equipment energy consumption |
| **A.1.1.2** | DC networking equipment energy consumption |
| **A.1.1.3** | DC cooling equipment energy consumption |
| **A.1.1.4** | DC general-purpose equipment energy consumption |
| **A.1.1.5** | DC area equipment energy consumption |
| **A.1.1.6** | DC area computing equipment energy consumption |
| **A.1.1.7** | DC area networking equipment energy consumption |
| **A.1.1.8** | DC area cooling equipment energy consumption |
| **A.1.1.9** | DC area general-purpose equipment energy consumption |
| **A.1.2** | DOLFIN federation layer energy consumption |
| **A.2** | DOLFIN system energy costs |
| **A.2.1** | DOLFIN-enabled DC energy costs |
| **B.1** | DOLFIN system Cross-DC VM migration operations |
| **B.2** | DOLFIN system workload time-shifting operations |
| **C.1** | DOLFIN system $CO_2$ footprint |
| **C.1.1** | DOLFIN-enabled DC $CO_2$ footprint |
| **C.2** | DOLFIN user $CO_2$ footprint |
| **C.3** | DOLFIN customer base |
| **C.4** | DOLFIN system direct incentives to lower peak load |
| **C.4.1** | DOLFIN-enabled DC direct incentives to lower peak load |
| **D.1** | DOLFIN system SLA violations |
| **D.2** | DOLFIN system penalties from SLA violations |
| **D.3** | DOLFIN system SLA renegotiations |

**Table 2 - DOLFIN testing measures overview.**

As previously presented, collections of quantifiable indicators are combined to provide clear and measurable testing objectives. Although the objectives to be demonstrated have to relate to a particular test scenario and are presented in detail in Section 3, each DOLFIN Use Case has a respective expected outcome (as described in the "Business Objectives" and "OPEX Analysis" fields for each Use Case description in [1]), which can be construed into a test objective by outlining the measurements that can be used to define and test it.

Aiming to extract the most relevant testing objectives, each use case's outcome is examined and then matched with the available measurements for relevance. Different sets of measurements are then organized into testing objectives that are only missing specific target values, providing realistic thresholds depending on every target scenario, as only then the operating environment is defined.

By examining each use case presented in D2.2 [4], we can derive the following testing objectives per use case, using the notation of D2.1, Section 4:

- **UC 1.1:** When the overall DC efficiency reaches a given threshold a re-organization is triggered to increase it. In this use case, the expected outcome is to reduce operational costs, by reducing the total energy consumed, while respecting the SLAs. As a result, the relevant measurements include A.1, A.1.1, A.1.1.1, A.1.1.2, A.1.1.3, A.1.1.4, A.1.1.5, A.1.1.6, A.1.1.7, A.1.1.8, A.1.1.9, A.2, A.2.1, D.1 and D.2, and can be targeted by "Test Objective (TO) 1".

- **UC 1.2 and 1.3:** Some DCs of the same administrative domain could find it profitable to increase the computational workload in order to get higher power efficiency. Hence, these DCs can contact cooperating DCs, offering to get part of their workload. In this way the overall efficiency of the DC network could be improved. Although the expected outcome is the same as in UC 1.1, the relevant measurements include additionally A.1.2 and B.1 due to the space-shifting of workloads between DOLFIN-enabled DCs, and are described in "TO2".

- **UC 2.1.1:** In this use case, the DC is aware of different tariffs from the local EP and can try to take advantage of this information by time-shifting workloads to reduce costs, whilst respecting user SLAs. So, the measurements for this UC include the same measurements as UC 1.1, adding metric B.2, leading to "TO3" for this UC.

- **UC 2.1.2 and 2.1.3:** These use cases deploy all available workload mitigation techniques whether within or across administrative domains. They are covered by "TO4", which includes measurements from "TO1" along with measurements A.1.2, B.1 and B.2.

- **UC 2.2.1:** This use case refers to a single DC, having contracted with the energy provider regarding Smart Grid requests pertaining to the stability of the grid. For example, the utility grid might need the DC to lower the energy consumption for a specific time period, and in response, incentives will be offered to the DC operator for compliance. In this case, the relevant test objective (TO5) contains all measurements relevant to "TO3" along with the measurements C.4 and C.4.1.

- **UC 2.2.2 and 2.2.3:** The contract assumed for UC 2.2.1 applies in these use cases, as well, but the contracting DC participates in a DC federation, which may be exploited for achieving grid stability. These use cases are covered by "TO6", which includes the measurements from "TO4", along with measurements C.4 and C.4.1.

- **UC 2.3.1:** This use case assumes additionally SLA renegotiation between the users and the DC in order to facilitate time-shifting operations. The testing objective for these cases is noted as "TO7" and builds on "TO3", adding metric D.3, as well.

- **UC 2.3.2 and 2.3.3:** These use cases cover SLA renegotiation with the additional option of space- shifting operations. The relevant test objective is "TO8" and is composed on "TO4" measurements, adding metric D.3.

- **UC 3.1.1**: In this use case, the energy mix and the efficiency of a single DC user workload are paramount, as the user has requested a green SLA for her workload. The outcome assumes lowering the $CO_2$ footprint and consequent tariffs, lowering of the overall electricity cost of the administrative domain, improving the brand recognition and increasing the customer base. "TO9" is defined for this case and contains "TO3" measurements along with C.1, C1.1, C.2 and C.3.

- **UC 3.1.2 and 3.1.3:** Similarly to UC 3.1.1, green SLAs are targeted for user workload, assuming synergetic DCs. The test objective is marked as "TO10" and builds on "TO4" measurements, adding C.1, C1.1, C.2 and C.3 measurements.

- **UC 3.2.1:** This use case tries to benefit from energy price incentives offered by national or European authorities in particular areas where single DOLFIN-enabled DCs are located. Thus, the relevant test objective is similar to "TO5", with the semantic difference that incentives are now provided from authorities instead of the EP.

- **UC 3.2.2 and 3.2.3:** These use cases are based on synergetic DCs' activities to exploit energy price incentives offered in the area they belong to by national or European authorities. "TO6" covers these cases, with the semantic difference that incentives are now provided from authorities instead of the EP.

- **UC 3.3.1:** This use case assumes that a single DC can further optimize its efficiency by participating in a Smart Cities initiative, and can take advantage of the incentives provided as part of such a project. Thus, the relevant test objective is "TO5".

- **UC 3.3.2 and 7.3:** These use cases assume synergetic DCs participating in Smart Cities initiatives, similarly to the single DC in UC 3.3.1. The relevant test objective is "TO6".

The following table (Table 3) summarizes the test objectives identified above, indicating the relevant measurements and involved UCs, following the use case and metric terminology adopted in D2.1 and repeated in Table 1 – DOLFIN Use Cases

and Table 2 respectively.

| Test Objective ID | Relevant Measurements ID | Use Case ID |
|---|---|---|
| **TO1** | A.1, A.1.1, A.1.1.1, A.1.1.2, A.1.1.3, A.1.1.4, A.1.1.5, A.1.1.6, A.1.1.7, A.1.1.8, A.1.1.9, A.2, A.2.1, D.1, D.2 | UC.1.1 |
| **TO2** | A.1, A.1.1, A.1.1.1, A.1.1.2, A.1.1.3, A.1.1.4, A.1.1.5, A.1.1.6, A.1.1.7, A.1.1.8, A.1.1.9, A.1.2, A.2, A.2.1, B.1, D.1, D.2 | UC.1.2, UC.1.3 |
| **TO3** | A.1, A.1.1, A.1.1.1, A.1.1.2, A.1.1.3, A.1.1.4, A.1.1.5, A.1.1.6, A.1.1.7, A.1.1.8, A.1.1.9, A.2, A.2.1, B.2, D.1, D.2 | UC.2.1.1 |
| **TO4** | A.1, A.1.1, A.1.1.1, A.1.1.2, A.1.1.3, A.1.1.4, A.1.1.5, A.1.1.6, A.1.1.7, A.1.1.8, A.1.1.9, A.1.2, A.2, A.2.1, B.1, B.2, D.1, D.2 | UC.2.1.2<br>UC.2.1.3 |
| **TO5** | A.1, A.1.1, A.1.1.1, A.1.1.2, A.1.1.3, A.1.1.4, A.1.1.5, A.1.1.6, A.1.1.7, A.1.1.8, A.1.1.9, A.2, A.2.1, B.2, C.4, C.4.1, | UC.2.2.1, UC.3.2.1, |

| | | |
|---|---|---|
| | D.1, D.2 | UC.3.3.1 |
| **TO6** | A.1, A.1.1, A.1.1.1, A.1.1.2, A.1.1.3, A.1.1.4, A.1.1.5, A.1.1.6, A.1.1.7, A.1.1.8, A.1.1.9, A.1.2, A.2, A.2.1, B.1, B.2, C.4, C.4.1, D.1, D.2 | UC.2.2.2, UC.2.2.3, UC.3.2.2, UC.3.2.3, UC.3.3.2, UC.7.3 |
| **TO7** | A.1, A.1.1, A.1.1.1, A.1.1.2, A.1.1.3, A.1.1.4, A.1.1.5, A.1.1.6, A.1.1.7, A.1.1.8, A.1.1.9, A.2, A.2.1, B.2, D.1, D.2, D.3 | UC.2.3.1 |
| **TO8** | A.1, A.1.1, A.1.1.1, A.1.1.2, A.1.1.3, A.1.1.4, A.1.1.5, A.1.1.6, A.1.1.7, A.1.1.8, A.1.1.9, A.1.2, A.2, A.2.1, B.1, B.2, D.1, D.2, D.3 | UC.2.3.2<br>UC.2.3.3 |
| **TO9** | A.1, A.1.1, A.1.1.1, A.1.1.2, A.1.1.3, A.1.1.4, A.1.1.5, A.1.1.6, A.1.1.7, A.1.1.8, A.1.1.9, A.2, A.2.1, B.2, C.1, C.1.1, C.2, C.3, D.1, D.2 | UC.3.1.1 |
| **TO10** | A.1, A.1.1, A.1.1.1, A.1.1.2, A.1.1.3, A.1.1.4, A.1.1.5, A.1.1.6, A.1.1.7, A.1.1.8, A.1.1.9, A.1.2, A.2, A.2.1, B.1, B.2, C.1, C.1.1, C.2, C.3, D.1, D.2 | UC.3.1.2<br>UC.3.1.3 |

Table 3 – The DOLFIN test objectives, followed by relevant measurements and involved UCs

Moreover, the following Venn diagram represents the relationship between the various testing objectives and the testing measurements they are derived from.

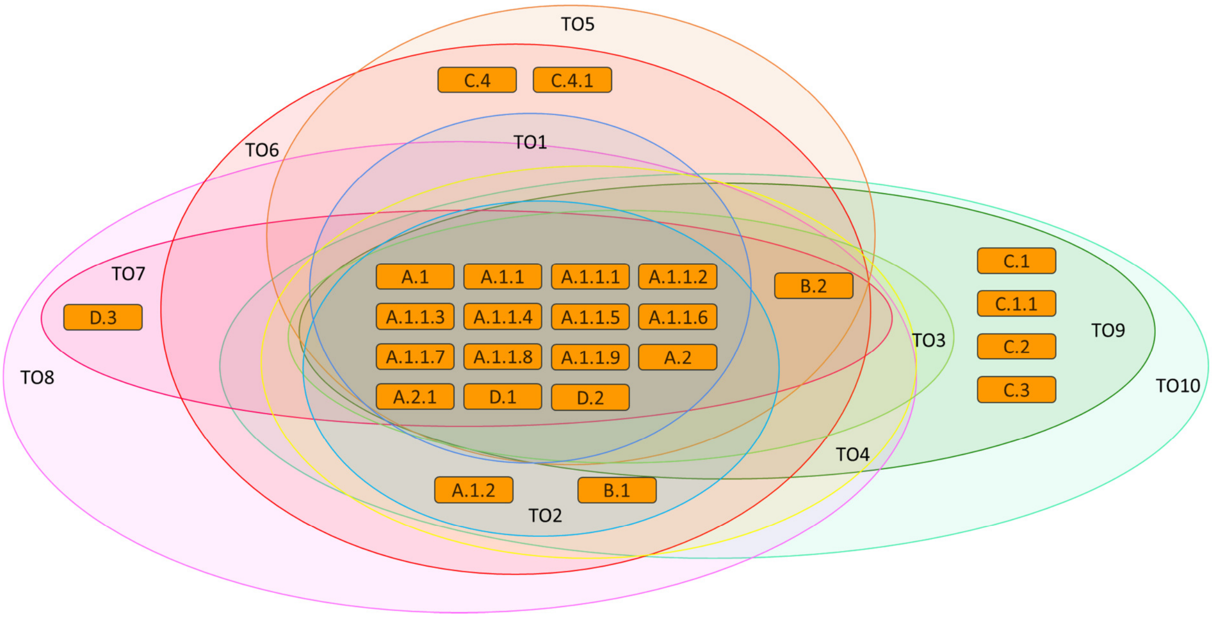


Figure 2-1: Venn diagram depicting the composition of testing objectives by testing measurements.

Thus, the resulting set of testing objectives is:

- **Reduce operational costs respecting SLAs (without shifting):** This test objective illustrates a reduction of total operational costs taking only a single DC into account with local re-organization actions and the SLAs signed with the users. As there are no shifting operations, only the most fundamental measurements need to be taken into account.
- **Reduce operational costs respecting SLAs (space-shifting):** For scenarios that try to take advantage of variations in operational costs (tariffs, incentives etc.) between DCs in different geographical areas, this test objective contains measurements.
- **Reduce operational costs respecting SLAs (time-shifting):** Scenarios that test the workload time-shifting capabilities of DOLFIN can define their outcome in terms of the following metric.
- **Reduce operational costs respecting SLAs (with shifting):** As can be easily discerned, all above T.O.s are special cases of T.O.4, which takes into account all relevant measurements for all optimization actions available to different scenarios.
- **Reduce operational costs using Smart Grid/Smart Cities incentives:** Scenarios that take into account incentives offered by either Energy Providers through the Smart Grid or by authorities through Smart Cities initiatives can make use of this TO, that also takes into account such available incentives and how they are used by DOLFIN to reduce operational costs.
- **Reduce operational costs utilizing SLA renegotiation:** Outcomes of scenarios that use SLA renegotiation with customers have to contain the relevant D.3 metric.
- **Enable green computing adoption while reducing operational costs:** Finally, environmentally friendly infrastructures take into account “green” measurements in order to attract an increasingly green-conscious segment of the Cloud market. This test objective takes such measurements into account and can be used to define outcomes of green test scenarios.

# 3. Testing scenarios

This section details the testing scenarios foreseen for the validation of the DOLFIN platform. Testing scenarios constitute testing activities under the perspective of hypothetical stories to assist end-to-end testing of a system. Formally, a test scenario features five key characteristics; it is a) a complete story that is b) motivating, c) credible, d) complex and e) easy to evaluate [5]. In this framework, a DOLFIN testing scenario indicates a situation to which a DOLFIN platform instantiation should adapt, in order to preserve optimality of the energy consumption of its composing (standalone or synergetic) DCs, by following a non-probabilistic series of actions and producing a verifiable outcome, indicating that the DOLFIN subsystems all interworked as they should and the energy consumption is minimized at aggregate DOLFIN level (depending on the active policies of the DOLFIN DCs supported).

In the following subsections the three main testing scenarios will be outlined:

- Intra DC optimization testing scenario, which evaluates DOLFIN capabilities in the context of a single DC. In this scenario the DC will be considered as a "solo eco-system" able to react to internal changes to reach the optimal energy consumption.
- SLA testing scenario, which introduces the capabilities related to DOLFIN inter-DCs cooperation in a federated DC's group, to share resources and reach the optimal energy consumption, while preserving the contractual SLA with the customers. The objective is to move the VMs between DCs while respecting specific SLA constraints.
- Smart Grid testing scenario, which is used to evaluate the DOLFIN DC adaptation capability when integrated in a Smart Grid environment. In this case the DC energy optimization logic could be directly affected by the information provided by the Utilities through the Smart Grid interfaces. The testing scenario is used to evaluate how DOLFIN reacts over changes to energy costs or energy availability.

## 3.1. Intra DC optimization testing scenario

### 3.1.1. Scenario description

This scenario tests the basic self-adaptation and optimization capabilities of DOLFIN in the context of single DC operation. Therefore, it is relevant to business objectives A and D. Although DOLFIN is designed to support networks of synergetic DCs operating in concert, a DOLFIN instantiation should be able to be deployed and fully tested in a single DC scenario, providing the means for:

- Monitoring and managing IT and non-IT infrastructures
- Accessing user information and providing accounting/billing services
- Monitoring the power efficiency of the DC, as calculated by means of sets of well-defined measurements, and
- Optimizing the operation of the DC in terms of energy consumed

In order to demonstrate the functionality of all DOLFIN components involved in this setup, a simple scenario will be outlined in the following paragraphs taking into account both user-initiated and DOLFIN-initiated actions. The user-initiated actions are used to facilitate the testing scenario and include the insertion of artificial load to the DC infrastructure. DOLFIN actions refer to the asynchronous actions initiated by the various DOLFIN components.

The basic scenario description is the following:

> The DOLFIN optimization policy is set to minimize the energy consumption of the DC in absolute terms. At a certain time, the load of a particular set of VMs running on different servers is rapidly and unexpectedly increased (but can be accommodated by the DC itself). After one hour, the load is reduced to normal levels. The DOLFIN platform should identify the load changes and reconfigure the DC load allocation to the servers/racks/rooms so that in both cases its energy consumption is as minimal as possible.

### 3.1.2. Use cases addressed

This testing scenario correlates to all uses cases that pretend the DC energy state optimization, although the main use case addressed is UC 1.1. The rest of the already identified use cases are only implicitly relevant to this testing scenario. The interested reader is advised to refer to [4] for further details on the identified DOLFIN use cases.

### 3.1.3. Test Components & Requirements

The DOLFIN components under evaluation in this scenario are primarily the core components composing eCOP, namely:

- ICT Performance & Energy Supervisor
- Energy efficiency policy maker and actuator
- eCOP Monitor DB

In addition to the eCOP core components, the various DCO Brokers interfacing the underlying infrastructures are subject to be evaluated as well, so as to demonstrate the ability of DOLFIN to integrate various virtualization and cloud management systems:

- DCO Hypervisor Manager
- DCO Monitor/Collector
- DCO Appliance Manager
- Monitoring Backend

Last, the SLA Renegotiation Controller is required to interface with the Energy efficiency policy maker and actuator group of components.

The formal requirements to perform the test scenario in hand are summarized as follows:

1. A DOLFIN-compatible cloud management platform (DOLFIN prototypes will initially support UCL's VLSP – see paragraph 5.1.1– and Openstack [6]) managing the DC resources. In this particular scenario, DOLFIN is assumed to be coordinating an OpenStack or OpenStack API-compatible installation.
2. A DOLFIN instantiation comprising all relevant components identified as test components.
3. Proper monitoring equipment should have been deployed to monitor the performance and characteristics of the DC elements of interest, including physical servers, server racks, DC rooms, HVAC equipment and lighting.

### 3.1.4. Setup

In order to the test to be successful, an operating DOLFIN instantiation applying the specific testing scenario, the following setup configuration should be performed:

- A Vanilla OpenStack installation is considered, so a default OpenStack Icehouse configuration (or other compatible implementation respecting OpenStack Compute API v2.0 [12] and above) is assumed, according to the official guides, e.g. [7]. Following the most minimalistic approach, we assume that at least the Identity (Keystone), Compute (Nova), Network (Neutron) and Telemetry (Ceilometer) services are to be configured. The Image (Glance) and Block (Cinder) services will most likely need to be configured in order to ensure proper cloud operation, but their configuration is not required to perform the testing scenario in hand.
- The rest of the eCOP components should be configured according to the instructions provided in D3.3 (ICT Performance and Energy Supervisor component – Implementation) and D3.4 (Energy Efficiency Policy Maker and Actuator component – Implementation), to be delivered in M27 and M28 of the project.
    - For the sake of clarity and test validation, the eCOP Monitor DB is assumed to be configured in such way that the eCOP DB Broker exposes at least the following API endpoints:

        *API 1. /api/measurements/by-type/{type}/{start}/{end}/*

        *API 2. /api/measurements/by-resource/{resource}/{resource_id}/{start}/{end}/*

*API 3. /api/measurements/aggregate/energy/{equipment_type}/{start}/{end}/*

where a {*type*} is a measurement type (e.g. power, energy, cpu_util etc.), a {*resource*} is a measurable resource (e.g. vm, server, rack, metric etc.), a {*resource_id*} is the id of the specific measurable resource (e.g. a vm UUID, a server serial number, a metric name etc.). In API 3, the {*equipment_type*} should be either IT (i.e. servers), or non-IT or total, depending on the equipment of interest. Last, {*start*} and {*end*} are used to define the time range of the query, expressed in ISO8601 format. More information on the configuration and capabilities of eCOP DB Broker will be given in D3.3.

