Complete Dynamic Multi-cloud Application Management

Project no. 644925

Innovation Action

Co-funded by the Horizon 2020 Framework Programme of the European Union

Call identifier: H2020-ICT-2014-1

Topic: ICT-07-2014 – Advanced Cloud Infrastructures and Services

Start date of project: January 1st, 2015 (36 months duration)

Deliverable D7.1

Description of testbed

Due date:	31/07/2015
Submission date:	07/09/2015
Deliverable leader:	QSC
Editors list:	D. Hacker (QSC)

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	Short Name	Contributor
Interoute S.P.A.	IRT	Domenico Gallico, Matteo Biancani
Sixsq SARL	SIXSQ	Charles Loomis
QSC AG	QSC	Doris Hacker, Maria Kourkouli
Technische Universitaet Berlin	TUB	Mathias Slawik
Fundacio Privada I2CAT, Internet I Innovacio Digital A Catalunya	I2CAT	Jose Aznar, Eduard Escalona
Universiteit Van Amsterdam	UVA	Miroslav Zivkovic
Centre National De La Recherche Scientifique	CNRS	Oleg Lodygensky

Change history

Version	Date	Partners	Description/Comments
0.1	30/6/2015	QSC	Initial Creation
0.2	17/7/2015	QSC	Section Use Case view
0.3	23/7/2015	IRT	Integration input from IRT
0.4	30/7/2015	QSC	First version for internal review
0.5	15/8/2015	I2CAT	Reviewed version
0.6	31/8/2015	ALL	Integrate comments on previous version
1.0-beta	31/8/2015	I2CAT	Final reviewed version
1.0-final	4/9/2015	I2CAT, IRT	Final reviewed for submission

Table of Contents

List of Figures

Executive Summary

CYCLONE's software and tools aim to facilitate the deployment, management and use of complex multi-cloud applications and to enhance the end-to-end security of those applications. The provided tools with the appropriate extensions will support the configuration of the cloud applications, its automated deployment onto cloud infrastructures, and control of the allocated cloud resources.

To reach the holistic approach of CYCLONE, a number of existing core components will be adapted, combined and improved concerning to needs identified by the use case requirements. Using an agile approach for the software development process, the distributed testbed is used to deploy a working version of the software components and to get a rapid and detailed feedback.

For the testing purpose of these tools and software and as a reference implementation, a testbed is necessary to be set up and deployed.

The CYCLONE testbed will play a critical role, enabling the full, continuous validation of the software before the foreseen production deployments on a federated cloud.

The initial set of requirements and features of the testbed was collected through discussions with the CYCLONE partners; both the cloud application developers based on the defined use cases and the developers of the CYCLONE components. The main section of this document concerns the architectural and technical view of the distributed testbed that will provide infrastructure-as-a-Service (IaaS).

The document addresses each of the following objectives:

1. Initial definition of the CYCLONE testbed, based on the selected CYCLONE use cases and from the CYCLONE tool developers
2. Collection and evaluation of required software tools that will be used in the testbed
3. Architectural and technical view of the distributed testbed
4. Determination of the testbed features and functionalities using infrastructure as a service

The flexible architecture and setup of the CYCLONE testbed provides an initial set of core features. Based on the open structure, the testbed can be extended and adapted to further needs during the project.

1. Introduction

1.1. Purpose of this deliverable

Cloud computing represents a paradigm shift in the way applications are developed, deployed, maintained and consumed in recent years. Cloud computing, having a pivotal role in the field of Information and Communication Technology (ICT), represents the last generation infrastructure and application hosting service and delivery model. Because of their elasticity, flexibility, and dynamic resource provisioning, cloud infrastructures as a service is widely used by the academic and commercial service providers. These application service providers have moved beyond simple, single-machine applications and now develop complex computing platforms that are deployed and maintained within the cloud, which is more and more multi-cloud environment. CYCLONE facilitates the management of multi-cloud applications by hiding the complexity of the underlying system.

In the project CYCLONE a testbed is provided in order to enable the developers to rapidly prototype their complex cloud applications. The testbed is a platform used for testing and improving new developed components, including software, and networking components.

This document provides a description of the CYCLONE testbed and specifies the architecture of the testbed matching its components with the deployment requirements coming from the various use cases. The testbed architecture will support a high degree of flexibility in the architecture so that emerging demands of developers can be satisfied.

The testbed architecture is needed to guide the implementation of the CYCLONE use cases environment, and will consist of the current and future available and software components contributed from CYCLONE partners and its integration based upon the developed architectural principles of the CYCLONE community. The testbed allows CYCLONE members to integrate and test their state of the art solution and the application service providers of the CYCLONE project will be able to control the deployment of their applications.