- The DCO Brokers should be configured to properly mediate the OpenStack installation, HW equipment and the eCOP DB components.

The scenario would be configured with the following well-known starting condition so that the final result will be predictable. The following information correspondences are assumed to be available:

#### 3.1.4.1. *USER-VM Mapping (eCOP DB)*

| VM_ID | User_ID |
|---|---|
| VM1 | User1 |
| VM2 | User1 |
| VM3 | User1 |
| VM4 | User1 |

**Table 4 – User VM mapping (eCOP DB) for testing scenario 1**

#### 3.1.4.2. *Green server mapping (eCOP DB)*

| Server_ID | Is_green |
|---|---|
| Server1 | TRUE |
| Server2 | FALSE |

**Table 5 – Green server mapping (eCOP DB) for testing scenario 1**

#### 3.1.4.3. *VM action info (eCOP DB)*

| VM_ID | Server_ID | Event | Timestamp |
|---|---|---|---|
| VM1 | Server1 | Started | 11/6/2015 08:30:01 |
| VM2 | Server1 | Started | 11/6/2015 08:30:23 |
| VM3 | Server2 | Started | 11/6/2015 08:30:50 |
| VM4 | Server2 | Started | 11/6/2015 08:32:54 |

**Table 6 – VM action information (eCOP DB) for testing scenario 1**

#### 3.1.4.4. *SLA contract info (DOLFIN DB)*

In this scenario the contractual SLA should not to be considered, in other words, the initial setup should be such that constraints are not applied.

| User_ID | VM_ID | Metric | Condition | Value | TimePeriod | Penalty |
|---|---|---|---|---|---|---|
| User1 | VM1 | ShutdownSecs | LESS_THAN | 15 | MONTH | A |
| User1 | VM2 | ShutdownSecs | LESS_THAN | 20 | MONTH | C |
| User1 | VM3 | ShutdownSecs | MORE_THAN | 20 | MONTH | D |
| User1 | VM4 | NotOnGSSecs | MORE_THAN | 26 | MONTH | E |

**Table 7 – SLA contractual information (eCOP DB) for testing scenario 1**

### 3.1.5. Detailed test steps

DOLFIN should be able to automatically optimize the DC state to match the DC load and minimize the energy consumption. The test actions can either be user-initiated (by for example, the insertion or removal of an artificial DC load in the interest of the testing) or DOLFIN-initiated. Also, as DOLFIN operates on an asynchronous basis and chained component reactions are expected to be observed.

| Step # | Test Action | Expected Results | Means of verification [1] |
|---|---|---|---|
| 1 | Set the current DOLFIN optimization policy to require absolute DC energy consumption minimization. Uses the API from Policy Repository to enable the right policy. | The current optimization policy of the DC should be set to absolutely minimize the DC energy consumption | The policy related to "absolute DC energy consumption minimization" should be reported as "enabled" |
| 2 | Insert artificial load to at least 4 VMs running on different servers, for at least one hour | DOLFIN monitoring infrastructure indicates that the energy consumption of the specific servers is increased | The response of a GET request to either API 1 or API 2 indicates that the energy consumption of the specific servers has increased |
| 3 | The energy metrics of the specific servers indicate a significant change | The Metrics Engine detects that the DC would benefit from VM placement optimization for the time slot referring to the last 1 hour | The GET request to eCOP DB API reports that the value from performing DC optimization has increased following the introduction of artificial load on VMs. |
| 4 | The Prediction Engine identifies that a significant change in the predicted load has occurred | The Prediction Engine notifies the Policy Maker of the expected load increase | A relevant entry in the DOLFIN Info DB exists, indicating that the Policy Maker has been informed of the notification from the Prediction Engine |
| 5 | The Policy Maker triggers the Policy Repository to get the current DOLFIN DC optimization policy | The Policy Repository should respond that the current DC optimization policy is set to absolutely optimize the energy consumption of the DC | A relevant entry in the DOLFIN Info DB exists, indicating that the Policy Maker has loaded the policy that represents the "absolute DC energy consumption minimization" |
| 6 | The Policy Maker notifies the | The Optimizer receives a notification from the Policy Maker | A relevant entry in the DOLFIN Info DB exists, indicating that |

[1] Throughout this test scenario, API 1 and API 2 refer to the eCOP DB Broker API endpoints briefly presented in public deliverable D3.1

| | | | |
|---|---|---|---|
| | Optimizer to optimize the DC state | to re-organize the DC load to minimize energy consumption | the Policy Maker has notified the Optimizer to re-organize the DC load |
| **7** | The Optimizer requests the VMs status and their SLA compliance from the VM Priority Classifier | The VM Priority Classifier should contact the SLA Renegotiation controller and provide a relevant consolidated input to the Optimizer | The input to the SLA renegotiation controller includes the VNs status and SLA governance and compliance details |
| **8** | The Optimizer calculates a new VM allocation and notifies the Policy Actuator | The Policy Actuator receives a new VM allocation to implement | A relevant entry in the DOLFIN Info DB exists, indicating that the Optimizer has notified the Policy Actuator to implement the new VM allocation |
| **9** | The Policy Actuator implements the new VM allocation | The new VM allocation should be implemented | Compare the VM allocation proposed by the Optimizer to the final allocation achieved by the Policy Actuator actions |
| **10** | Check the energy consumption | The energy consumption of the DC after the DOLFIN operations is less or at most equal to the one before the operations | The response of two GET requests to API 3 should indicate that the energy consumption of the specific servers has decreased |
| **11** | Check the metrics after 1 hour (to assure the convergence of KPIs) | After a timeslot of 1 hour the Metrics Engine calculates the energy consumption benefit from performing an optimization. | The GET request to eCOP DB API reports a decreased value for optimization relevant to the calculated value in step 3. Intuitively, values closer to 0 indicate energy consumption is near to the power baseline. |
| **12** | After one hour, when the increased load stops, DOLFIN optimizes the DC state again | Steps 3-10 are repeated. | Steps 3-10 are repeated. |

Table 8 – Green server mapping (eCOP DB) for testing scenario 1

### 3.1.6. Measured objectives and relation to requirements

The successful completion of all test steps indicates that:

- All involved components eCOP components are able to communicate and interwork;
- DOLFIN is able to identify DC load changes and react upon them;
- DOLFIN is able to identify optimal states of DC operation as dictated by an administrator-set DC optimization policy without breaking the SLAs of the users;
- DOLFIN is able to manage VMs to accomplish the optimal allocation plans produced by the Energy efficiency policy maker and actuator components;

The particular testing scenario is related to UC 1.1 and TO 1, thus being subject to the A.1, A.1.1, A.1.1.1, A.1.1.2, A.1.1.3, A.1.1.4, A.1.1.5, A.1.1.6, A.1.1.7, A.1.1.8, A.1.1.9, A.2, A.2.1, D.1 and D.2 measurements (see section 2 and Appendix A for details and discussion). These measurements are specific evidence of compliance of business objectives A. Optimization of energy consumption and D. Ensuring DC user experience by defining and re-negotiating SLAs.

The relevant project aggregated requirements can be thought of as the specific demonstrable steps for satisfying the use case under examination in this testing scenario. The project aggregated requirements (as defined in D2.2) satisfied by this testing scenario include:

| Requirement ID & Name | Requirement Description |
|---|---|
| **Q7 MonitorParams** | The monitoring subsystem shall always monitor DC parameters. These requirements identify the necessity to make a formal collection for all parameters that have to be monitored to build KPI and other triggers levels used to evaluate the DC performance and schedule the appropriate policies' actions. The list of parameters to be published should be configurable in terms of frequency of measurements and way to deliver. Also, the access to this functions should be controlled. |
| **Q8 MonitorTranslate** | The monitoring subsystem shall always translate parameters to metrics. |
| **Q9 MonitorInterface** | The monitoring subsystem shall be able to present metrics-information to the management subsystem. This interface is considered to expose in a normalized form a number of metrics and their frequency. It could receive the metrics to retrieve, their frequency and response format (e.g. publish to a message broker). The system shall monitor the DC infrastructure: IT devices, power units, other DC facilities that might be involved in the process of energy optimization. |

| | |
|---|---|
| **Q10 MonitorVariousData** | The monitoring subsystem shall handle many kinds of data and with different nature (i.e. performance information, energy consumption, power dissipation, VM status and deployment, semi-static parameters, etc). This function will complement in terms of Data the other peer Monitoring interfaces. |
| **Q11 MonitorResilience** | The monitoring subsystem should be as much resilient as possible (i.e. implement heartbeat, watchdog functions). |
| **Q12 MonitorOpenDataFmt** | The monitoring subsystem should utilize open data formats is available. |
| **Q13 DCManagement** | The system shall include a management entity for orchestrating the whole process. |
| **Q20 DCVMHandling** | The PolicyEnforcement subsystem shall be able to manipulate VMs (migrate them from server to server, from DC to DC, shutdown, etc.) |
| **Q56 DCVMHandling (VM params)** | In accordance with Ask for VMParamsAdjustment" this requirement implies that the Workload and VM Manager should be able to fulfil a request for changing the parameters of a certain set of VMs". |

Table 9 – DOLFIN requirements as defined in D2.2 for testing scenario 1

## 3.2. SLA testing scenario

### 3.2.1. Scenario Description

In this scenario the ability of DOLFIN to optimize the placement of VMs within the federated DC architecture while respecting the user-defined SLAs is tested. As performance is important for proper cloud operation, strong guarantees should be provided by the DC operator so as not to adversely affect the end-user Quality of Experience (QoE) in the attempt to minimize power consumption. DOLFIN considers user-defined SLAs as a guide to expected performance from an end-user standpoint when optimizing DC power consumption.

To this end, a set of periodic actions under normal operating conditions is simulated, in order to examine the number and performance of all active DC VMs (in terms of SLA compliance). Once the process of matching SLA requirements to DC availability is complete, a plan for energy optimization is produced, which is the expected outcome of this test scenario. The basic scenario description is the following:

| |
|---|
| The SLA Renegotiation Controller (SLARC) will receive regular updates on the states of all existing VMs, in order to assure that the SLA requirements are met. In order to simulate |

an outage or a system failure, the following actions will be performed: (i) Unexpectedly one or more VMs will be shut down; and (ii) some of the remaining running VMs will be moved to non-Green Servers and have their allowed hosting time on non-green servers set to expire at various random dates in the future.

SLARC will provide to the VM Priority Classifier a list which will contain for the VMs, among other information, the time left until a violation of SLAs will occur -- for this scenario cases (i), and (ii). In accordance to these data values, the Optimizer will define a set of actions needed to fulfil the SLA requirements.

Also, SLARC will asynchronously generate an SLA violation event onto the Rabbit MQ bus when a violation of an SLA has occurred (e.g. as a result of an unexpected VM shutdown).

### 3.2.2. Use Cases addressed

This scenario is primarily related to TO8 and, secondarily, to TO1, TO4, TO7 and TO9 thus pertaining to UC.1.1, UC.2.1.2, UC.2.1.3, UC.2.3.1, UC.2.3.2, UC.2.3.3, UC.3.1.1 (see section 2 for details and discussion).

### 3.2.3. Test Components & Requirements

The DOLFIN components under examination in this scenario are:

- ICT Performance & Energy Supervisor
- Energy efficiency policy maker and actuator
- eCOP Monitoring DB
- SLA Renegotiation Controller
- DOLFIN DB
- Cross-DC Orchestrator

### 3.2.4. Setup

As this scenario also involves intra-DC optimization, all relevant elements from the Setup of the first testing scenario (paragraph 3.1) are required. Also, a number of active VMs in the DOLFIN enabled DC are assumed whose SLAs, billing information and current placement in the DC are known. Moreover, all relevant monitoring information is available. Last, the following information are assumed to be available:

#### 3.2.4.1. *USER-VM Mapping (eCOP DB)*

| VM_ID | User_ID |
|---|---|
| VM1 | User1 |
| VM2 | User3 |
| VM3 | User1 |
| VM4 | User1 |
| VM5 | User3 |
| VM6 | User2 |

**Table 10 – User VM mapping (eCOP DB) for testing scenario 2**

#### 3.2.4.2. *Green server mapping (eCOP DB)*

| Server_ID | Is_green |
|---|---|
| Server1 | TRUE |
| Server2 | FALSE |
| Server3 | FALSE |

**Table 11 – Green server mapping (eCOP DB) for testing scenario 2**

#### 3.2.4.3. *VM action info (eCOP DB)*

| VM_ID | Server_ID | Event | Timestamp |
|---|---|---|---|
| VM2 | Server1 | Create | 11/6/2015 08:30:01 |
| VM3 | Server2 | Started | 11/6/2015 08:30:23 |
| VM2 | Server1 | Started | 11/6/2015 08:30:50 |
| VM1 | Server2 | Stopping | 11/6/2015 08:32:54 |

| | | | |
|---|---|---|---|
| VM1 | Server2 | Stopped | 11/6/2015 08:33:10 |
| VM1 | Server2 | Deleting | 11/6/2015 08:33:11 |
| VM1 | Server2 | Deleted | 11/6/2015 08:33:55 |

**Table 12 – VM action information (eCOP DB) for testing scenario 2**

#### 3.2.4.4. *SLA contract info (DOLFIN DB)*

| User_ID | VM_ID | Metric | Condition | Value | TimePeriod | Penalty |
|---|---|---|---|---|---|---|
| User1 | VM1 | ShutdownSecs | LESS_THAN | 15 | MONTH | A |
| User1 | VM1 | ShutdownSecs | LESS_THAN | 20 | MONTH | C |
| User1 | VM1 | ShutdownSecs | MORE_THAN | 20 | MONTH | D |
| User1 | VM1 | NotOnGSSecs | MORE_THAN | 26 | MONTH | E |
| User3 | VM5 | AVGShutdownSecs | MORE_THAN | 50 | WEEK | A |

**Table 13 – SLA contractual information (eCOP DB) for testing scenario 2**

Since the interaction of possibly more than one DC is assumed, all the relevant entries and configurations are assumed to hold of all Synergetic DCs involved in the scenario.

Given the above setup, at a certain time $t_0$, a message to reduce DC energy consumption is emitted by the eCOP prediction engine, due to a predetermined trigger condition. Although for this test the condition is not important, the following conditions could act as triggers:

1. Due to temperature decrease forecast, you may take advantage of the cool air.

2. Due to cloudy day, the PV is expected to produce less electricity.

3. Do periodic energy optimization check.

### 3.2.5. Detailed test steps

| Step # | Test Action | Expected Results | Means of verification |
|---|---|---|---|
| **1** | eCop Prediction Engine notifies the Policy | Policy Maker receives message and initiates the optimization | When the Policy Maker has been informed of the |

| | | | |
|---|---|---|---|
| | Maker. | process. | notification from the Prediction Engine a log is saved in DOLFIN info DB. |
| **2** | Policy Maker notifies the optimizer. | Optimizer is notified. | The policy that represents the "optimize energy performance at Synergetic DCs level" should be reported as "enabled" |
| **3** | The Optimizer requests input from the VM Priority Classifier. | The VM Priority Classifier gets a relevant request. | The Optimizer's request is logged in the DOLFIN Info DB, along with the VM Priority Classifier request. |
| **4** | The VM Priority Classifier requests a list of VMs and their status from the SLA Controller. | The SLA Controller which consumes monitoring data looks into its own data store for VM info. | A relevant log including the VM Priority Classifier request is posted in the DOLFIN Info DB. |
| **5** | The SLA Controller replies to the VM Priority Classifier with the list of VMs. | The VM Priority Classifier receives the message. | When the VMs list is received a log in the DOLFIN Info DB is saved. |
| **6** | The VM Priority Classifier enriches the list with more information (e.g. I/O to RAM/HDD, memory size etc.) and sends the ordered list to the Optimizer. | The Optimizer receives the list. | The enriched list of VMs received by the Optimizer will be logged in the DOLFIN Info DB. |
| **7** | The Optimizer makes a plan to optimize the DCs energy efficiency. | The Policy Maker may activate a request to the "Cross DC Workload Orchestrator" if the necessary eCOP conditions are true. | All relevant logs are posted in the DOLFIN Info DB with the indication of the original request to the Cross DC Workload Orchestrator and the result of the operation. |
| **8** | The Cross DC Workload Orchestrator polls remote DCs. | The CDC Workload Orchestrators at other sites respond with availability status. | The indication of the original request to the Cross DC Workload Orchestrators and the result of the operation will be logged in the DOLFIN |

| | | | |
|---|---|---|---|
| | | | Info database. |
| **9** | The Cross DC Orchestrator contacts the Cross DC VM Manager to initiate the VM relocation. | On completion of the VM migration the Cross DC VM Manager responds with the status. | The Cross DC VM Manager response and the VM migration status are saved in a log posted in the DOLFIN Info DB. |
| **10** | Asynchronously, during the migration action, the SLA for the VM being migrated is violated. The SLARC detects the violation and notifies the Policy Maker with the SLA details of the VM. | The Policy Maker checks the VM UUID against the VMs in the plan, as violation may occur during migration.<br><br>If the relevant VM is in the plan being implemented, then the notification should be ignored. | The violation event generated by the SLARC will be saved as a log in the DOLFIN Info DB, along with the Policy Maker response. |
| **11** | If the VM is not in the plan, the Policy Maker takes steps to respect the VM SLA (e.g. VM resume). | If a new plan is necessary, steps 2-9 are repeated to resolve the violation. | The Policy Maker response will be logged in the DOLFIN Info DB. |
| **12** | At any time during the operation of the DC, the SLARC may asynchronously generate a violation event | A violation event will be sent onto Rabbit MQ bus to be read by interested components. | The violation event will be logged into the DOLFIN Info DB, along with the SLARC violation event. |

Table 14 – Test steps for testing scenario 2

### 3.2.6. Measured objectives and relation to requirements

A successful test scenario completion indicates that all measurements bound to the relevant TOs, namely TO1, TO4, TO7, TO8 and TO9 are tested, that is measurements A.1, A.1.1, A.1.1.1, A.1.1.2, A.1.1.3, A.1.1.4, A.1.1.6, A.1.1.7, A.1.1.8, A.1.1.9, A.1.2, A.2, A.2.1, B.1, B.2, C.1, C.1.1, C.2, C.3, D.1, D.2 and D.3. In other words, this test encompasses all DOLFIN business objectives (A, B, C and D) as described in Section 2. In short, the above are summarized as follows:

- All involved eCOP components are able to communicate and interwork;
- All involved SDC components are able to communicate and interwork;
- The eCOP is able to coordinate with the SDC.

- DOLFIN is able to identify DC state changes and react upon them;
- DOLFIN is able to identify optimal states of DC operation as dictated by an administrator-set DC optimization policy without breaking the SLAs of the users, even if this involve SLA renegotiation;
- DOLFIN is able to manage VMs to accomplish the optimal allocation plans produced by the Energy efficiency policy maker and actuator components;
- DOLFIN is able to manage VMs at a cross-DC level in order to accomplish the optimal allocation plans produced by the Energy efficiency policy maker and actuator components;

The relevant project aggregated requirements as defined in D2.2 include:

| Requirement ID & Name | Requirement Description |
|---|---|
| **Q5 MgmntPolicyDecision** | The management subsystem shall automatically identify the actions that could be taken to optimize the DC energy state. |
| **Q6 MgmntSLARenegotiation** | The management subsystem shall be able to (automatically) request the SLA renegotiation to better adapt the SLA KPI to current optimal assets. |
| **Q51 Ask for RenegotiationPolicy** | The Energy Eff. Policy Maker and Actuator should be able to issue a request of the Renegotiation Policy (e.g. renegotiation of Workload and VM Manager to shift computing load (postpone it for later) when such a need arises, when this load shifting could result in substantial cost reduction due to electricity price or it is considered to be profitable in the case of an end-user SLA renegotiation process, or such a request arrives from a DCO / Smart City operator. |
| **Q37 DC2BInterface MgmntSubOptimalDetection** | The management subsystem shall detect DC suboptimal states (i.e. high KPI) by utilizing current DC metrics and defined thresholds. |
| **Q57 HandleExceptions** | When a customer denies a SLA renegotiation proposal or a synergetic DC denies cooperation, the DCO should be able to handle the rejection and consider alternative self-optimization actions |
| **Q58 LogRenegotiationSteps** | The SLA Renegotiation Controller should be able to persist the SLA renegotiation steps when they occurs. |
| **Q66 MonitorSLA Requests** | The SLA Renegotiation Controller should keep track of the SLA Renegotiation requests for historical reasons. |
| **Q67 NegotiateSLAUpdate** | When required to do so (e.g. by the MgmntSLARenegotiation requirement), the SLA Renegotiation Controller should be able to actually perform the renegotiation |
| **Q69 LogReplanningSteps** | This log should support development for Monitoring, Analytics, Audit and Security. It can be implemented as a |

| | |
|---|---|
| | utility package for Management of Replanting of energy scheme. |

Table 15 – DOLFIN Requirements as defined in D2.2 for testing scenario 2

## 3.3. Smart Grid testing scenario

### 3.3.1. Scenario description

DOLFIN is designed to monitor a large number of metrics, assisting towards an environmentally friendly computing infrastructure. To this end, efficient use of power provided from the Power Grid is a key to the project objectives. Interfacing with the Smart Grid so as to receive relevant information from the DC energy providers enables DOLFIN to increase power savings while at the same time reducing costs, as DOLFIN can take measures in response to Smart Grid notifications, such as price variations, renewable mix information and Grid usage statistics.

In this scenario we envisage to test DOLFIN behaviour against a series of common Smart Grid events and measure relevant KPIs that highlight the increased efficiency of DOLFIN-enabled DCs. In DOLFIN we integrate the Open Automated Demand Response (OpenADR) [8] protocol to test our approach. OpenADR is a communications protocol designed to facilitate transmission and reception of demand-response signals from a utility or independent system operator to electricity customers. The DOLFIN Smart Grid Controller (SGC) acts as a gateway, receiving demand-response events from a Demand Response Automation Server and converting them into information for the Energy Efficiency Policy Maker and Actuator. The basic scenario description is the following:

| |
|---|
| The ADR Server will provide to the SGC daily information about the energy prices in one hour intervals. At random points during the day, the electricity prices will be adjusted to reflect a new situation in the grid generation facilities and the ADR Server will inform the SGC of these changes. The SGC will inform the Policy Maker to re-evaluate the energy consumption in DC and if possible to take advantage of the energy prices or to reduce the energy demand. |

### 3.3.2. Use cases addressed

This scenario primarily relates primarily to TO5 and secondarily to TO2, TO3, TO4 and TO6, hence the UCs addressed are UC.1.2, UC.1.3, UC.2.1.1, UC.2.1.2, UC.2.1.3, UC.2.2.1, UC.3.2.1, UC.3.3.1, UC.2.2.2, UC.2.2.3, UC.3.2.2, UC.3.2.3, UC.3.3.2 and UC.7.3. The main objective is to show DOLFIN's responsiveness under a number of test conditions (price variation; green power availability; grid overload) without compromising the quality of the service as perceived by the user and defined in the customer SLAs.

### 3.3.3. Test Components & Requirements

The DOLFIN components under examination in this scenario are:

- Smart Grid Controller
- ICT Performance & Energy Supervisor
- Energy efficiency policy maker and actuator
- eCOP Monitoring DB
- DCO services
- HVAC/Cooling subsystem

### 3.3.4. Setup

In this scenario, we assume a number of active VMs in the DOLFIN enabled DC, or a number of VMs in the VLSP soft DC.

The ADR Server will be configured to send pricing schedules using three kinds of messages: (i) PRICE_ABSOLUTE message, which contains the hourly energy prices valid throughout an established time interval, (ii) PRICE_RELATIVE message, which contains a new energy price for a certain time period and (iii) PRICE_MULTIPLE message, used when the EP applies a multiplier to the current energy price for a specific interval. The first message is sent once a day and (ii) and (iii) at random intervals as a result of changes in the Smart Grid.

The SGC will listen on a pre-defined input interface for these messages and consume them. The SGC will build an internal calendar of prices, and send price data on a pre-defined output interface.

### 3.3.5. Detailed test steps

We will demonstrate the transmission of ADR messages from the ADR server. These messages contain a schedule with a set of prices for different time intervals.

After receiving a PRICE_ABSOLUTE message the SGC will build an internal calendar based on the startDate and endDate of the message and the pricing data held in the schedule of the message. The schedule contains different time intervals with different prices during a day.

The SGC will consume these pricing messages and build an internal calendar for each hour of each day in the schedule, such as the one shown below:

| Day | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sun | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |
| Mon | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |
| Tue | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |
| Wed | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |
| Thu | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |
| Fri | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |
| Sat | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |

**Table 16 – Internal calendar of prices in hourly slots after receiving the PRICE_ABSOLUTE message**

After the receipt of the PRICE_RELATIVE message, which represents Monday 7th September between 18:00 and 20:00 the pricing calendar will be updated. Messages of type PRICE_RELATIVE cause the calendar price to have the price in the message added to the value. In this case 0.4 will be added to the base price of 0.2.

The resulting calendar will look like:

| Day | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sun | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |
| Mon | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.6 | 0.6 | 0.2 | 0.2 | 0.2 | 0.2 |
| Tue | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |
| Wed | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |
| Thu | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |
| Fri | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |
| Sat | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |

**Table 17 – Internal calendar showing PRICE_RELATIVE adjustment**

The SGC will periodically send the current price of electricity to the DOLFIN components that have an interest in current price information.

| Step # | Test Action | Expected Results | Means of Validation |
|---|---|---|---|
| 1. | Messages on the EP/Smart Grid queues are received by the ADR Server | The ADR server aggregates the significant events. | The Smart Grid messages will be stored in the DOLFIN Info DB. Every message will be logged in the DB. |
| 2 | The ADR server will push messages to the SGC | The SGC will consume the messages and build an internal calendar. | The ADR messages in JSON format will be posted in the DOLFIN Info DB.<br><br>The internal calendar will be exposed using a web service. |
| 3 | The SGC exposes new messages to the Policy Maker and other interested components through the SGC REST API. | The Policy Maker will consume the current energy price. | When the REST method (GET) is called a log will be stored in the DOLFIN Info DB. |
| 4 | The ADR server will push messages to the Smart Grid Controller when changes occur | The SGC will consume the messages and update its internal calendar. | A relevant entry will be logged in the DOLFIN Info DB, along with the ADR message. |
| 5 | The SGC will push immediately the high priority messages to the Policy Maker and other interested components. | Consumers of these messages will respond as necessary. e.g. the Energy Viewer of the VLSP will show the current Watts used by each VM as well as the cost for using that energy | A log will be generated when a message will be pushed via the Rabbit MQ bus. |
| 6 | The SGC notifies the Policy Maker of the request and incentives. | The Policy Maker gets from the Policy Repository the list of policies that best match the request. The policy directives and constraints are passed to the Optimizer | An entry is logged in DOLFIN Info DB with the list of policies extracted by the Policy Maker and the request sent to Optimizer |
| 7 | The Policy Maker notifies the optimizer. | The Optimizer receives the notification and initiates the optimization procedure. | A relevant entry in the DOLFIN Info DB exists, indicating that the Policy Maker has loaded the policy that represents the “optimize energy consumption level” |

| | | | |
|---|---|---|---|
| **8** | The Optimizer requests input from the VM Priority Classifier. | The VM Priority Classifier receives the relevant request. | The Optimizer's request will be logged in the DOLFIN Info DB. |
| **9** | The VM Priority Classifier requests a list of VMs and their status from the SLARC. | The VM Priority Classifier receives the list and sends it to the Optimizer. | When the VMs list is received a log in the DOLFIN Info DB is saved. |
| **10** | The Optimizer makes a plan to optimize the DCs energy consumption. | The plan is returned to the Policy Maker. | All relevant logs are posted in the DOLFIN Info DB with the indication of the original request and the result of the operation |
| **11** | The plan returned by the Optimizer is further evaluated in terms of the direct incentives | Depending on the cost of the plan, the Policy Maker informs the Smart Grid Controller.<br>(If the incentives are not enough to offset the penalties due to SLA violations, the original sub-optimal plan is selected) | Along with the Policy Maker messages, a log is saved in the DOLFIN Info DB. |
| **12** | The SGC provides appropriate responses to ADR requests, as per the instructions of the Policy Maker. | The ADR responds to the Smart Grid | The SGC response will be logged in the DOLFIN Info DB, along with the SGC messages. |

Table 18 – Test actions for testing scenario 3

### 3.3.6. Measured objectives and relation to requirements

As the scenario is tested against primarily TO5 and secondarily to TO2, TO3, TO4 and TO6, all relevant measurements are involved, including: A.1, A.1.1, A.1.1.1, A.1.1.2, A.1.1.3, A.1.1.4, A.1.1.5, A.1.1.6, A.1.1.7, A.1.1.8, A.1.1.9, A.1.2, A.2, A.2.1, B.1, B.2, B.2, C.4, C.4.1, D.1 and D.2. Therefore, all DOLFIN business objectives (A, B, C, and D) are verified by the successful execution of this testing scenario.

On test completion:

- The original trigger condition is met (e.g. 10% energy reduction) or an error is returned if the request cannot be satisfied.

- CDC VM migrations and time-shifting operations do not cause user VMs to violate their SLAs and incur no penalties.
- Potential SLA violation penalties incurred during energy optimization are offset by Smart Grid incentives to lower DC load.

The relevant project aggregated requirements as defined in D2.2 [4] include:

| Requirement ID & Name | Requirement Description |
|---|---|
| **Q17 PolicyEnfoercementInterface** | The system shall be able to enforce different power dissipation policies (i.e. using DVFS and/or ACPI functions, control HVAC equipment and also through VM migration). The PolicyEnforcement subsystem shall utilize an interface to communicate with the management subsystem that should be common across all platforms. |
| **Q18 ICTPowerControl** | The PolicyEnforcement subsystem shall be able to manage ICT devices power dissipation (i.e. DVFS, ACPI, etc.) |
| **Q19 HVACPowerControl** | The PolicyEnforcement subsystem shall be able to manage non ICT devices power dissipation (i.e. control/shutdown HVAC equipment). |
| **Q21 MgmntEPsRequests** | The system shall handle incoming EP requests and notifications about changes and trends in the energy provision service. |
| **Q22 MgmntDOLFINResponses** | The system shall be able to send feedback and notifications to the Eps. |
| **Q23 Energy Provider Emission rate** | The EPs shall be able to produce energy production statistics and forecasts |
| **Q24 Energy Provider Requests Mngmnt** | The EPs shall be able to forward requests and to consumers about status changes of the energy provisioning |
| **Q41 HVACParamsInterface** | This interface provides information to the monitor subsystem concerning power dissipation and status of HVAC equipment. |
| **Q42 ICTParamsInterface** | This interface provides information to the monitor subsystem concerning power dissipation and status of IT equipment. |
| **Q50 Ask for PostponeExecutionPolicy** | The Energy Eff. Policy Maker and Actuator should be able issue a request of the Workload and VM Manager to shift computing load (postpone it for later) when such a need arises, e.g. when this load shifting could result in substantial cost reduction due to electricity price or it is considered to be profitable in the case of an end-user SLA renegotiation process, or such a request arrives from a DCO / Smart City operator. |

| | |
|---|---|
| **Q53 DC2SmartGridInterface** | In order for the Smart Grid Controller to be able to communicate with both the Smart Grid/EPs and the DOLFIN-enabled DCs, a proper interface serving the relevant requests/responses of such communication is needed. |
| **Q54 DCHeatControl** | In case the Smart Grid Controller retrieves a demand for a heat exchange state change from the DC part, the Energy Eff. Policy Maker and Actuator should be able to control the heat exchange towards the Smart City (turn it off or on). |
| **Q62 Energy Provider Cost profile** | The Aggregators should be able to request for the tariff schemes of a certain set of EPs, through the Smart Grid Controller. |
| **Q63 MgmntCostDetection** | The system shall be able to enforce different power dissipation policies (i.e. using DVFS and/or ACPI functions, control HVAC equipment and also through VM migration). The PolicyEnforcement subsystem shall utilize an interface to communicate with the management subsystem that should be common across all platforms. |
| **Q70 MgmntCostsOpportunity** | Represents an Analytics functional package used my Management Interfaces to evaluate the impact of potential actions taking in account history logs and current state. Main result is a decision support regarding the opportunity to search and/or accept a different cost scheme and it's time based impact on managed DC. |

Table 19 – DOLFIN requirements as in D2.2 for testing scenario 3

# 4. Integration, Testing, Validation, Evaluation and Demonstration plans

This section describes the scheduling of the integration, testing and demonstration phases of DOLFIN. It also covers the validation of the DOLFIN components (either in a standalone or an integrated manner) and the evaluation of the efficiency and performance of the DOLFIN platform as a whole. The goal is to demonstrate from a functional point of view the testing scenarios detailed in Section 3 and ensure a quantitative evaluation approach regarding the performance of DOLFIN.

This section outlines the strategy that will be used to evaluate the DOLFIN readiness and ability to meet the requirements identified in D2.2 [4] including:

- Design Verification – where the development of the different modules for the eCOP and the SDC subsystems are evaluated separately.
- System Integration – where the system is evaluated as an integrated, singleton entity.
- System Validation - to be performed as required over the system in an iterative manner to overcome bugs and problems that have appeared in the integration phase. As validation guides, the set of testing scenarios detailed in Section 3 will be employed.
- System Demonstration - to be performed on the final operational DOLFIN result, to verify that the expected functionality is actually present and the system is adaptable to changes in other environment aspects (e.g., changing conditions in the DC, external temperature conditions change, massive migrations of VM that create a sudden increase in the workload of the DC, etc.).

It is expected that DOLFIN testing phases will initially produce high level tests to address the overall requirements of the testing scenarios and also a specific set of test to support a more detail evaluation of the functionality of the DOLFIN system and its components.

The testbeds that will be used in order to carry out the actual evaluation of the DOLFIN system for all the phases mentioned above is described in depth detail in Section 5 of this document and includes the testbed description that the partners (Interoute, Wind, UCL, PSNC and GRNET) will grant for the DOLFIN system integration and demonstration.

It is assumed that the integration and testing processes will be initiated when the modules have been completely developed and are ready to be tested in a standalone mode. This is depicted in Figure 4-1 where the orange box is the initial starting point. Once the key modules of the DOLFIN system have been developed, they will be tested separately in a standalone mode (unit testing), the relevant activities being represented in the light blue box (process numbered as 1). The successful unit testing will be followed by the integration of the components (with a slight overlap of the two processes), depicted as a blue box in Figure 4-1 and numbered as process number 2. The overlap of the two processes is intended and it serves the easier and problem-free integrated components testing which follows the integration and testing processes. A key part of the integration process of the whole system will be the development of the DCO Brokers that will interact with the DC system. Three DCOs have been specified by DOLFIN: DCO Hypervisor broker, DCO Monitoring and Collection broker and DCO Appliance broker that will be the interfaces between the actual DC system and the DOLFIN system; further information about them can be found in D3.1 [2] and D4.1 [3] deliverables. The development of these 3 DCO modules has already started and will continue to take place simultaneously to the finalization of the eCOP system and the SDC system.

The integrated testing (dark blue box in Figure 4-1 and numbered as process number 3) will consider parts of the testing scenarios identified in Section 3 of the document. When the integrated testing has been successfully completed, the testing scenarios will be considered to be demonstration ready, leading DOLFIN to the Validation, Evaluation and Demonstration phase.

It is worth noting that the whole process will be iterative following an agile approach, where testing will be performed in the form of component testing *sprints* followed by code reviews and bug fixing when problems arise, until all logical and coding errors are eliminated.

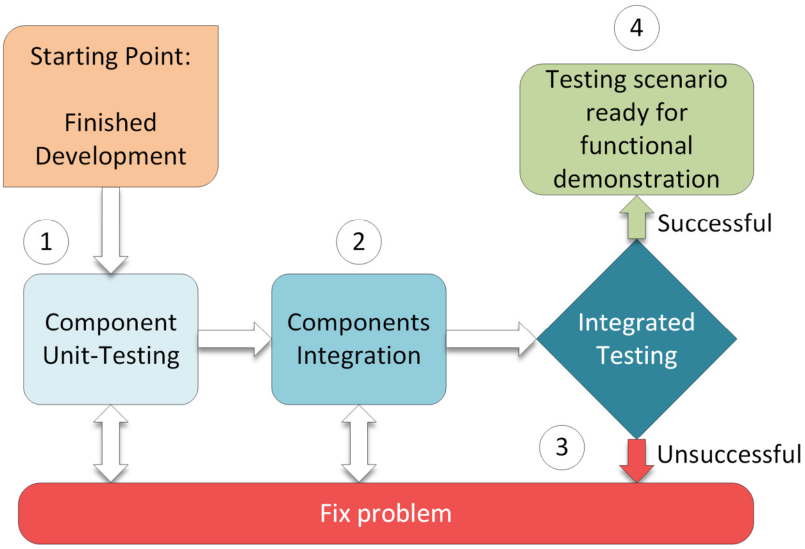


**Figure 4-1: Integration and Testing processes**

It is not expected that the testing of the DOLFIN system will require a specific platform other than the ones that the current developers use in a daily basis. The actual set up of the testing environment is defined according to the needs of each testing scenario.

The DOLFIN Consortium has agreed to commit to the phases explained in the following paragraphs. In order to avoid the slippage of the tasks, the phase of validation of the system, which effectively means, the iterative pattern described above, has been given sufficient so that there is enough time for testing the system and for gathering enough data so as to properly evaluate its performance.

The following phases have been determined for the integration, testing, validation, evaluation and demonstration of the DOLFIN subsystems (eCOP and SDC) and components.

**PHASE A: Integration (M25 - M29)**

The primary goal of the integration phase in to ensure that the various DOLFIN components are able to successfully interface and properly interwork at components, subsystems and global DOLFIN level. Moreover, the integration activities will comprise developments oriented towards ensuring that DOLFIN will be able to successfully interact with popular software platforms such as Openstack.

The integration of the various components will take place in the integration testbed of GRNET. In particular, the components developed in the context of WP3 and WP4 will be deployed in the integration testbed and their requirements will be validated. Detailed instructions on the configuration to achieve proper integration of the components will be compiled in order to facilitate the deployment of the DOLFIN platform as a whole in the demonstration infrastructures (see paragraph 5.2) as well as third parties who would like to use the DOLFIN platform either as a whole or as a collection of components exhibiting part of the overall DOLFIN functionality.

In the attempt to achieve integration with already existing DC infrastructures, the various DCO Brokers, including the DCO Hypervisor, DCO Collection and Monitoring, DCO Appliances Brokers will be developed; these are key components enabling the interaction of DOLFIN with the software infrastructures that monitor and manage the (possibly virtual) DC resources. It is envisioned that the initial versions of the DCO Brokers of DOLFIN will support Openstack (versions Icehouse and later) and VLSP (UCL Testbed detailed in paragraph 5.1.1) for the IT equipment. The non-IT equipment will be interfaced through the DCO Appliances Brokers that will acts as relay to the legacy DC's infrastructure, for the communications with facilities such as cooling system, power network, lights, etc. In the context of the testbed, the DCO Appliance Brokers would be limited to support only the standard interfaces available in the specific test environment, which could be Modbus protocol (generally used to interface to field devices such as cooling control unit and power meter), ILO and SNMP interface to collect monitoring information (for example power consumption) directly from servers or PDUs (Power Distribution Unit).

To facilitate the integration, the DOLFIN consortium has already deployed;

1. A flexible integration infrastructure, employing state-of-the art virtualization techniques, in order to create a virtual DC infrastructure able to virtualize physical servers and offer them as a service. The Virtual DC infrastructure is based on nested virtualization technology, available in a number of modern hypervisors (e.g. KVM, XEN, VMWare), and will enable testing DOLFIN at scale. It consists of a set of VMs representing an OpenStack installation on physical servers. The VMs can be placed on a single or a number of physical hosts and are then indistinguishable from an actual physical installation, as they can perform all necessary actions for a test infrastructure (e.g. spawn and manage nested VMs, provide accurate statistics etc.). The benefits of a virtual testing infrastructure are: (i) minimal physical server provisioning for tests at scale, as a physical server can represent multiple virtual infrastructure servers, up to a desirable ratio (e.g. 4 virtual servers for each physical server); (ii) easy and rapid deployment of a testbed, as the precise installation details are abstracted by the virtual servers; (iii) easy development, as a pristine installation can be always had by spawning a testbed from scratch; (iv) valid testing procedures, as a new instance of a virtual infrastructure can be guaranteed to contain no “contaminants” of previous experiments (e.g. data, code, running or latent configurations, etc.).

2. A source code versioning system employing the GIT source control system, hosted in the premises of I2CAT, in order to facilitate code lookups and sharing and enable concurrent code modifications. The tool implementing the GIT functionality is STASH, whose usage has been explained in deliverable D6.3. STASH works as a common repository providing not only the capability to store the code but also to work through the different versions of the code that has been produced and to commit changes in a distributed manner.

**PHASE B: Testing (M26 - M31)**

The purpose of the Testing phase is to ensure that the functionality exposed by the various DOLFIN components is in line with the specifications detailed in [2] and [3] at components, subsystems (namely eCOP and SDC) and DOLFIN level. To this end, specific testing procedures will be defined at components (e.g. unit tests), subsystems (also testing the intra-subsystem integration) and integrated DOLFIN level (initially targeting at the testing scenarios already identified).