The CYCLONE testbed will provide infrastructure-as-a-Service (IaaS) in form of an overlay on top of a cloud computing platform network that will feature a diversity of installed systems and hardware in order to reach a high level of flexibility. The suggested testbed should contribute to test and improve the developed components in CYCLONE. It aims to provide a rapid prototyping of complex cloud applications and to give continuous feedback to the developers of the different CYCLONE tools.

Figure 1-1 depicts the overview of CYCLONE tools and software and how application service providers will take advantage of the innovative platform. The distributed CYCLONE testbed is designed to support the development of the needed features.

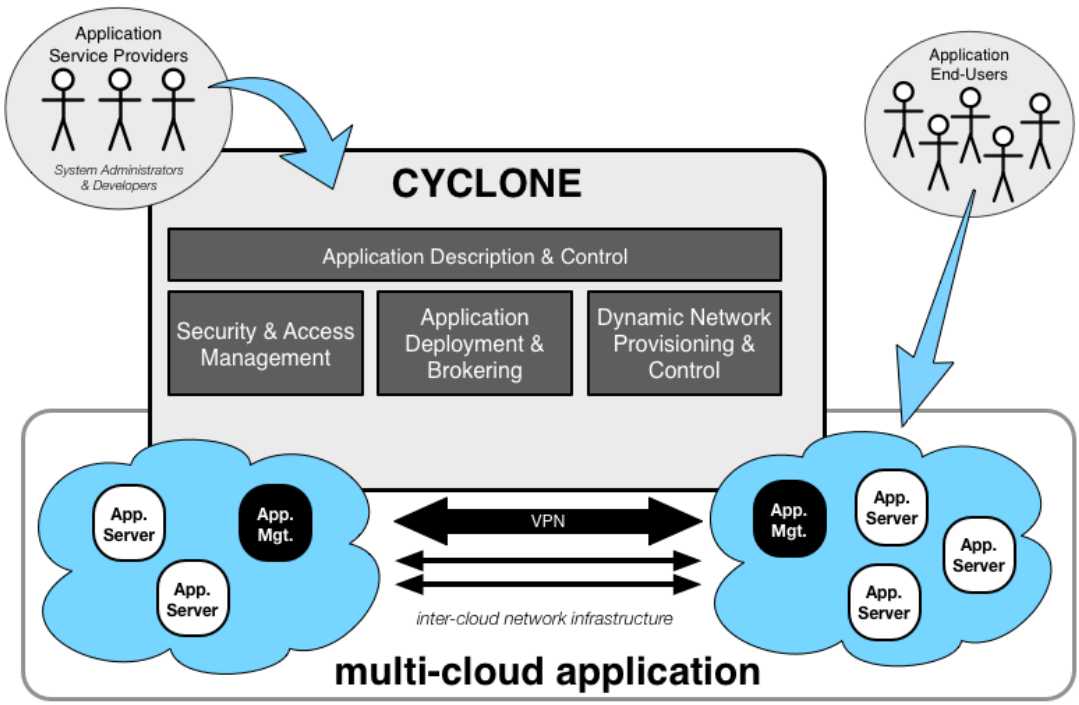

Figure 1-1 Overview of CYCLONE tools and software

1.2. Expected outcome of the CYCLONE testbed

The described CYCLONE testbed targets to present a reference implementation of a multi-cloud computing environment, allowing the user to use service interfaces to launch instances with a variety of operating systems, load them with his custom application environment, manage the network's access permissions, and run his image using as many or few systems as he desires. The testbed should guarantee a modular system, which is flexible, scalable, virtualized and automated. The testbed will guarantee services such as: elasticity, reliability, and security.

Elasticity: it means that the user should be able to increase or decrease capacity within a short time. He can commission multiple server instances simultaneously. As all are controlled with APIs, the application can be automatically scaled up and down depending on its needs.

Reliability: it will offer a highly reliable environment where replacement instances can be rapidly and predictably commissioned.

Security: it should provide mechanisms for securing the computer resources. It will include interfaces to configure firewall settings that control network access to and between groups of instances. Moreover, through the access management for cloud solutions different functionality, such as authentication, authorization, accounting, logging, and monitoring should be ensured.

The CYCLONE testbed is aimed to allow all the users to rapidly prototype their complex cloud applications and to provide continuous feedback to the developers, thus validating the CYCLONE software's design, implementation, and production quality.