In detail, the following testing activities will be carried out:

1. First, the integration of DOLFIN with the DCs and their cloud management and monitoring infrastructures will be tested.
2. Then, component-specific unit tests will be executed for each of the components in order to guarantee that software bugs (both logical and coding) are minimized and the components standalone functionality is the one intended.
3. Next, the interfacing of the components and their combined ability to act in a coordinated way towards achieving energy efficiency as described in [2] and [3] will be tested. Large parts of the scenarios identified in Section 3 of this document will be employed for the realization of these tests.

**PHASE C: Validation and Evaluation (M32-M35)**

Using the integrated and tested DOLFIN platform as a basis, this task will employ the full extent of the scenarios identified in Section 3 in order to validate and evaluate DOLFIN scope, efficiency and performance. During this phase, the consortium will focus on gathering data and translating them into meaningful insights and composite energy efficiency metrics (see [4] for details and discussion) that will allow for the exhibition of the impact of DOLFIN to the energy consumption of compatible DCs.

The validation and Evaluation of DOLFIN will allow for validated extracting information related to:

- The coverage of the project requirements
- The validation of the architecture and technological choices made
- The adequacy of the workload and VM migration processes
- The level of adaptation on the changing conditions of energy consumption within a Data Centre, as a result of the workload fluctuations
- The performance evaluation of the system in terms of accuracy, integrity, speed and flexibility
- The acceptance factor of the proposed solution on behalf of the DC owners
- An estimation of the changes that are necessary for the adoption of the solution at large, commercial scales.

Moreover, this phase will result in the definition of sets of recommendations with respect to actions that can assist toward reducing the energy footprint of a DC.

In order to maximize the utility of the evaluation results, the DOLFIN platform will operate in one or more of the demonstration testbeds, so that all efficiency and performance metrics are based on data originating from real, operational environments.

**PHASE D: Demonstration (M35 – M36)**

During this phase, the integrated, tested, validated and evaluated DOLFIN platform will be used as a basis for defining a demonstration plan that can clearly highlight the strengths and efficiency of the DOLFIN platform. The DOLFIN platform instantiation to be demonstrated will be the one identified during the validation and evaluation phase, namely, one of the operating in one or more of the demonstration testbeds.

The deliverable "D5.2 – DOLFIN system integration & evaluation" which will be due by the end of M36 will determine the milestone of the accomplishment of the demonstration phase and all the testing, validation and evaluation that will be carried out as part of the DOLFIN system.

## 4.1. Gantt chart

A visual representation of the developments intended in the context of the WP5 during the third year of the project is provided in Table 20 where 2 milestones of this phasing have been clearly highlighted: the Y2 review demonstration (that will provide an initial demonstration of the DOLFIN proof of concept together with the feedback of the project review team with respect to the validity of the project approach until now) and the last WP5 deliverable D5.2 that will resume the system integration and evaluation.

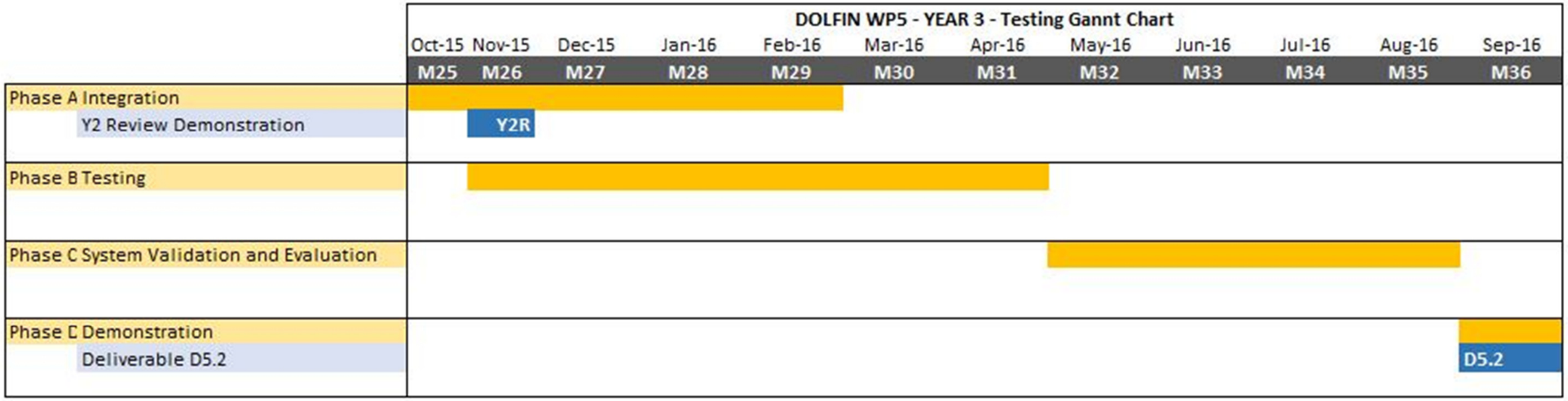


Table 20 - Gantt diagram of the DOLFIN evaluation.

## 4.2. Risk Management and Contingencies for the testing

| Risk ID | Risk Description | Likeli-hood | Impact | Risk Mitigation Measures |
|---|---|---|---|---|
| R1 | Modules development are not finished on time | M | M | To avoid the late delivery of the software, the partners have planned and worked through it during these two years. Several communication tools have allocated resources to facilitate the fast communication between partners that can be used to quickly raise problems within the consortium. Allowing fast communication amongst partners will be beneficial for the successful achievement of the DOLFIN system. |
| R2 | Lack of personnel from any of the partners dedicated to this phase | L | H | The current group of people involved in DOLFIN so far is expected to carry on with the work for the testing and evaluation phases. Every partner is expected to provide the necessary personnel in case the current involvement requires different or increased effort. |
| R3 | Lack of hardware resources dedicated in the testbeds | L | H | There are sufficient resources dedicated to the project consecution, in case there is a partner that cannot provide the resources they've stated, there are ways to provide extra capacity in terms of HW and SW. |
| R4 | Difficulties in the connectivity between the testbeds and the testers | L | M | Connectivity within the testbeds is an issue that needs to be overcome during the validation phase. It is not very likely that the basic connectivity is not available. Some of the partners are connected to the GEANT web which means that connectivity can be solved through these existing networks, at least 2 of the partners are NRENs (GRNET and PSNC). |
| R5 | The DC does not provide the information the way the DCO brokers are expecting in terms of standard messaging system or the frequency that is | L | M | Re-educate the DCO Brokers on the way the DC works. Simulate a DC system as close as possible to the normal functional actions of a working DC system. |

| | | | | |
|---|---|---|---|---|
| | expected initially. | | | |
| R6 | Difficulties connecting to the Smart Grid | L | M | The Smart Grid connectivity is being implemented by one of the partners. In case whatever problem in the integration with the Smart Grid system, an interim solution based in developing specific piece of SW will be implemented to overcome those issues. |

**Table 21 – Risk management and contingency plan**

# 5. Testbed description

This section provides details about the setup of the different testbeds provided by the partners. It has been carefully taken to provide the following core details that are relevant to the integration procedure at each testbed:

1. Hardware architecture of the DC which includes: number and specs of the server and other hardware, the organization of the servers in the DC tiers (rack, area-floor, DC), specialized equipment available on site, etc.

2. Software stack that is used for the management of the DC.

3. Monitoring capabilities of the testbed.

In addition to the above core details, particular features of each Data Centre are highlighted in the relevant sections.

To better organize the description of the different testbeds, a distinction is made between *Integration and Demonstration* testbeds. Integration testbeds are small, ad-hoc setups that ideally share most common elements and can be used for DOLFIN module development and preliminary integration work on site. Demonstration testbeds are the main testing sites that follow the DOLFIN release schedule and on which detailed operational tests are performed on the DOLFIN system as a whole.

## 5.1. Integration testbeds

### 5.1.1. UCL TestBed

#### 5.1.1.1. *Overview and general information*

The following presents the developed Very Lightweight Software Driven Network and Service Platform (VLSP) testbed which is continuously developed and enhanced at UCL and has been released as open-source software under the LGPL licence[2].

It represents a software Data Centre allowing hosts and virtual machines to be created, managed, and destroyed under software management control. Multiple instances of VLSP can represent multiple Data Centres. Due to the lightweight nature of VLSP and its speed it is an ideal test environment for testing and evaluating new and novel DC management capabilities.

VLSP is a unique testbed as it has a common management interface that manages combined infrastructure in a way not seen before. It presents a common interface for Computing, Service, and Networking infrastructure which allows management and control operations that are not available with current management stacks such as OpenDaylight or OpenStack.

Such functionality is ideal for DOLFIN as we can test multiple DC topologies in a lightweight and flexible way, as well as using a single stack.

#### 5.1.1.2. *Hardware architecture*

VLSP can run on one node or can be hosted on UCL's 12 H/W servers.

#### 5.1.1.3. *Software architecture*

VLSP is an all software testbed, and can have up to 700 virtual nodes on the 12 H/W servers. More virtual nodes can be created if more H/w is available.

The software elements of VLSP are presented in the following sections.

[2] http://clayfour.ee.ucl.ac.uk

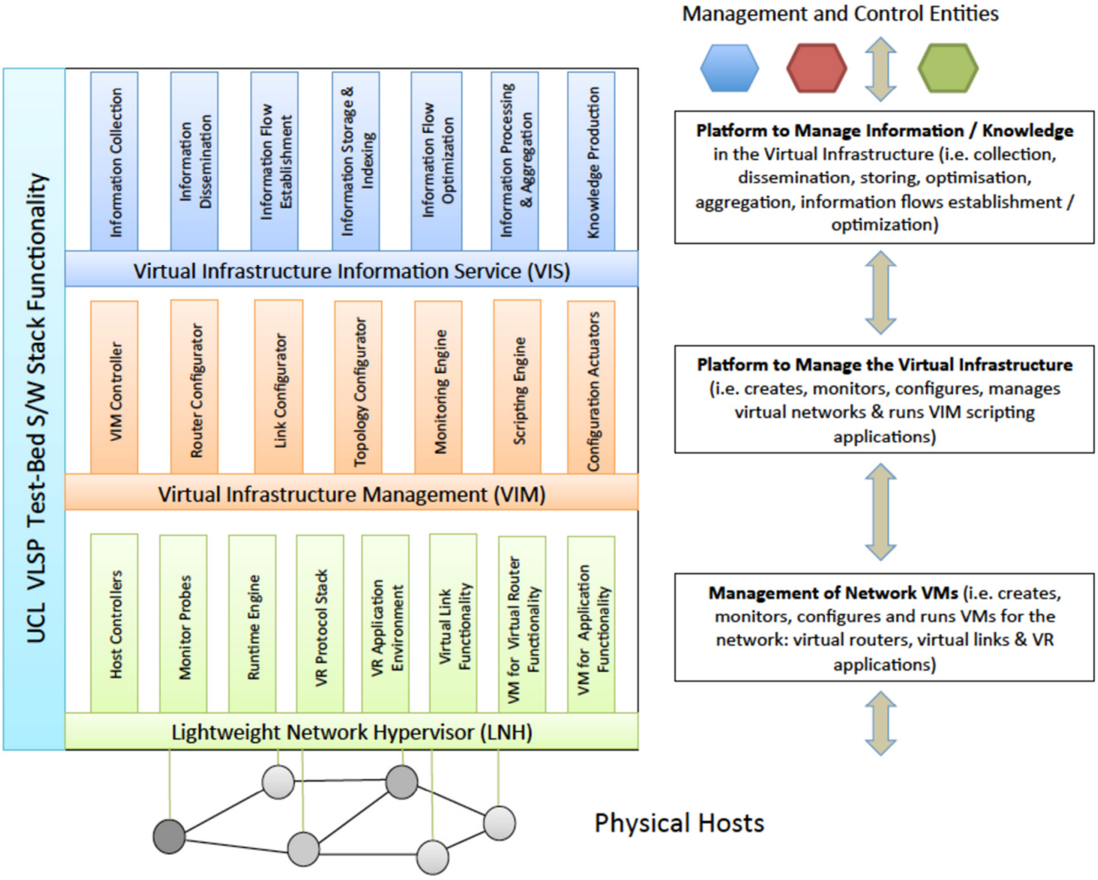


**Figure 5-1: Overview of VLSP Test-bed Software Stack – University College London**

An architectural overview of the software stack is shown Figure 5-1. The VLSP test-bed software consists of over 700 java classes and more than 150 k-lines of code. In our experiments with VLSP, we have executed over 100 virtual routers on each of 12 dedicated physical servers. We detail the three main layers of the VLSP test-bed below.

The VLSP test-bed includes a new lightweight network hypervisor, a novel infrastructure for the management and manipulation of virtual networks on top of the hypervisor and facilities for handling information / knowledge in the SDN environment.

The VLSP software stack consists of three layers:

- the Lightweight Network Hypervisor (LNH),
- the Virtual Infrastructure Manager (VIM), and
- the Virtual Infrastructure Information Service (VIS).

**The Lightweight Network Hypervisor Component**

The Lightweight Network Hypervisor includes a fully operational lightweight virtual router (VR) combined with virtual network connectivity. These elements can be combined in order to build any network topology required. The created virtual network is designed with the goal of transmitting and routing datagrams from any source to any destination. It behaves like a lightweight virtual network, but it has management facilities to start and stop virtual routers on-the-fly, together with the ability to create and tear down network connections between virtual routers dynamically.

Furthermore, these lightweight routers have an application layer interface that provides the capability to start and stop Java software applications. These applications use a transport protocol API which can send and receive datagrams or packets, and thus act as the service elements within the platform.

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

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

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

| **Name** | **Description** |
|---|---|
| Host Controllers | The host controllers execute on every physical machine, allowing for manipulating & configuring virtual routers, links and virtual router applications. |
| Monitor Probes | The monitor probes are tiny configurable applications probing the software or hardware for monitoring data. |
| Runtime Engine | It is responsible for the runtime operation of the LNH, including support for event-based notifications and time scheduling. |
| Virtual Router Protocol Stack | The lightweight network protocol stack of the VRs. |
| Virtual Router Application Environment | The application environment that hosts VR applications. |
| Virtual Link Functionality | The functionality of the virtual links, including link weighting and other configuration options. |
| Virtual Machine for Virtual Router & | A virtual machine with the virtual router and the relevant |

| | |
|---|---|
| Application Functionality | applications functionality. |

Table 22 – The main LNH functions

**The Virtual Infrastructure Manager**

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

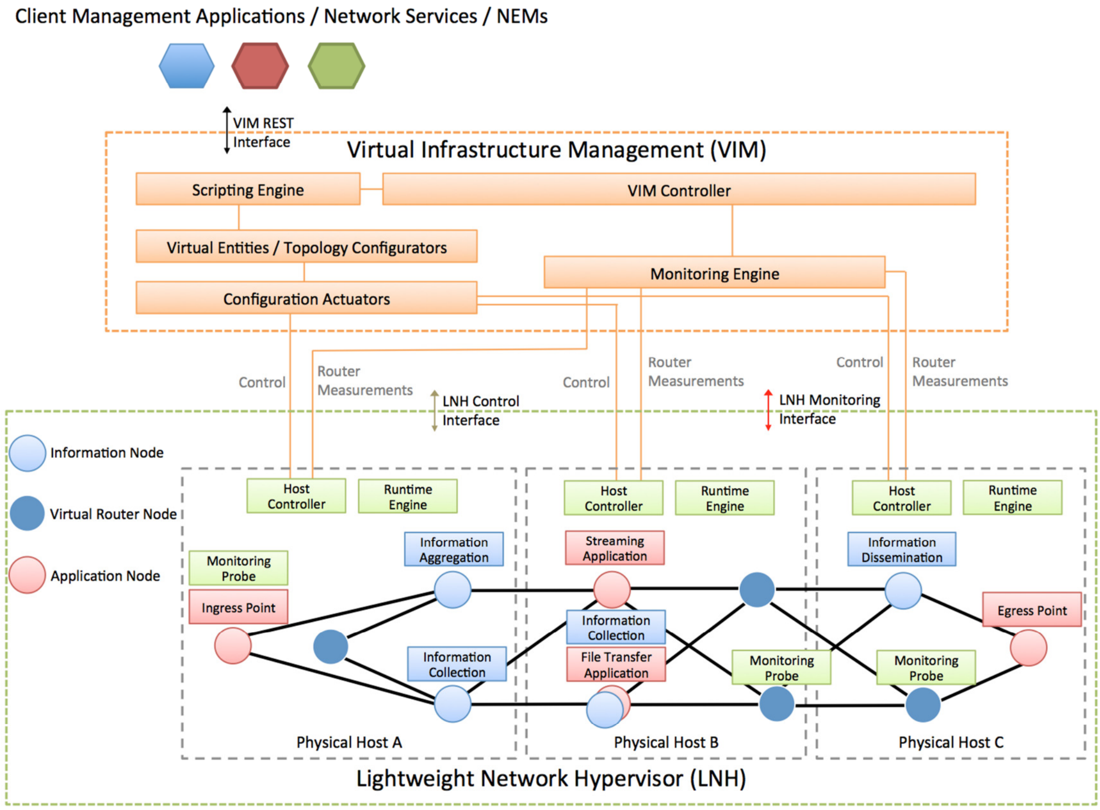


Figure 5-2: The Virtual Infrastructure Management Component

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

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

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

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

The diagram in Figure 5-2 shows how the VIM component interacts with the virtual network elements that will be present in a running virtual network. All of the elements of the VIM component constitute a fully distributed system, whereby an element or node can reside on any host. A full virtual network can be instantiated on a single machine, for demonstration or testing purposes, or instantiated across multiple servers, in a full deployment situation.

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

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

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

The main VIM functions specification follows.

| **Name** | **Description** |
|---|---|
| VIM Controller | It is the heart of the component, providing the central control of the VIM operations. |
| Scripting Engine | VIM can be configured using Closure scripting. |
| Monitoring Engine | It is the main monitoring component of the infrastructure, i.e., collecting & manipulating measurements from the monitoring probes residing at the LNH. |

| | |
|---|---|
| Virtual Entities / Topology Configurators | These functions are responsible for the configuration of virtual routers, links and topologies, supporting different levels of abstraction. |
| Configuration Actuators | The Virtual Entities / Topology configurators communicate with the configuration actuators which in turn enforce the configuration changes through the LNH's host controllers. |

Table 23 – The main VIM functions

**The Virtual Infrastructure Information Service**

The Virtual Infrastructure Information Service component is a critical part of the VLSP and UMF since it plays the role of information / knowledge collection, aggregation, storage/registry, knowledge production and distribution across all the functional components, management applications and network services in the network environment. It can run on top of VIM, since it is fully integrated within the virtual network, e.g., the virtual routers have embedded information / knowledge manipulation capabilities. Furthermore, it is used by any client management application /network services or core service. As we have shown above, VIS is fully integrated within VLSP but acts as a standalone component as well. In this context, VIS is the knowledge core service and supports a wide-range of network environments beyond SDNs.

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

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

Table 24 – Main VIS functions

#### 5.1.1.4. *Monitoring facilities*

The VLSP has embedded monitoring in all of its components.

There will be a stream of events for each of 4 main aspects.  Here we show list of event for each of the aspects.

***VIM Manager Info***

START_VM_MANAGER localhost:8888

START_HOST localhost:10000

START_VM 1 Router-1

START_VM 2 Router-2

START_VM 3 Router-3

….

STOP_VM 1

STOP_HOST localhost:10000

STOP_VM_MANAGER localhost:8888

***Green Host***

Also send if host is green or not.

localhost:10000 NORMAL

localhost:10001 GREEN

localhost:10002 NORMAL

***Host Attribute Info***

"cpuIdle":81.98, "cpuSys":11.71, "cpuUser":6.3, "freeMemory":5074, "hostId":"localhost:10000", "inBytes":11193, "inPackets":67, "outBytes":18120, "outPackets":63, "usedMemory":11303

***VM Attribute Info***

"cpuSys":8.696999549865723, "cpuTotal":75.61599731445312, "cpuUser":66.91899871826172, "elapsedTime":145352, "energy":8.93398818969726, "energyTotal":248.3977958679199, "hostId":"localhost:10002", "memory":73723, "starttime":1438333836458, "vmId":3, "vmName":"Router-3"

##### *5.1.1.4.1. Energy Model and Energy Viewer*

An extension to the VLSP platform has been built that calculates the amount of energy used by a VM. In DOLFIN, we manage and manipulate VMs in order to optimize the energy consumption of either one DC or a group of DCs. DOLFIN determines any possible energy saving in Data Centres through careful placement of these VMs. In essence, VMs are the basic unit of management decisions. Although energy is consumed by servers and other equipment, it is the management of VMs that affects the outcome of any decisions related to energy usage.

In order to make a decision regarding any optimization, we need to determine the current energy consumption within a Data Centre, so that decisions about placement of virtual machines can be made. The current energy consumption is calculated from the resources within the Data Centre, and this including both the physical resources and the virtual resources. In order to bring about energy savings, the VMs must be placed carefully on physical machines. Such an optimization process is a continuous loop of energy monitoring, decision making, and plan actuation.

It is not effective enough to make such decisions, considering only the power consumption of a rack or each of the hosts. On the contrary, the energy consumption of each VM is needed. However, there are currently no mechanisms for determining the power consumption of each virtual machine. In reality, it is unlikely that any hardware monitor can be built to determine VM power consumption, and there is currently no support in operating systems to support such functionality. The only solution is to observe the actual dynamics of VMs on a host and evaluate the energy consumption through the use of an *Energy Model.*

The *Energy Model* takes as input various attributes of each virtual machine and derives an energy usage value for each VM. The Energy Viewer collects per-VM data and displays it. Below is a screen snapshot of the *Energy Viewer* output. On the top line, it shows, the CPU usage, CU idle and memory consumption for a nominated server. On the second line is the total amount of energy consumed by all of the VMs on the server – as calculated by the Energy Model. After these two summary lines the Energy Viewer shows the details for each running VM, including: name, elapsed run time, total CPU, user space CPU, system CPU, memory used, energy used, and energy delta since last cycle of the viewer.

```
CPU Usage: 30.32% used  69.67% idle        Mem Usage: 15.97 Gb used  0.02 Gb free

Energy Usage: 1189.71 Total (Wh)   41.59 Delta (Wh)


name            elapsed (s)  cpu (ms)    user (ms)   sys (ms)    mem (k)    energy(Wh)  delta(Wh)

Router-7         20.27       397.606     366.285     31.321      13107       45.003      0.000

Router-8         5.26        222.793     202.076     20.717      6663        24.944      0.000

Router-2         75.31       695.509     626.732     68.777      19235       77.245      0.000

Router-1         90.59       1463.370    1339.990    123.380     41562       162.962     6.862

Router-3         65.30       695.421     627.915     67.506      19483       77.335      0.000

Router-4         55.54       652.130     590.521     61.609      18758       72.716      2.187
                                                                             ----------  ------
```

| | | |
|---|---|---|
| | 460.206 | 9.049 |

We intend to extend the *Energy Viewer* to incorporate data from the Smart Grid Controller, including the current energy price. In this way, the *Energy Viewer* will be able to calculate and present the cost of the energy consumption for each VM and for the total server.

### 5.1.2. GRNET TestBed

#### 5.1.2.1. *Overview and general information*

The main GRNET DC is located within the premises of the Greek Ministry of National Education and Religious Affairs in Athens. The DC is currently equipped with 28 racks hosting servers and storage equipment reaching a total of 7132 logical CPUs and 1800 TB of storage. The average energy consumption of the equipment hosted at this DC is currently around 250 kW. The achieved Power Usage Effectiveness (PUE) varies close to 1.8, while the energy consumption in 2011 was close to 850 MWh.

This DC has been designed and implemented following high standards regarding cooling efficiency. In-row cooling techniques are applied, while the equipment is fully virtualized. PUE is further optimized with free cooling techniques. The DC chillers are connected in parallel with air cooled heat exchangers. When the ambient air temperature drops to a set temperature, a modulating valve allows all or part of the chilled water to by-pass the existing chillers and run through the free cooling system, which uses less power and uses the lower ambient air temperature to cool the water in the system.

DOLFIN will be integrated on-site, using part of the infrastructure to provide a small but representative integration testbed for use by DOLFIN. Specifically, the testbed can be composed of at most 16 virtual servers, using a Virtual Machine Container server (VMC) (whose technical characteristics are detailed in the following table.

| | |
|---|---|
| **Processor model** | AMD Opteron 6172 |
| **Number of processors** | 2 |
| **Number of cores per processor** | 12 |
| **Number of threads** | 12 |
| **Memory** | 192GB |
| **Frequency** | 2.4GHz |
| **Local Storage** | 480GB |

| | |
|---|---|
| **Power consumption** | 80W |
| **Size** | 2U |

Table 25 – Technical characteristics of a VMC of ~okeanos.

These servers can be used to provide a representative development snapshot of an Openstack installation, which will be the primary target DCO Hypervisor Manager. Specific monitoring inputs can be simulated so as to be provided during testing to DOLFIN development snapshots installed on the testbed. Using this setup, we can simulate a large variety of network, monitoring and Smart Grid APIs to test DOLFIN against different DC configurations. This setup also can be easily bootstrapped each time a test is needed to be run from scratch, so as to facilitate DOLFIN development and integration, by providing developers with a pristine and consistent integration environment.

#### 5.1.2.2. *Hardware architecture*

The hardware infrastructure of the GRNET TestBed in shown in Figure 5-3. The GRNET Data Centre hosts 234 HP DL385G7 servers to act as virtual machine containers. Servers include a dedicated Ethernet port for remote administration over TCP/IP providing both remote console and Video Graphics Adapter (VGA). Servers can be monitored remotely using protocols such as Intelligent Platform Management Interface (IPMI) 2.0 and Systems Management Architecture for Server Hardware (SMASH) Command Line Protocol (CLP) providing information on the server's status including hardware health, power consumption, temperature etc. Server's power supply and cooling facilities can be remotely administered. Remote on and off powering is also supported.

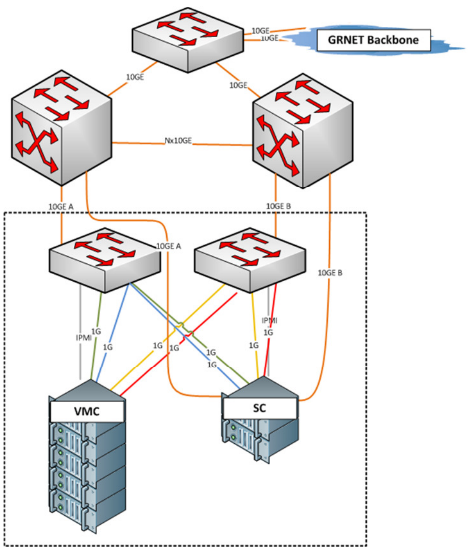


Figure 5-3: The hardware infrastructure of the GRNET testbed

**Network:** The GRNET TestBed is well interconnected with Nx10 Gbps network links with the GRNET backbone network, ensuring the reliable and efficient provision of services. Dedicated circuits (even optical circuits) may be provided within specific locations in the GRNET network for testing purposes as well as for occasions where very high Quality of Service (QoS) requirements are imposed.

### 5.1.2.3. *Software architecture*

The GRNET DC hosts an Infrastructure-as-a-Service (IaaS) platform called ~okeanos operating as a public cloud for the Greek academic community. The goal of the ~okeanos project is to deliver production-quality IaaS to GRNET's direct and indirect customers, IT departments of connected institutions and students/researchers respectively. ~okeanos offers to its users access to VMs, Virtual Ethernets, Virtual Disks, and Virtual Firewalls, over a simple web-based user interface (UI). ~okeanos was conceived to offer easy and secure access to GRNET's DCs, focusing on user friendliness and simplicity, while being able to scale up to thousands of VMs, users, terabytes of storage.

The software underlying ~okeanos, called Synnefo, is custom open-source cloud management software. It encompasses a number of distinct components, all sharing a single installation and configuration mechanism, to streamline operations. Synnefo has a compatible API with OpenStack, and its software stack is shown in Figure 5-4 next to that of OpenStack.

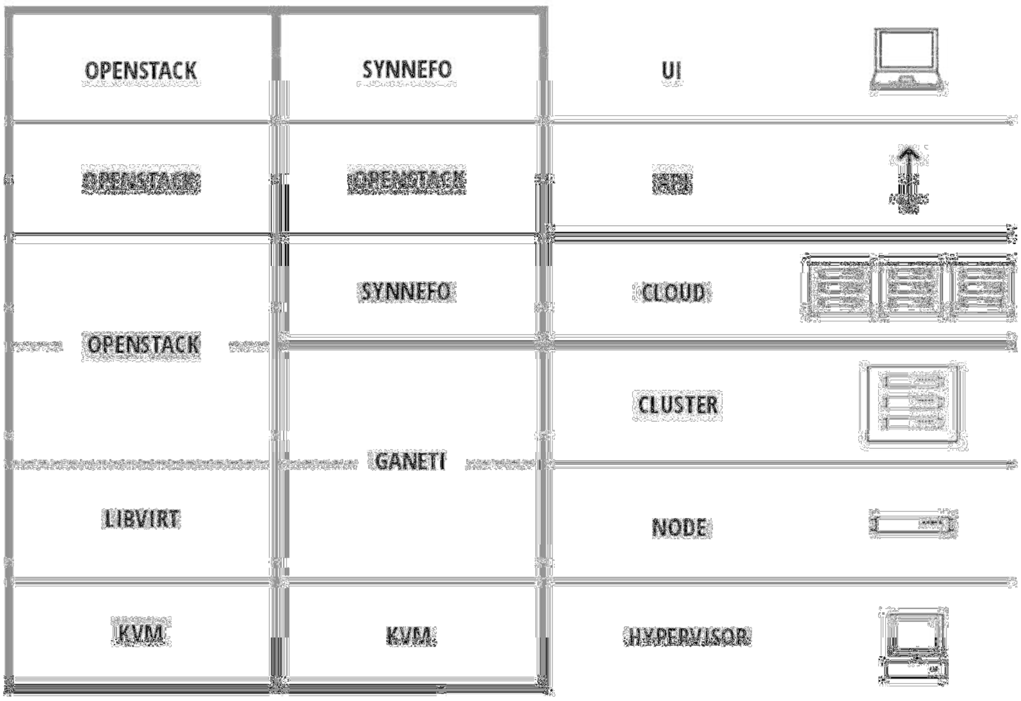


**Figure 5-4: The synnefo software stack for GRNET ~okeanos IaaS platform**

Cyclades is the Compute/Network part of ~okeanos. Its design combines a Google ganeti backend for VM cluster management (VM creations, migrations, etc.) with a Python/Django implementation of the user-visible API at the front-end. Ganeti organizes the VMs of the platform in Clusters and assigns one Ganeti master per cluster to manage VM creations, migrations and failovers. The REST API for VM management is OpenStack compute v1.1 compatible and can interoperate with third-party tools and client libraries. GRNET has added custom extensions for unsupported functionality.

### 5.1.2.4. *Monitoring facilities*

Monitoring infrastructure is deployed supporting real-time monitoring of energy consumption per rack, as well as real-time estimation of PUE.

Energy consumption is monitored with specialized metering devices situated in the electrical panels, so that the total power consumption, the chillers and the rest infrastructure components can be also measured. In each rack there are installed power distribution units with monitoring capabilities to allow fine grain monitoring of the equipment inside the racks. Also, modern server hardware technologies are deployed with an on-board management controller that allows power monitoring in a per server basis. All this information can be accessed through standard protocols like HTTP, FTP or SNMP and can be easily integrated into a third-party system.

## 5.2. Demonstration infrastructures

### 5.2.1. WIND TestBed

#### 5.2.1.1. *Overview and general information*

One of the DOLFIN testbed environments will be implemented within Wind's data centres to explore the advantage coming from the DOLFIN energy reducing approach, achieved by moving computation and services on a "federation" of IT data centres sites.

The Wind testbed is located in a laboratory facility completely separated from the production environment for security reasons. In Wind testbed, the federation will be simulated utilizing two separate physical rack units, each of which acting as a single DC, plus a third rack unit acting as controller/supervisor of the federation. The three rack units are co-located in the same room and in the same rack, so the "federation" is actually just a simulation; nevertheless, from a logical point of view, all the conditions needed to have a "federation" are met.

In this section, we describe the hardware and software setup of Wind testbed taking into account the fundamental logic blocks:

1. The Hardware architecture
2. The Software architecture
3. The Monitoring equipment

#### 5.2.1.2. *Hardware Architecture*

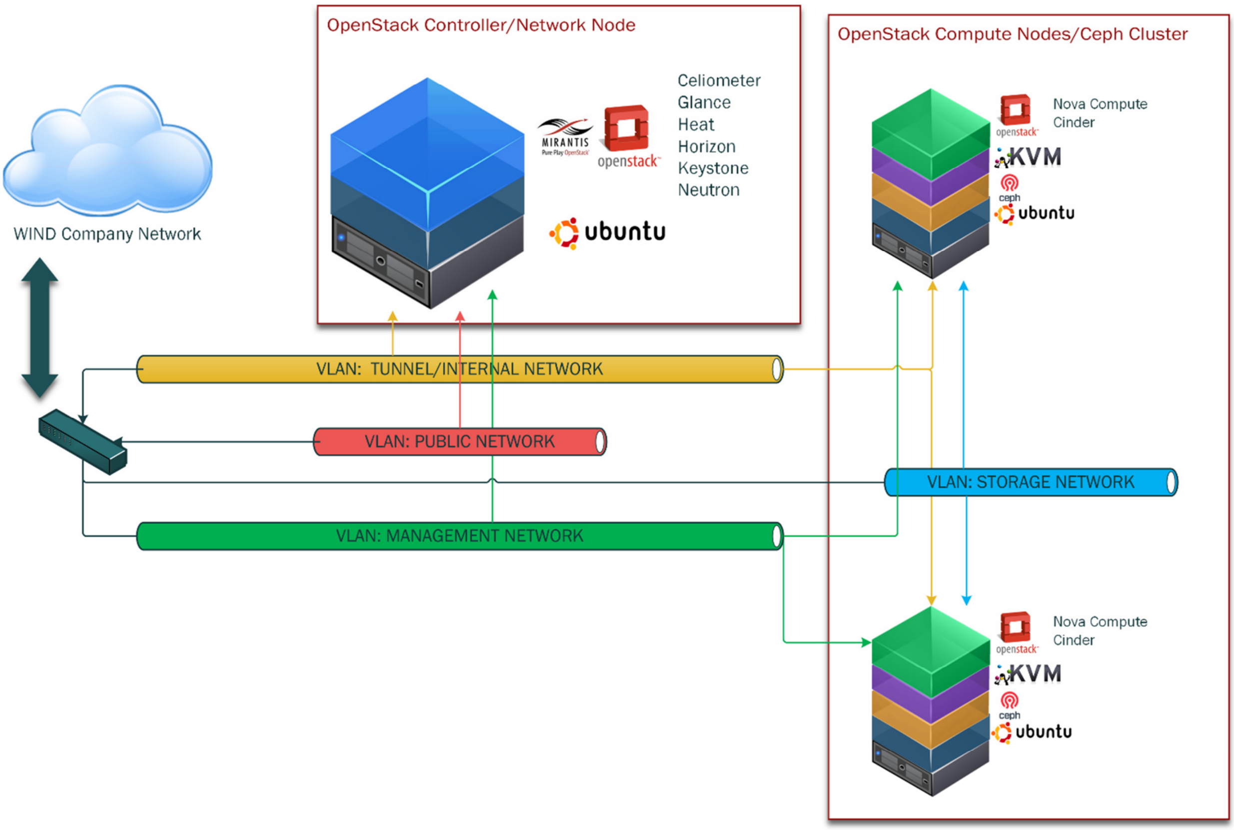


**Figure 5-5: Wind Testbed physical and logical schema**

In the above table a view of Wind Testbed is presented. A total of 3 rack units compose the Wind testbed. One unit acts as OpenStack Controller and Network node, thus managing the whole infrastructure, while the other two act as OpenStack compute nodes, thus providing the actual virtualization capacity.

The three rack units are located in the same room on the same rack. The room is used only for hosting this testbed, so no other physical components of Wind datacentres are co-located with the DOLFIN Testbed.

The architecture is based on the following components:

- **Controller and Network Node:**
    - HP DL380 gen 9 rack server
    - CPU: 8 core
    - RAM: 32 GB
    - HDD: 500 GB
- **2x Compute node:**
    - HP DL380 gen 9 rack server
    - CPU: 16 core
    - RAM: 64 GB

- HDD: 500 GB

- **48 port Switch**
- **1 Laptop acting as a firewall**

The network architecture between the two datacentres is pure layer 2 network architecture. Given the co-location of the two computation nodes, no layer 3 appliance is needed. In case of distant physical locations, a LAN (to WAN) Extension would be necessary. A LAN Extension is a technical solution for extending the LAN between two sites using dedicated fibres and Layer 2 over Layer 3 transport mechanism for long distances.

From a logical point of view, the network infrastructure must provide four completely separated and dedicated LANs between the compute nodes and the controller node. This is done using the following VLANs, here described:

- PUBLIC (or EXTERNAL) NETWORK: This is the only network that has access to the company internal network. Only the controller node is connected to this VLAN: all the other compute nodes (and all the VMs of course) can communicate with the external network passing through the controller node.
- MANAGEMENT NETWORK: This is the internal network used by the nodes to communicate between themselves for management purposes. All the three nodes are connected to this network
- TUNNEL NETWORK: This is the internal network where the VMs' virtual networks are set up using GRE tunnels. All the three nodes are connected to this network.
- STORAGE NETWORK: This is the internal network used by Ceph to exchange the VMs' data between the virtualization nodes. Only the two compute nodes are connected to this network

In our tests, the performance capabilities of the network infrastructure will not be considered.

#### 5.2.1.3. *Software Architecture*

The chosen cloud platform for testing DOLFIN components is OpenStack. Wind's testbed is supplied with Mirantis Openstack Juno Distribution, on Ubuntu 14.04 x64 server edition.

OpenStack relies on KVM as hypervisor, and is equipped of its own SDN component, called Neutron.

The storage is managed by *Ceph*: an open-source software storage platform. The compute nodes form a completely distributed storage cluster. A dedicated VLAN used only by Ceph is shared between the compute nodes, thus ensuring data mobility, high availability and enhanced cooperation between the two sites. Data is therefore presented in a consistent manner within and between sites.

#### 5.2.1.4. *Monitoring Facilities*

All the rack units are equipped with iLO 4 boards.

These boards can provide the following information regarding the rack unit status:

- Current temperature.

| Sensing location | N# of sensors |
|---|---|
| Ambient | 2 |
| CPU | 1 |
| Memory | 2 |
| System | 8 |
| IO Board | 2 |
| Power Supply | 3 |

- Fans:
  - Fan status
  - Fan speed
- Power:
  - Maximum power supply capacity
  - Actual power supply capacity used

This information can be retrieved via SNMP from the iLO boards. The boards, being hardware components, do not rely on the OS installed on the node, and therefore remain active in case of software failure on the server.

There are no sensors at the rack or room level.

### 5.2.2. Interoute Testbed

#### 5.2.2.1. *Overview and general information*

Interoute testbed is located in the Interoute Milan PoP in which a dedicated co-location area is made available to host dedicated servers for the DOLFIN software and demonstration. The co-location area provides all the technical equipment and infrastructures that any Interoute's DC or production facility implements. In the following figure a generic co-location site is represented.

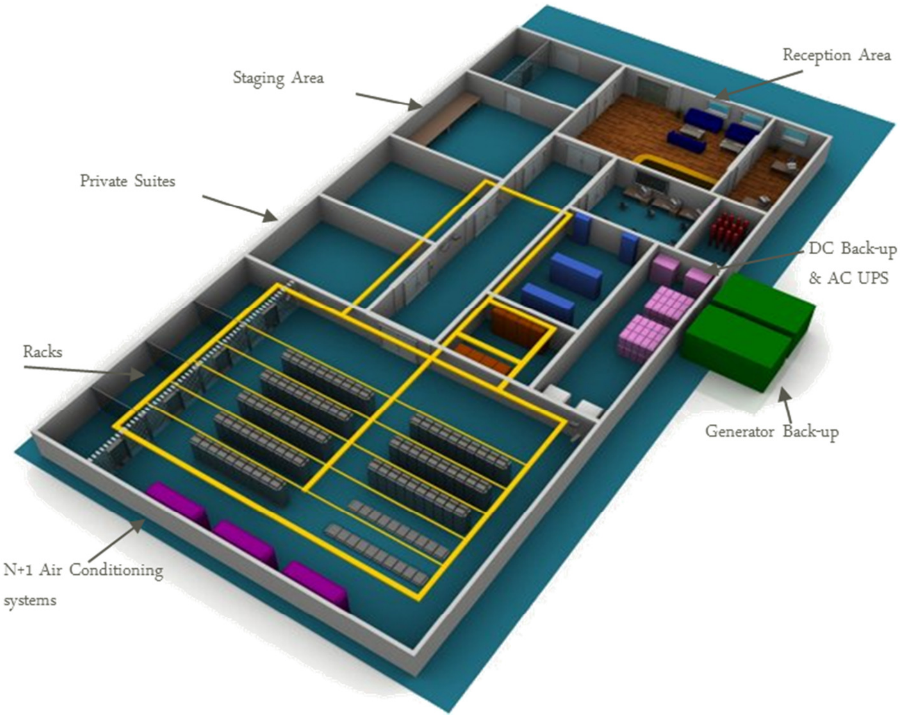


Figure 5-6: Co-location site infrastructure overview

##### *5.2.2.1.1. Power supply management*

All Interoute co-location sites are provided with AC & DC power as standard. Both power systems are designed with N+1 resilience configuration provisioned to all key elements including Uninterruptible Power Supply (UPS) and rectifiers, with initial battery back-up and generator facilities available for instances of prolonged supply loss. Each cabinet or footprint within the housing facility is supplied with either AC (up to 16A at 220 V) or DC (up to 40A at 48V) current depending on the technical need. Each power supply (AC and DC) has two separated and independent power feeds in order to ensure redundancy and guarantee 99.99% power availability.

##### *5.2.2.1.2. Cooling system*

Temperature and humidity control is provided by a heat rejection chiller system with N+1 redundancy configuration (each with capacity of 360kW) that ensure a temperature of 23°C (+/- 3°C) and humidity rate of 50% (+/- 10%).

##### *5.2.2.1.3. Network infrastructure*

The co-location facility provides double fibre entry and diverse paths within the building connecting directly to the Interoute Multiprotocol Label Switching (MPLS) backbone network.

The generic architecture for the co-location networks is based upon the Interoute PoP network architecture and features the following:

- A double access router layer to provide IP-layer termination.

- A switch aggregation layer to provide aggregation of multiple edge switch functions.
- An access switch device to provide Ethernet-layer termination on 10/100/1000 Mbps Ethernet ports.

Each co-location site can support up to 9 edge switches with 48 10/100/100 Mbps ports, with a 2 Gbs aggregated link between them.

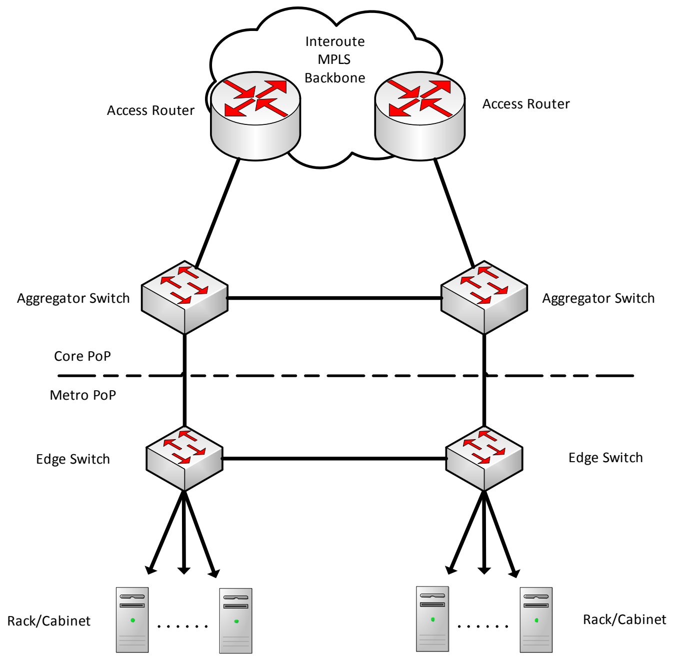


Figure 5-7: Interoute PoP's high level network design

Access to the public Internet is provided using a redundant connection to the Interoute MPLS Backbone and it is managed by two access router working in fail-over configuration. The customer equipment is connected to a cell switch that receives a direct Gigabit fibre connection from each Edge switch. Each customer is assigned his own VLAN on the cell switches named with the customer's service number. Each customer is also assigned his own subnet from the public IP space and the first useable IP address from the subnet is attached to the customer VLAN.

#### 5.2.2.2. *Description of testbed*

The demonstration testbed provides a dedicated hardware infrastructure that will simulate, through a virtualized environment, two separated DCs. The main testbed's purpose is the evaluation of the DOLFIN components and the way they will behave during a complete simulation when all the functionalities will have to mutually interact. To address this purpose the following subsections report a description of the HW infrastructure, of the SW architecture and of the setups that will be used to simulate the testing scenarios described in section 3.

##### 5.2.2.2.1. *Hardware architecture*

For the purposes of DOLFIN demonstration, a setup of 2 servers hosted in the dedicated DOLFIN co-location area is planned.

Each server will have the following specifications:

- (2x) 8 or 10 Core CPU
- Hyper-Threading support up to 4x physical core
- 256 GB of Memory
- 1TB storage HDD
- (4) 1GbE Ports
- iLO Chassis Lights Out Management Card

The proposed hardware specifications and the specific hypervisor software will provide a virtualized environment where a considerable number of VMs can be hosted in order to emulate a DC environment. Within the emulated DC, the DOLFIN components can be tested and evaluated. The DOLFIN servers will access the Internet through a 2Mbps dedicated outbound connection, mostly used for servers' management purposes.

##### 5.2.2.2.2. *Software architecture*

Both servers can be equipped with any free-licence hypervisor software (like KVM), but in order to ease the overall management of the virtual environment, both servers will run Ubuntu Server 14.04.3 including the Icehouse release of OpenStack that will offer a wide range of configurations and settings, a stable virtualized environment, and the possibility to interact with the DOLFIN software components.

The virtualized environment can be deployed in two different configurations. The choice will be taken in agreement with partners depending on which is the most suitable solution in order to comply with the demonstration's needs.

The first configuration envisages a single OpenStack controller deployed on a single physical server linked to two OpenStack Compute Nodes deployed on both physical servers. Following is depicted the first configuration's logical overview:

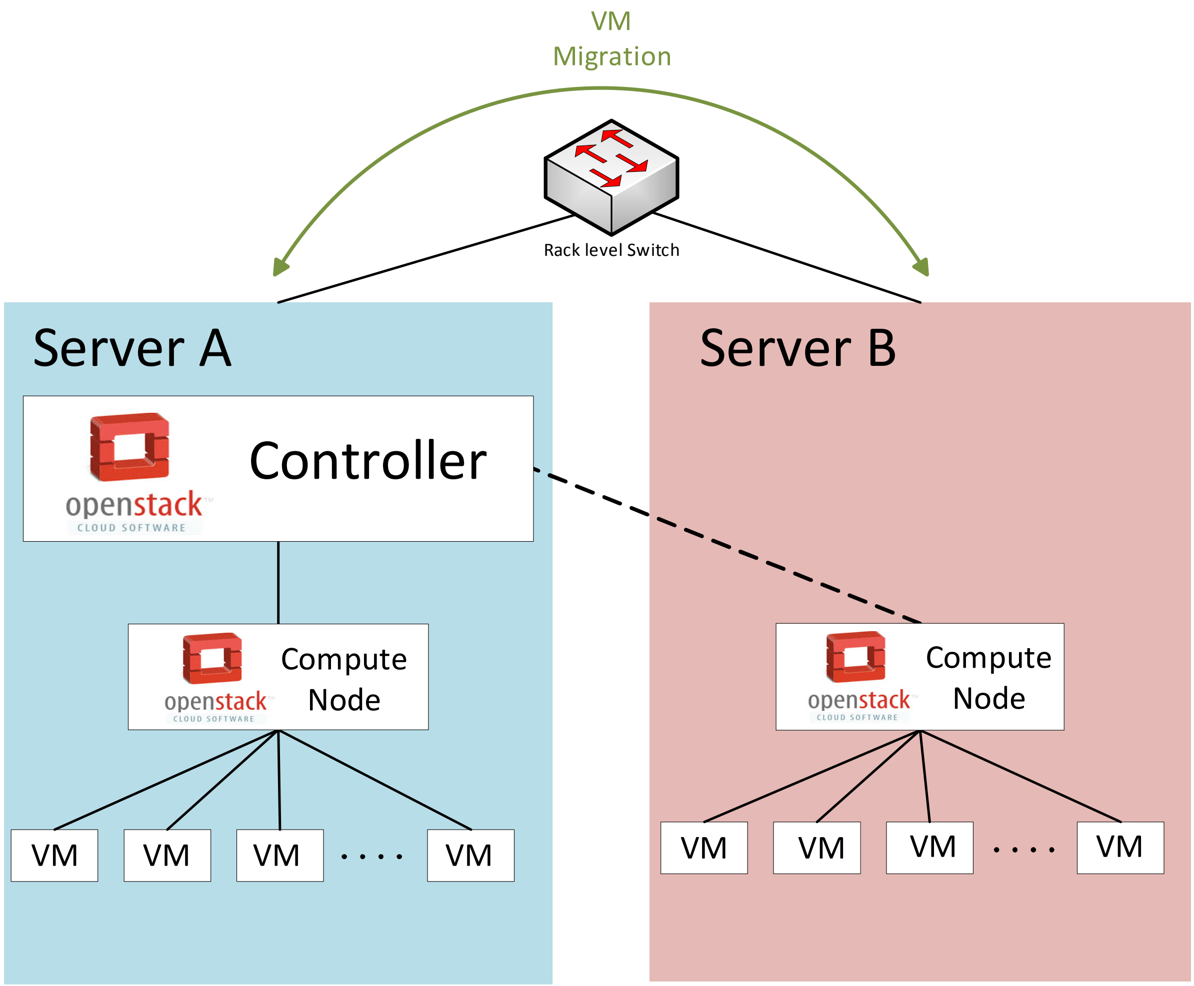


Figure 5-8: Single-homed Controller configuration overview

The second configuration will provide two independent OpenStack Controllers deployed on both physical servers, each one managing a single OpenStack Compute Node that will instantiate the VMs needed to run the demonstration. In the following figure is depicted the logical overview.

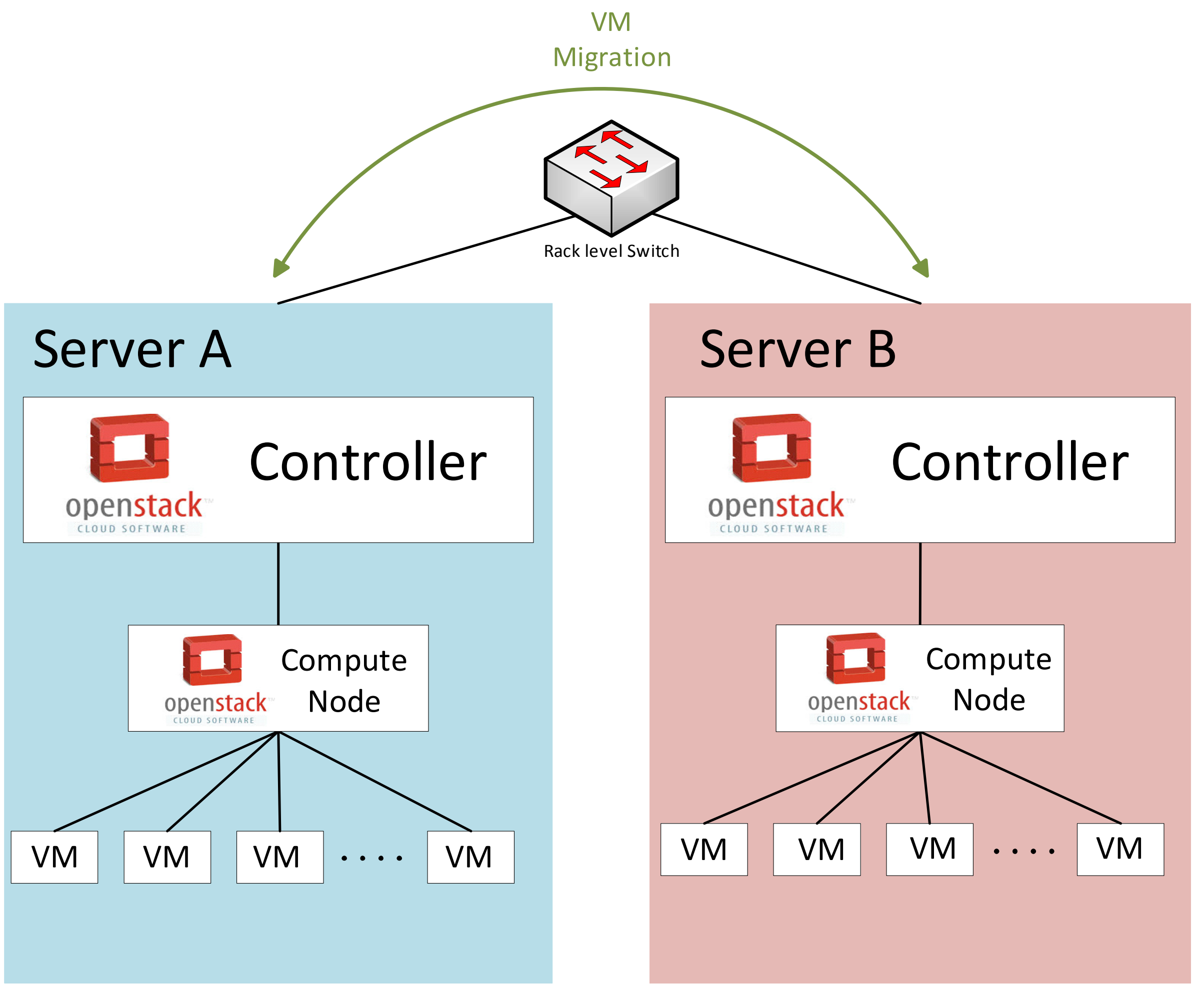


Figure 5-9: Multi-homed Controller configuration overview

Additionally, the following OpenStack components (among others) will be provided:

- OpenStack Compute – Nova: this component allows the user to create and manage virtual servers using the machine images, and to provision large networks of virtual machines.
- Block Storage – Cinder: this component provides persistent block storage and enable flexible management of storage devices
- OpenStack Networking – Neutron: the use of Neutron allows users to manage all the networking features including VLAN, Firewalls and IP addresses assignment.
- OpenStack Telemetry Service – Ceilometer: this tool provides efficient collection of metering data, in order to monitoring CPU and network resources.
- Dashboard – Horizon: this component provides a web-based interface that enables the interaction with all the underlying OpenStack services.

##### 5.2.2.2.3. *Monitoring facilities*

Both servers are provided with iLO Chassis Lights Out Management Card which allows the system administrator to manage and monitor servers via remote control. The iLO card exposes an easy-to-use web interface with the following features (among others):

- Monitor system fans, temperatures, and voltages remotely.
- Monitor system BIOS messages remotely.
- Monitor system operating system messages remotely
- Monitor system status (health check) remotely
- Interrogate system network interface cards remotely for MAC addresses.
- Manage system power status remotely (power on, power off, power reset).

Monitor and manage environmental settings for key system components (CPUs, motherboards, fans)

#### 5.2.2.3. *DOLFIN testing scenarios configuration*

Depending on the test scenario that will be implemented, the testbed virtual configuration will be deployed according to the two models described in the previous section (sec 5.2.2.2.2).

Referring to the previously defined scenarios, the testbed is structured to meet the following requirements:

- Two servers will be used to host the VMs, the OpenStack Controller and the DOLFIN software components devoted to evaluate the specific scenario. One of the two servers will be used as a “green” server in order to simulate the energy optimization procedures at a local level. In this case, since the two servers have exactly the same features, the energy consumption of the “green” server will be revalued in order to simulate the lower energy consumption.

- Functions for monitoring the IT hardware consumption (servers, rack level switch, etc.). This information could be provided to the respective DOLFIN Broker through the appropriate interfaces. In particular the data related to the server state, can be provided either through the iLO board interface, or through the SNMP protocol.

- Non-IT facilities monitoring (e.g. server room cooling system). This requirement is finalized to monitor the energy consumption of the cooling system depending on the workload. The testbed environment does not envisage the possibility to control this kind of parameters.

Following is reported the software configuration that will be used for the evaluation of each scenario.

**Intra DC optimization**

In this case the whole testbed has to be considered as a unique DC. Each of the two servers will emulate a single room of the same DC in order to evaluate the DOLFIN components’ reaction in case of an unexpected increase of work load. In this situation the DOLFIN components should be able to manage the energy consumption without affecting the ongoing DC operations.

This scenario will be implemented using the “Single-homed Controller” architecture (sec. 5.2.2.2.2):

- *Server A* will be configured as following:
    - Openstack Controller
    - DOLFIN modules: ICT Performance & Energy Supervisor, Energy efficiency policy maker and Actuator, eCOP Monitor DB, DCO Brokers.
    - VMs Hosting
    - Non-Green footprint (in terms of energy consumption)
- *Server B* will be configured as following:
    - Green-server; we consider the baseline consumption of this server lower than the consumption on Server A.
    - VMs hosting

In this scenario will be validated the reallocation (VMs migration) of workload within the same DC thus within the two servers. In particular it is expected a workload distribution between Server A and B ("green" server) during the period of increasing workload.

**SLA testing scenario**

This scenario envisages the use of cloud federation. The "Multi-homed Controller" configuration (see section 5.2.2.2.2) will be used in order to simulate the VMs migration between two different DCs. In this scenario the eCOP and SDC component, implemented on both servers, will cooperate to manage the VMs' migration in order to meet the rules produced by the *Energy efficiency policy maker*. This test will simulate an inter-DC migration triggered by the eCOP prediction engine. In this case both servers will have the same starting configuration in order to better identify and evaluate the whole testbed behavior depending on the SLA state.

- *Server A and B*:
    - Openstack Controller
    - DOLFIN modules: ICT Performance & Energy Supervisor, Energy efficiency policy maker and Actuator, eCOP Monitor DB, SLA Renegotiation controller, Cross-DC Orchestrator, DCO Brokers.
    - VMs hosting

In this scenario it is expected to observe two different type of reaction conditioned by the SLA configuration of each VM:

- No VMs migration between the two DCs (server A and B), but a local reduction of workload leveraging on the performance parameter of the VMs having flexible enough SLAs that can be forced in stand-by status.
- A migration request will be triggered between the two simulated DCs

**Smart grid testing scenario**

In this scenario will be introduced a new set of input events from the Smart Grid with the purpose of evaluating the efficiency of DOLFIN-enabled DCs to lowering the energy consumption as response to the Smart Grid events. The testbed deployment will support a new DOLFIN module, the Smart Grid Controller, which will act as a gateway and will receive demand-response events from a Demand Response Automation Server.

The architecture that will be implemented to evaluate this scenario will be very close to the one used for the "Intra DC optimization" scenario but, in this case, there will be added the modules needed to interact with the Smart Grid network and the ones needed to simulate the demand-response events coming from the Demand Response Automation Server.

- *Server A* will be configured as following:
    - Openstack Controller
    - DOLFIN modules: ICT Performance & Energy Supervisor, Energy efficiency policy maker and Actuator, eCOP Monitor DB, DCO Brokers, SmartGrid Controller.
    - VMs hosting
    - No-Green footprint (in terms of energy consumption)

- *Server B* will be configured as following:
    - Green-server: we consider the baseline consumption of this server lower than the consumption of Server A.
    - VMs hosting

This scenario will evaluate the reaction of the DOLFIN components as a function of the variation of price and energy availability provided by the Smart Grid Controller. Depending on the parameters set up on each VM (energy policies or SLA configuration), it is expected to observe different behaviors e.g. VMs migration between servers, or VMs being forced in stand-by.

### 5.2.3. PSNC Testbed

#### 5.2.3.1. *Overview and general information*

Poznan Supercomputing and Networking Centre operates a data centre, which is mostly used to execute complex scientific HPC workloads. Thus, PSNC has an access to the real computing infrastructure used by scientists to run their advanced applications. The centre's IT equipment includes diverse top class systems such clusters of high performance servers, SMP machines, and hybrid CPU-GPU systems. In 2015 PSNC has just opened a new data centre. The 1600 square meter data centre is planned for up to 180 racks and 2-16MW of power use (currently 2.5MW transformer is installed). The PSNC DC has 2 floors (+ floor with technical equipment) designed for networking equipment and low density servers (Floor 1), and HPC servers (Floor 2). Direct liquid cooling is planned for the HPC part of the data centre. Schemes of Floors 1 and 2 are illustrated in Figure 5-10.


**Figure 5-10: Floors of the PSNC DC: Floor 1 (bottom) and Floor 2 (top)**

Main parameters of the PSNC DC are summarized in Table below.

| | |
|---|---|
| Floor area | 1600 sq m |
| No of racks | Up to 180 |
| Estimated power loads | up to 16 MW (currently 2MW) |
| Cooling systems | i. IT boxes: 16 racks with common hot aisle, 100kW or 200 kW for IT box depending on the installed hardware<br>ii. All air cooling units will use chilled water from free cooling enabled aggregates (100% free cooling from 6C |

| | |
|---|---|
| | ambient). Max power 4MW<br>iii. Direct Liquid Cooling system. Warm water from the cooling loops used for heating offices |
| Heat re-use system | Heat from DC recovered and used by the LG VRF system |

**Table 26 – Main characteristics of the new PSNC DC**

For the Dolfin project the part of DLC cluster equipped with Xeon processors and CoolIT warm water cooling system can be used for demonstrations. The whole DLC cluster has performance 109,5 Tflops whereas single node 901,33 Tflops. The cluster uses 54,3kW of peak power. The inlet water temperatures can reach 30-40C. The cluster with CoolIT manifolds is shown in figure on the left.

**Figure 5-11: PSNC DC server racks**

In addition to the DC, PSNC has also smaller experimental setups, e.g. high density multi-node computers and very efficient direct liquid cooling system with detailed measurement system.

These systems allow the DOLFIN project to apply arbitrary tools and workloads for testing and demonstration purposes.

**Energy generation and management equipment**

In addition to the ICT infrastructure PSNC will have equipment for generation of energy from renewable energy sources and energy management using energy storage. These elements can be applied to execute dynamic scenarios with variable energy availability and costs.

**Photovoltaic system**

PSNC will have a photovoltaic system on the roof of its building. The installation will have 20kWp of peak power and will include around 80 PV modules. It will be off-grid installation with batteries (~75kWh). The system will also include sun trackers and a mix of mono- and polycrystalline photovoltaic modules. Energy produced by the system is going to be used by external lighting and will be also available to various equipment in laboratories.

Energy generated by the PV system and energy consumption by particular mains will be monitored and send, using Modbus, to the Building Management System.

**Fuel cells**

PSNC will also have an energy management system based on fuel cells and hydrogen generator. Fuel cells deliver a power of 1,2kW. The hydrogen generator produces 30l of hydrogen per hour consuming 300W of power.

Examples of experiments may include backup, emergency power supply (UPS), and autonomous power supply.

#### 5.2.3.2. *Hardware Architecture*

**Intel Xeon-based servers**

The infrastructure consists of two, twin setups differing in the way of cooling. First system is cooled using Direct Liquid Cooling (DLC) method, while the second one exploits air based cooling (AC). The former setup follows Ebullient (http://ebullientcooling.com/) approach, supporting two-phase cooling process based on liquid and vapour heat removal. At the server level DLC is applied both to the processors, GPUS as well as to the memory units.

Figure 5-12: PSNC DC: Intel Xeon

The latter setup uses standard, fan-based scheme with the ability to adjust the fan speed to the current heat-load of the system. The idea of the infrastructure is presented in the figure below:

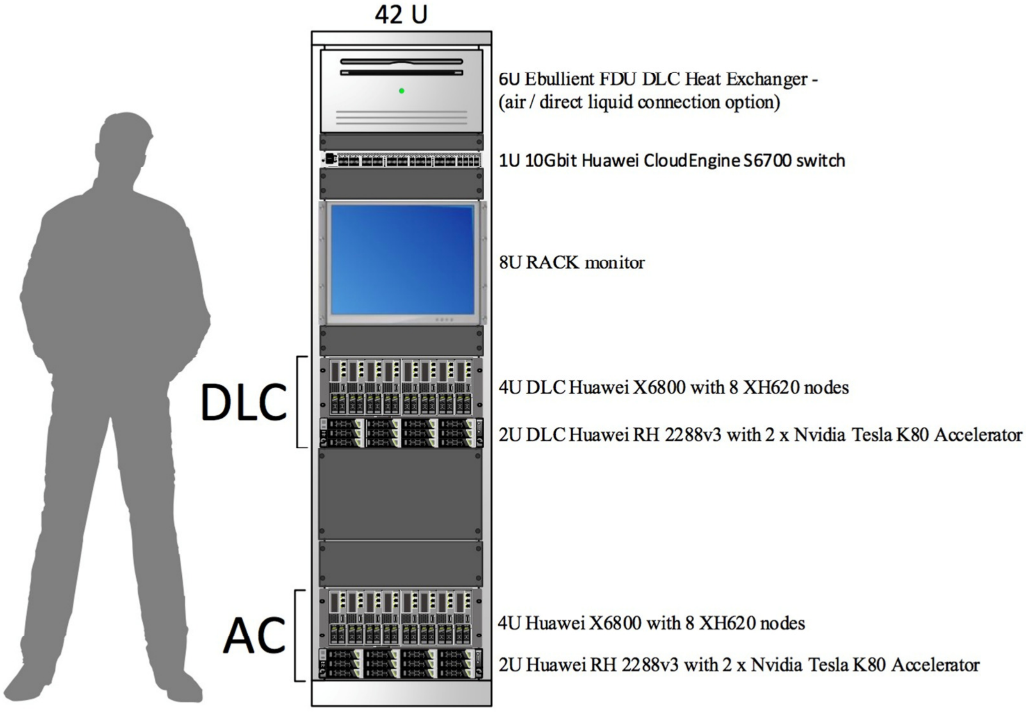


**Figure 5-13: PSNC DC: Rack equipment distribution**

Applied Ebullient FDU unit dissipates heat to the surrounding environment via a liquid-to-air heat exchanger. However, the applied solution allows also connecting the testbed to almost any external cooling facilities using SPT couplings.

**Figure 5-14: PSNC DC: Liquid to air exchanger**

Both setups are based on the HUAWEI FusionServer X6800. Each node comes with the following configuration:

- processor: Intel Xeon E5-2600 v3
- memory: 64GB DDR4, 2133MHz
- storage: 1TB, 7200RPM
- network: 10GbE

Moreover each setup is equipped with the addition server containing two Intel Xeon E5-2600 v3 processors and two GPU Accelerators based on the NVIDIA Tesla K80. Memory and storage configuration is similar to the one presented above.

Total number of the processing is presented in the table below:

| | Single Setup | Whole Setup |
|---|---|---|
| Nodes | 9 | 18 |
| CPUs | 18 | 36 |
| CPU cores | 144 | 288 |
| GPU units | 2 | 4 |

Table 27 – PSNC DC: Number of processors

**Multi-node efficient systems**

Another testing platform is based on the RECS system from the Christmann company. The test case system, called RECS, is a high density multi-node computer that consists of 18 single server nodes within one Rack Unit. To enable the user to have a fine-grained monitoring- and controlling-system, the RECS has a dedicated master-slave system of micro-controllers integrated that can gather different metrics directly without the need of polling every single node or the need of Operation System support. This enables us to gather many metrics like power usage, status and temperature for every node via only one request. The main advantage of this integrated monitoring system is to avoid the significant overhead, which would be caused by measuring and transferring data on operation system layer, which would consume lots of computing capabilities, particularly in a large-scale environment. Importantly, RECS can be equipped with diverse computing nodes ranging from high performance Intel i7 processors to Intel Atom CPUs or even embedded ARMs. The testbed consists of 6 RECS systems equipped with diverse kinds of CPUs: Intel i7, AMD Fusion, Intel Atom, and ARM as well as GPU and FPGA accelerators. In addition to RECS power usage and CPU temperature data server's inlets and outlets are equipped in temperature sensors. This testbed provides us with very detailed information about energy and thermal impact of workloads and applied management techniques.

#### 5.2.3.3. *Networking infrastructure*

Open Network Hardware Laboratory consists of ATCA platform – standardized network platform dedicated mostly to telecommunication operators, system designers and developers. ONH Laboratory in PSNC holds six ATCA chassis delivered by RadiSys Corporation.

The lab at PSNC is one of the eleven research laboratories designed, implemented and deployed during the Future-ICT project. The ONH Laboratory is part of PIONIER-LAB (Polish Research e-Infrastructure). The ONH Laboratory in PSNC holds six 40G ATCA nodes each of them equipped with several blades with different network processors (NPUs) and digital signal processors (DSPs). Full control over pass-through frames and packets provides opportunities to deliver real-time information about the network traffic including security monitoring aspects.

Each ATCA chassis in the ONH Lab has: two switch Ethernet blades with EZchip NP5 network processor and 100G ports; dual Intel Xeon E5-2600 blade with SAS disk; dual Cavium Octeon II CN68xx blade together with Cavium Octeon SDK; dual Broadcom XLP832 module with EZchip NP4 Mezzanine Card; media resource module with twenty Texas Instrument DSPs. The ATCA platform (where ATCA chassis are shown on Figure 5-15) is going to be used and programmed with set obbf tools, frameworks and Software Development Environments. Wirespeed packet classification together with layer 2-4 DPI Software Framework will support programmers during the implementation of advanced traffic monitoring system.

For the DOLFIN project, network performance measurement module will be developed providing real-time network statistics to help estimate costs of Data Centre migration.

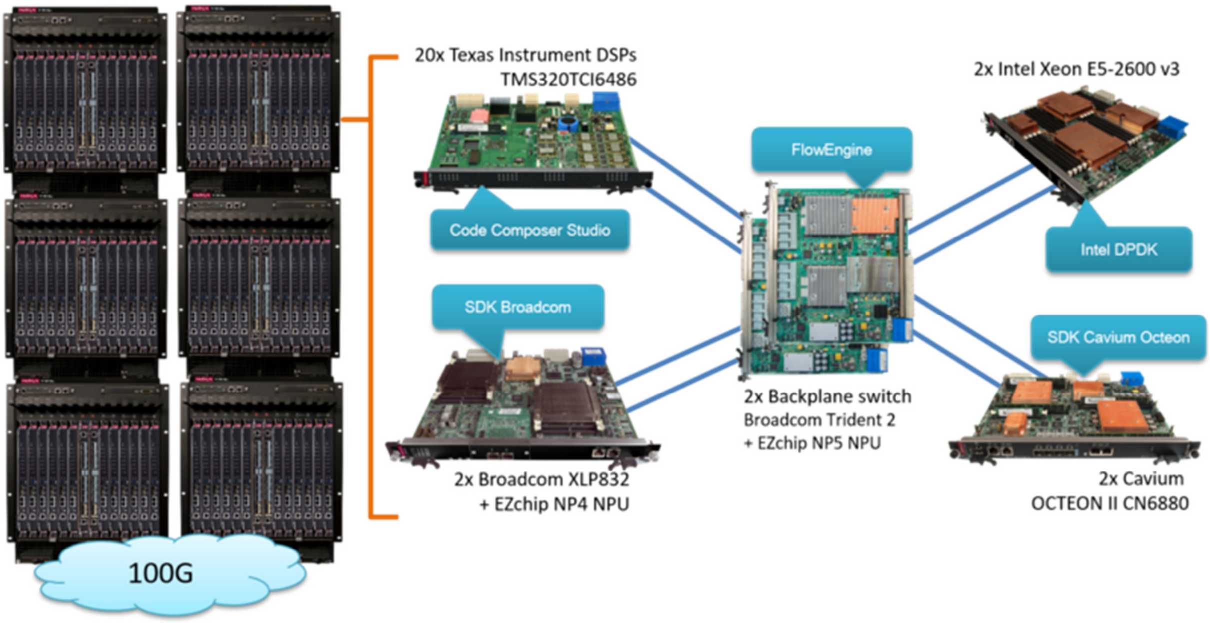


Figure 5-15: Open Network Hardware laboratory

#### 5.2.3.4. *Software architecture*

**Benchmarks and workloads**

It is possible to easily carry out a set of tests on the PSNC testbeds using installed tools. The Workload Executor System is a set of tools and services to run and control both single tasks and complex workloads on the hardware layer (physical testbed). The main aim of these tools is to facilitate the process of repeatable execution of HPC experiments on the testbed to gather information about responses of the system (mainly power consumption) to various combinations and configurations of hardware and applications running on it. Task submitted for the execution on the testbed are managed and controlled by the SLURM queueing system was configured to run jobs in exclusive mode (one job per node). Such configuration is often used for at least dedicated part of HPC resources.

**Virtualized infrastructure**

In the micro data centre (dedicated testbed) 3 servers (2 Intel Xeon CPUs each) will be equipped with VMware hypervisor software. The remaining servers (15 Intel Xeon-based, including direct liquid cooled, and around 30 other nodes) will be equipped with Xen/KVM virtualization software.

On top of that the OpenStack controller will be installed that will offer a wide range of configurations and settings, a stable virtualized environment, and the possibility to interact with the DOLFIN software components.

There will be two OpenStack instances in two locations:

- Actual installation of OpenStack in the DC (internal laboratory for PSNC, on top of VMware),
- OpenStack controller for the dedicated testbed (18 Intel Xeon-based and around 30 other servers) will be installed.

In the first round the following OpenStack components will be deployed:

- OpenStack Compute – Nova: this component allows the user to create and manage virtual servers using the machine images, and to provision large networks of virtual machines.
- OpenStack Networking – Neutron: the use of Neutron allows users to manage all the networking features including VLAN, Firewalls and IP addresses assignment.
- OpenStack Telemetry Service – Ceilometer: this tool provides efficient collection of metering data, in order to monitoring CPU and network resources.

In the next steps, if needed, Block Storage and Dashboard can be added.

#### 5.2.3.5. *Monitoring facilities*

PSNC testbed is equipped with a comprehensive set of sensors and meters enabling measurements of power usage, temperatures of inlet/outlet air/water, ambient temperatures, humidity, airflow, and other environmental parameters.

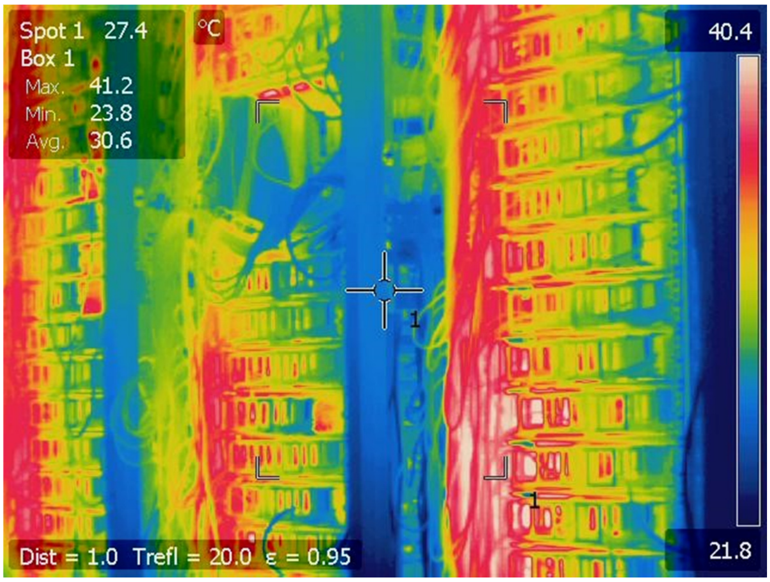


Figure 5-16: PSNC DC: Temperature monitoring

**Power and environmental measurements**

Apart from the internal server sensors, the whole system is monitored using external thermocouple meters measuring both the inlet and outlet temperatures with the single node granularity. The following data is measured and gathered within the testbed:

- power usage of particular nodes and the whole system,
- temperature of air at the inlet side of nodes,
- temperature of air at the outlet side of nodes,
- temperature of the liquid supplied to the system,
- temperature of the liquid living the system,
- temperature near the heat exchanger,
- infrared images of servers,
- airflow speed in selected locations (e.g., next to fans),
- humidity, and
- vibrations.

Moreover, software developed at PSNC allows following the power usage and temperature of particular processors, together with their P-States and utilization level.

Hardware and applications' related metrics and measurements are captured by NAGIOS and stored in a database.

**Benchmarking, Monitoring, Statistics**

Benchmarking Monitoring Statistics (BMS) is also installed to facilitate the process of running benchmarks (Benchmarking module), gathering system-wide and application-wide runtime data

(Monitoring module) and generating graphs and statistics from collected data (Statistics module). These three modules may be run independently. The suite is written in Python.

The benchmarking module allows to:

- schedule start of benchmarks at a specified date and time,
- schedule start of benchmarks when a specified list of benchmarks ends,
- control CPU frequency that is used when benchmarks run,
- limit the benchmarks duration to a specified amount of time,
- run benchmarks concurrently,
- specify a CPU usage limit for the benchmarks.

Any application available on the node that is tested may be run.

The goal of the monitoring module is to monitor selected values from the nodes and insert the values into the database. The data that may be monitored includes: CPU throttling frequency, CPU and node temperature, CPU load, CPU and node power consumption, values of performance counters, execution time. The sampling frequency may be specified.

The statistics module exploits the results generated by the benchmarking and the monitoring modules. It creates graphs from collected data, including changes of selected values in time as well as average values. The results are saved as PDF and PNG files.

There are different already configured benchmarks available for tests, all of them are described in Table 28.

| Application | Characteristics |
|---|---|
| Abinit | Widely used application for computational physics simulating systems made of electrons and nuclei to be calculated within density functional theory. |
| C-Ray | Ray-tracing benchmark that stresses floating point performance of a CPU. The default configuration file is called 'scene' at 16000x9000 resolution. |
| Linpack | Benchmark is used to evaluate system floating point performance. It is based on the Gaussian elimination methods that solve a dense N by N system of linear equations. |
| FFTE | Benchmark measures the floating-point arithmetic rate of double precision complex one-dimensional Discrete Fourier Transforms of 1-, 2-, and 3-dimensional sequences of length $2^p * 3^q * 5^r$. |

| | |
|---|---|
| OpenSSL | The OpenSSL application that ships with the OpenSSL libraries can perform a wide range of crypto operations and can be used as a benchmark for processors and memory. As a benchmark, it tests the signing and decoding of messages using several cryptographic algorithms such as MD5, SHA1, RSA. |
| Pybench | It offers a standardized way to measure the performance of Python implementations. In the past it has been used to track down performance bottlenecks or to demonstrate the impact of optimization and new features in Python. Pybench is a single-threaded application; it therefore places much smaller load on the processor. |

**Table 28 – Description of benchmarks already available on the PSNC testbed**

#### 5.2.3.6. *Summary*

The PSNC testbed and demonstration infrastructure consists of dedicated micro-DC, the PSNC DC, networking infrastructure, and energy management systems. They will allow to conduct experiments and demonstration with managing servers with respects to Demand Response signals, detailed monitoring and management of network connection, migration of workloads between dedicated testbed and the DC.

# 6. References

[1] DOLFIN, "Deliverable D2.1: Business scenarios and use case analysis," 2014.

[2] DOLFIN, "Deliverable D3.1: Data Centre energy consumption optimization platform (eCOP) Design," 2015.

[3] DOLFIN, "Deliverable D4.1: Synergetic Data Centres for energy efficiency (Design)," 2015.

[4] DOLFIN, "Deliverable D2.2: DOLFIN requirements and system architecture," 2014.

[5] C. Kaner, "An Introduction to Scenario Testing," June 2003. [Online]. Available: http://www.kaner.com/pdfs/ScenarioIntroVer4.pdf. [Accessed September 2015].

[6] "Openstack," [Online]. Available: https://www.openstack.org/.

[7] "OpenStack Configuration Reference," [Online]. Available: http://docs.openstack.org/kilo/config-reference/content/.

[8] "openADR Alliance," [Online]. Available: http://www.openadr.org/.

[9] "GEYSERS project," [Online]. Available: http://cordis.europa.eu/project/rcn/93786_en.html.

[10] "BONFIRE project," [Online]. Available: http://www.bonfire-project.eu/home.

[11] "OFERTIE project," [Online]. Available: http://www.ofertie.org/.

[12] "OFELIA project," [Online]. Available: http://www.fp7-ofelia.eu/.

# Appendix A. Test Measurement Definitions

In this section, we present a set of test measurements that can be used to define the outcome of each test scenario. A number of these me along with the respective targets can be used to clearly define each test scenario's objectives.

The metrics are organized in this section with regard to the pertaining DOLFIN feature. Each metric definition follows the schema outlined in Table 29 for clarity and easy reference.

| | |
|---|---|
| **Measurement ID:** | A globally unique identification of the measurement |
| **Name:** | A user-defined identification. |
| **Key feature:** | The key feature(s) addressed by this measurement |
| **Key actors:** | Specifies which (human and computer) actors can measure the test indicator |
| **Measurement unit:** | The unit used in the measurement |
| **Example value:** | An example value that can be measured |
| **Related use cases:** | All the Use Cases which can in part be verified with this measurement |
| **Assumptions:** | Particular preconditions that must be verified |

Table 29 – Measurements definition schema.

## A. Optimization of energy consumption

| | |
|---|---|
| **Measurement ID:** | A.1 |
| **Name:** | DOLFIN system energy consumption |
| **Key feature:** | A. Reduction of energy consumption |
| **Key actors:** | DC Operator<br>ICT Energy & Policy Supervisor<br>DOLFIN Workload Orchestrator<br>eCOP DB |

| | |
|---|---|
| | DOLFIN DB |
| **Measurement unit:** | kW |
| **Example value:** | 30,000kW |
| **Related use cases:** | 1.1, 1.2, 1.3, 2.1.1, 2.1.2 2.1.3, 2.2.1, 2.2.2, 2.2.3, 2.3.1, 2.3.2, 2.3.3, 3.1.1, 3.1.2, 3.1.3, 3.2.1, 3.2.2, 3.2.3, 3.3.1, 3.3.2, 7.3 |
| **Assumptions:** | Measurement of the total federated DOLFIN system energy consumption is performed on-demand by WP4 components. It reflects the sum of the participating DCs energy consumption rate. |

| | |
|---|---|
| **Measurement ID:** | A.1.1 |
| **Name:** | DOLFIN-enabled DC energy consumption |
| **Key feature:** | A. Reduction of energy consumption |
| **Key actors:** | DC Operator<br>ICT Energy & Policy Supervisor<br>eCOP DB |
| **Measurement unit:** | kW |
| **Example value:** | 10,000kW |
| **Related use cases:** | 1.1, 1.2, 1.3, 2.1.1, 2.1.2 2.1.3, 2.2.1, 2.2.2, 2.2.3, 2.3.1, 2.3.2, 2.3.3, 3.1.1, 3.1.2, 3.1.3, 3.2.1, 3.2.2, 3.2.3, 3.3.1, 3.3.2, 7.3 |
| **Assumptions:** | It reflects the sum of DC’s energy consumption rate as reported by the monitoring system. |

| | |
|---|---|
| **Measurement ID:** | A.1.1.1 |
| **Name:** | DC computing equipment energy consumption |
| **Key feature:** | A. Reduction of energy consumption |
| **Key actors:** | DC Operator<br>ICT Energy & Policy Supervisor<br>eCOP DB |
| **Measurement unit:** | kW |
| **Example value:** | 3,000kW |
| **Related use cases:** | 1.1, 1.2, 1.3, 2.1.1, 2.1.2 2.1.3, 2.2.1, 2.2.2, 2.2.3, 2.3.1, 2.3.2, 2.3.3, 3.1.1, 3.1.2, 3.1.3, 3.2.1, 3.2.2, 3.2.3, 3.3.1, 3.3.2, 7.3 |
| **Assumptions:** | It reflects the sum of DC’s computing equipment energy consumption rate as reported by the monitoring system. |

| | |
|---|---|
| **Measurement ID:** | A.1.1.2 |

| | |
|---|---|
| **Name:** | DC networking equipment energy consumption |
| **Key feature:** | A. Reduction of energy consumption |
| **Key actors:** | DC Operator<br>ICT Energy & Policy Supervisor<br>eCOP DB |
| **Measurement unit:** | kW |
| **Example value:** | 1,000kW |
| **Related use cases:** | 1.1, 1.2, 1.3, 2.1.1, 2.1.2 2.1.3, 2.2.1, 2.2.2, 2.2.3, 2.3.1, 2.3.2, 2.3.3, 3.1.1, 3.1.2, 3.1.3, 3.2.1, 3.2.2, 3.2.3, 3.3.1, 3.3.2, 7.3 |
| **Assumptions:** | It reflects the sum of DC's networking equipment energy consumption rate as reported by the monitoring system. |

| | |
|---|---|
| **Measurement ID:** | A.1.1.3 |
| **Name:** | DC cooling equipment energy consumption |
| **Key feature:** | A. Reduction of energy consumption |
| **Key actors:** | DC Operator<br>ICT Energy & Policy Supervisor<br>eCOP DB |
| **Measurement unit:** | kW |
| **Example value:** | 1,000kW |
| **Related use cases:** | 1.1, 1.2, 1.3, 2.1.1, 2.1.2 2.1.3, 2.2.1, 2.2.2, 2.2.3, 2.3.1, 2.3.2, 2.3.3, 3.1.1, 3.1.2, 3.1.3, 3.2.1, 3.2.2, 3.2.3, 3.3.1, 3.3.2, 7.3 |
| **Assumptions:** | It reflects the sum of DC's cooling equipment energy consumption rate (chillers etc.) as reported by the monitoring system. Free cooling components are not taken into account. |

| | |
|---|---|
| **Measurement ID:** | A.1.1.4 |
| **Name:** | DC general-purpose equipment energy consumption |
| **Key feature:** | A. Reduction of energy consumption |
| **Key actors:** | DC Operator<br>ICT Energy & Policy Supervisor<br>eCOP DB |
| **Measurement unit:** | kW |
| **Example value:** | 1,000kW |
| **Related use cases:** | 1.1, 1.2, 1.3, 2.1.1, 2.1.2 2.1.3, 2.2.1, 2.2.2, 2.2.3, 2.3.1, 2.3.2, 2.3.3, 3.1.1, |

| | |
|---|---|
| | 3.1.2, 3.1.3, 3.2.1, 3.2.2, 3.2.3, 3.3.1, 3.3.2, 7.3 |
| **Assumptions:** | It reflects the sum of DC's all other equipment energy consumption rate as reported by the monitoring system. |

| | |
|---|---|
| **Measurement ID:** | A.1.1.5 |
| **Name:** | DC area equipment energy consumption |
| **Key feature:** | A. Reduction of energy consumption |
| **Key actors:** | DC Operator<br>ICT Energy & Policy Supervisor<br>eCOP DB |
| **Measurement unit:** | kW |
| **Example value:** | 1,000 kW |
| **Related use cases:** | 1.1, 1.2, 1.3, 2.1.1, 2.1.2 2.1.3, 2.2.1, 2.2.2, 2.2.3, 2.3.1, 2.3.2, 2.3.3, 3.1.1, 3.1.2, 3.1.3, 3.2.1, 3.2.2, 3.2.3, 3.3.1, 3.3.2, 7.3 |
| **Assumptions:** | It reflects the sum of a DC's area (floor, room, or other applicable spatial subdivision) energy consumption rate as reported by the monitoring system. |

| | |
|---|---|
| **Measurement ID:** | A.1.1.6 |
| **Name:** | DC area computing equipment energy consumption |
| **Key feature:** | A. Reduction of energy consumption |
| **Key actors:** | DC Operator<br>ICT Energy & Policy Supervisor<br>eCOP DB |
| **Measurement unit:** | kW |
| **Example value:** | 1,000kW |
| **Related use cases:** | 1.1, 1.2, 1.3, 2.1.1, 2.1.2 2.1.3, 2.2.1, 2.2.2, 2.2.3, 2.3.1, 2.3.2, 2.3.3, 3.1.1, 3.1.2, 3.1.3, 3.2.1, 3.2.2, 3.2.3, 3.3.1, 3.3.2, 7.3 |
| **Assumptions:** | It reflects the sum of a DC's area computing equipment energy consumption rate as reported by the monitoring system. |

| | |
|---|---|
| **Measurement ID:** | A.1.1.7 |
| **Name:** | DC area networking equipment energy consumption |
| **Key feature:** | A. Reduction of energy consumption |
| **Key actors:** | DC Operator<br>ICT Energy & Policy Supervisor |

| | |
|---|---|
| | eCOP DB |
| **Measurement unit:** | kW |
| **Example value:** | 1,000kW |
| **Related use cases:** | 1.1, 1.2, 1.3, 2.1.1, 2.1.2 2.1.3, 2.2.1, 2.2.2, 2.2.3, 2.3.1, 2.3.2, 2.3.3, 3.1.1, 3.1.2, 3.1.3, 3.2.1, 3.2.2, 3.2.3, 3.3.1, 3.3.2, 7.3 |
| **Assumptions:** | It reflects the sum of a DC's area networking equipment energy consumption rate as reported by the monitoring system. |

| | |
|---|---|
| **Measurement ID:** | A.1.1.8 |
| **Name:** | DC area cooling equipment energy consumption |
| **Key feature:** | A. Reduction of energy consumption |
| **Key actors:** | DC Operator<br>ICT Energy & Policy Supervisor<br>eCOP DB |
| **Measurement unit:** | kW |
| **Example value:** | 1,000kW |
| **Related use cases:** | 1.1, 1.2, 1.3, 2.1.1, 2.1.2 2.1.3, 2.2.1, 2.2.2, 2.2.3, 2.3.1, 2.3.2, 2.3.3, 3.1.1, 3.1.2, 3.1.3, 3.2.1, 3.2.2, 3.2.3, 3.3.1, 3.3.2, 7.3 |
| **Assumptions:** | It reflects the sum of a DC's area cooling equipment energy consumption rate as reported by the monitoring system. |

| | |
|---|---|
| **Measurement ID:** | A.1.1.9 |
| **Name:** | DC area general-purpose equipment energy consumption |
| **Key feature:** | A. Reduction of energy consumption |
| **Key actors:** | DC Operator<br>ICT Energy & Policy Supervisor<br>eCOP DB |
| **Measurement unit:** | kW |
| **Example value:** | 1,000kW |
| **Related use cases:** | 1.1, 1.2, 1.3, 2.1.1, 2.1.2 2.1.3, 2.2.1, 2.2.2, 2.2.3, 2.3.1, 2.3.2, 2.3.3, 3.1.1, 3.1.2, 3.1.3, 3.2.1, 3.2.2, 3.2.3, 3.3.1, 3.3.2, 7.3 |
| **Assumptions:** | It reflects the sum of a DC's area general-purpose equipment energy consumption rate as reported by the monitoring system. |

| | |
|---|---|
| **Measurement ID:** | A.1.2 |
| **Name:** | DOLFIN federation layer energy consumption |

| | |
|---|---|
| **Key feature:** | A. Reduction of energy consumption |
| **Key actors:** | DOLFIN Workload Orchestrator<br>DOLFIN DB |
| **Measurement unit:** | kW |
| **Example value:** | 1kW |
| **Related use cases:** | 1.2, 1.3, 2.1.2, 2.1.3, 2.2.2, 2.2.3, 2.3.2, 2.3.3, 3.1.2, 3.1.3, 3.2.2, 3.2.3, 3.3.2, 7.3 |
| **Assumptions:** | It reflects the sum of energy consumption as a result of DOLFIN SW/HW component operation as reported by the monitoring system. |

| | |
|---|---|
| **Measurement ID:** | A.2 |
| **Name:** | DOLFIN system energy costs |
| **Key feature:** | A. Reduction of energy consumption |
| **Key actors:** | DC Operator<br>ICT Energy & Policy Supervisor<br>DOLFIN Workload Orchestrator<br>eCOP DB<br>DOLFIN DB |
| **Measurement unit:** | Euro(s) |
| **Example value:** | 1,000 euros |
| **Related use cases:** | 1.1, 1.2, 1.3, 2.1.1, 2.1.2 2.1.3, 2.2.1, 2.2.2, 2.2.3, 2.3.1, 2.3.2, 2.3.3, 3.1.1, 3.1.2, 3.1.3, 3.2.1, 3.2.2, 3.2.3, 3.3.1, 3.3.2, 7.3 |
| **Assumptions:** | It reflects the sum of the energy expenditure of all the participating DCs in the DOLFIN system. |

| | |
|---|---|
| **Measurement ID:** | A.2.1 |
| **Name:** | DOLFIN-enabled DC energy costs |
| **Key feature:** | A. Reduction of energy consumption |
| **Key actors:** | DC Operator<br>ICT Energy & Policy Supervisor<br>DOLFIN Workload Orchestrator<br>eCOP DB<br>DOLFIN DB |
| **Measurement unit:** | Euro(s) |
| **Example value:** | 1,000 euros |

| | |
|---|---|
| **Related use cases:** | 1.1, 1.2, 1.3, 2.1.1, 2.1.2 2.1.3, 2.2.1, 2.2.2, 2.2.3, 2.3.1, 2.3.2, 2.3.3, 3.1.1, 3.1.2, 3.1.3, 3.2.1, 3.2.2, 3.2.3, 3.3.1, 3.3.2, 7.3 |
| **Assumptions:** | It reflects the energy expenditure of the DC participating in the DOLFIN system. |

## B. Temporal and geographical shifting of computer processing

| | |
|---|---|
| **Measurement ID:** | B.1 |
| **Name:** | DOLFIN system Cross-DC VM migration operations |
| **Key feature:** | B. Temporal and geographical shifting of computer processing |
| **Key actors:** | DC Operator<br>ICT Energy & Policy Supervisor<br>DOLFIN Workload Orchestrator<br>eCOP DB<br>DOLFIN DB |
| **Measurement unit:** | # of Cross-DC VM migration operations |
| **Example value:** | 1,000 operations |
| **Related use cases:** | 1.2, 1.3, 2.1.2, 2.1.3, 2.2.2, 2.2.3, 2.3.2, 2.3.3, 3.1.2, 3.1.3, 3.2.2, 3.2.3, 3.3.2, 7.3 |
| **Assumptions:** | The number of Cross-DC VM migrations in the relevant time period. |

| | |
|---|---|
| **Measurement ID:** | B.2 |
| **Name:** | DOLFIN system workload time-shifting operations |
| **Key feature:** | B. Temporal and geographical shifting of computer processing |
| **Key actors:** | DC Operator<br>ICT Energy & Policy Supervisor<br>DOLFIN Workload Orchestrator<br>eCOP DB<br>DOLFIN DB |
| **Measurement unit:** | # of workload time-shifting operations |
| **Example value:** | 1,000 operations |
| **Related use cases:** | 2.1.1, 2.1.2, 2.1.3, 2.2.1, 2.2.2, 2.2.3, 2.3.1, 2.3.2, 2.3.3, 3.1.1, 3.1.2, 3.1.3, 3.2.1, 3.2.2, 3.2.3, 3.3.1, 3.3.2, 7.3 |

| | |
|---|---|
| **Assumptions:** | The number workload time-shifting operations in the relevant time period. |

## C. Responsiveness to energy grid price and power fluctuations

| | |
|---|---|
| **Measurement ID:** | C.1 |
| **Name:** | DOLFIN system $CO_2$ footprint |
| **Key feature:** | C. Responsiveness to energy grid price and power fluctuations |
| **Key actors:** | DC Operator<br>ICT Energy & Policy Supervisor<br>DOLFIN Workload Orchestrator<br>eCOP DB<br>DOLFIN DB |
| **Measurement unit:** | $CO_2$ equivalent |
| **Example value:** | 15 tonnes of $CO_2$e |
| **Related use cases:** | 3.1.1, 3.1.2, 3.1.3 |
| **Assumptions:** | The sum of $CO_2$ equivalent for the DOLFIN system. Calculated as the sum of $CO_2$ equivalent for each participating DC. |

| | |
|---|---|
| **Measurement ID:** | C.1.1 |
| **Name:** | DOLFIN-enabled DC $CO_2$ footprint |
| **Key feature:** | C. Responsiveness to energy grid price and power fluctuations |
| **Key actors:** | DC Operator<br>ICT Energy & Policy Supervisor<br>DOLFIN Workload Orchestrator<br>eCOP DB<br>DOLFIN DB |
| **Measurement unit:** | $CO_2$ equivalent |
| **Example value:** | 7 tonnes of $CO_2$e |
| **Related use cases:** | 3.1.1, 3.1.2, 3.1.3 |
| **Assumptions:** | The amount of $CO_2$ equivalent for the DOLFIN-enabled DC, calculated dependant on the energy consumption over a time period and the energy mix as reported by the energy provider. |

| | |
|---|---|
| **Measurement ID:** | C.2 |
| **Name:** | DOLFIN user $CO_2$ footprint |
| **Key feature:** | C. Responsiveness to energy grid price and power fluctuations |
| **Key actors:** | Customer<br>DC Operator<br>ICT Energy & Policy Supervisor<br>DOLFIN Workload Orchestrator<br>eCOP DB<br>DOLFIN DB |
| **Measurement unit:** | $CO_2$ equivalent |
| **Example value:** | 0.01 tonnes of CO2e |
| **Related use cases:** | 3.1.1, 3.1.2, 3.1.3 |
| **Assumptions:** | The amount of $CO_2$ equivalent for the DOLFIN user, calculated dependant on the total energy consumption of the user's VMs over a time period and the energy mix as reported by the energy provider. |

| | |
|---|---|
| **Measurement ID:** | C.3 |
| **Name:** | DOLFIN customer base |
| **Key feature:** | C. Responsiveness to energy grid price and power fluctuations |
| **Key actors:** | Customer<br>DC Operator<br>DOLFIN DB |
| **Measurement unit:** | # of customers served by DOLFIN-enabled DCs |
| **Example value:** | 1000 customers |
| **Related use cases:** | 3.1.1, 3.1.2, 3.1.3 |
| **Assumptions:** | The size of DOLFIN's customer base as a concrete measure of brand recognition, user satisfaction and environmentally-friendly computing demand. |

| | |
|---|---|
| **Measurement ID:** | C.4 |
| **Name:** | DOLFIN system direct incentives to lower peak load |
| **Key feature:** | C. Responsiveness to energy grid price and power fluctuations |

| | |
|---|---|
| **Key actors:** | DOLFIN Operator<br>ICT Energy & Policy Supervisor<br>DOLFIN Workload Orchestrator<br>eCOP DB<br>DOLFIN DB |
| **Measurement unit:** | Euro(s) |
| **Example value:** | 1,000 euros |
| **Related use cases:** | 3.2.1, 3.2.2, 3.2.3, 3.3.1, 3.3.2, 7.3 |
| **Assumptions:** | Some (or all) of the DCs participating in the DOLFIN network have signed agreements with their regional EPs or have direct incentives from other sources to lower energy consumption during specific time periods. |

| | |
|---|---|
| **Measurement ID:** | C.4.1 |
| **Name:** | DOLFIN-enabled DC direct incentives to lower peak load |
| **Key feature:** | C. Responsiveness to energy grid price and power fluctuations |
| **Key actors:** | DC Operator<br>ICT Energy & Policy Supervisor<br>DOLFIN Workload Orchestrator<br>eCOP DB<br>DOLFIN DB |
| **Measurement unit:** | Euro(s) |
| **Example value:** | 1,000 euros |
| **Related use cases:** | 3.2.1, 3.2.2, 3.2.3, 3.3.1, 3.3.2, 7.3 |
| **Assumptions:** | The DOLFIN-enabled DC has signed an agreement with the regional EP or receives direct incentives from other sources to lower energy consumption during specific time periods. |

## D. Ensuring DC user experience by defining and re-negotiating SLAs

| | |
|---|---|
| **Measurement ID:** | D.1 |
| **Name:** | DOLFIN system SLA violations |
| **Key feature:** | D. Ensuring DC user experience by defining and re-negotiating SLAs |
| **Key actors:** | User |

| | |
|---|---|
| | DC Operator<br>ICT Energy & Policy Supervisor<br>DOLFIN Workload Orchestrator<br>eCOP DB<br>DOLFIN DB |
| **Measurement unit:** | # of SLA violations incurred (SLAV) / billing period (month) |
| **Example value:** | 100 SLAV/ month |
| **Related use cases:** | 1.1, 1.2, 1.3, 2.1.1, 2.1.2 2.1.3, 2.2.1, 2.2.2, 2.2.3, 2.3.1, 2.3.2, 2.3.3, 3.1.1, 3.1.2, 3.1.3, 3.2.1, 3.2.2, 3.2.3, 3.3.1, 3.3.2, 7.3 |
| **Assumptions:** | This indicator provides the absolute number of SLA violation instances that were incurred over the billing period. |

| | |
|---|---|
| **Measurement ID:** | D.2 |
| **Name:** | DOLFIN system penalties from SLA violations |
| **Key feature:** | D. Ensuring DC user experience by defining and re-negotiating SLAs |
| **Key actors:** | Customer<br>DC Operator<br>ICT Energy & Policy Supervisor<br>DOLFIN Workload Orchestrator<br>eCOP DB<br>DOLFIN DB |
| **Measurement unit:** | Euro(s) |
| **Example value:** | 100 euro(s) |
| **Related use cases:** | 1.1, 1.2, 1.3, 2.1.1, 2.1.2 2.1.3, 2.2.1, 2.2.2, 2.2.3, 2.3.1, 2.3.2, 2.3.3, 3.1.1, 3.1.2, 3.1.3, 3.2.1, 3.2.2, 3.2.3, 3.3.1, 3.3.2, 7.3 |
| **Assumptions:** | This value represents the financial cost of SLA violations for the DC operators. It measures the total cost of penalties levied and price reductions offered to users for SLA violations. This could be offset from energy savings and/or other monetary incentives. |

| | |
|---|---|
| **Measurement ID:** | D.3 |
| **Name:** | DOLFIN system SLA renegotiations |
| **Key feature:** | D. Ensuring DC user experience by defining and re-negotiating SLAs |

| | |
|---|---|
| **Key actors:** | Customer<br>DC Operator<br>ICT Energy & Policy Supervisor<br>DOLFIN Workload Orchestrator<br>eCOP DB<br>DOLFIN DB |
| **Measurement unit:** | # of SLAs renegotiated / month |
| **Example value:** | 100 SLAs renegotiated |
| **Related use cases:** | 2.3.1, 2.3.2, 2.3.3 |
| **Assumptions:** | This indicator provides the number of SLAs that were successfully renegotiated between the DOLFIN system and the customers. |