Given the very heterogeneous nature of the use cases the testbed aggregates in a flexible manner all the required technologies. To this extent the testbed will include software, hardware, and networking elements that will allow the testing of distributed cloud applications. The testbed aims to permit the easy rapid deployment of both test and production instances of multi-cloud applications, encouraging the developers of those applications to similarly take advantage of rapid prototyping and feedback from their users.

This testbed will facilitate the deployment of the cloud infrastructure management tools and cloud services developed by the consortium partners, enabling their integration and testing. It will be deployed in a distributed fashion to satisfy the project requirements.

Therefore, the CYCLONE testbed has the following purposes:

1. Support the development of an IaaS cloud architecture that works as a distributed federated cloud and uses network virtualization approaches to address efficient multi-tenancy and network resources utilization.
2. Maintain the flexibility in the architecture so that emerging demands of developers can be satisfied and the solution could work as a testbed.
3. Support the testing of CYCLONE software and the validation of the proposed use cases.

2. Architectural overview of the testbed

2.1. CYCLONE testbed overview

Based on the architectural and technical view of the testbed, an abstract design of the implementation of the initial setup is depicted in Figure 2-1, which includes CYCLONE applications, SW modules, a Virtualization Layer (virtualized resources) and the HW Infrastructure (network connections, CPUs, storage).

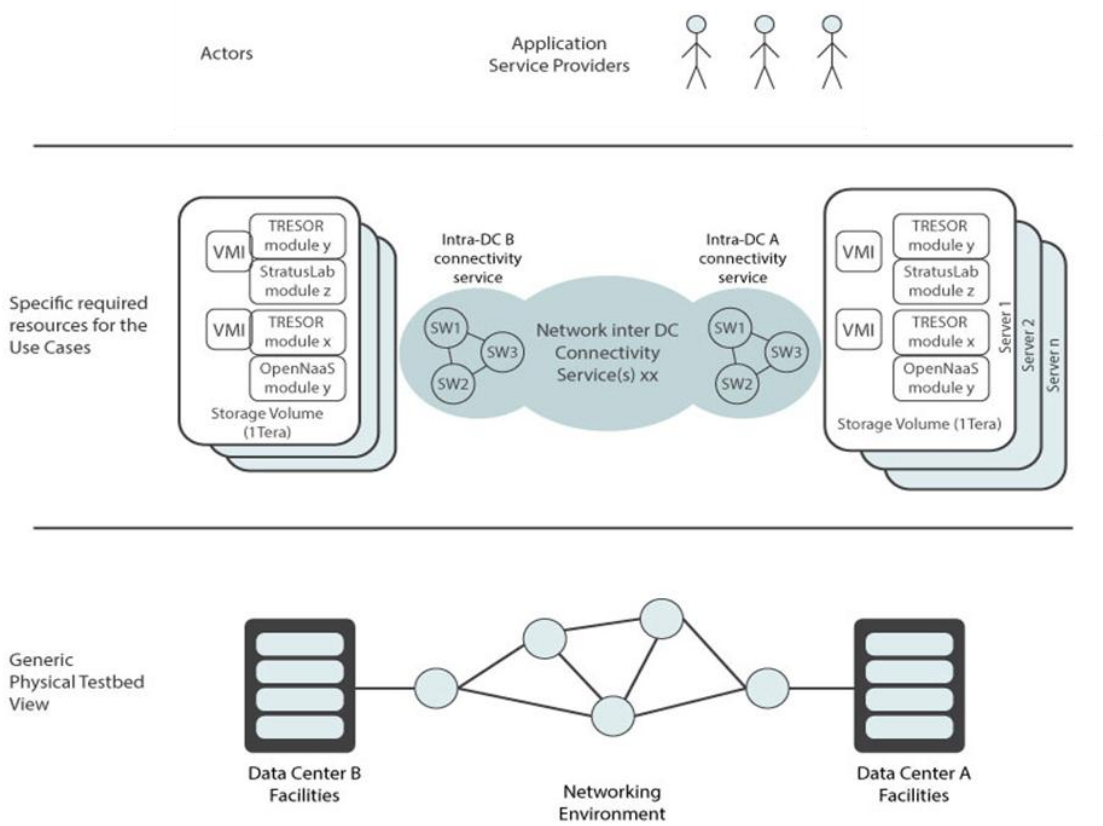

Figure 2-1 Outline of the CYCLONE testbed architecture

This scheme includes the view of the topology, the available infrastructure and the network facilities, the SW modules present at each VM/server and the involved actors. It represents a full view scenario from the pure infrastructure layer, to the workflow of the use case.

2.2. CYCLONE testbed infrastructure

The initial testbed infrastructure is composed by three geographically distributed testbeds located at CNRS/LAL, IRT and QSC AC premises, interconnected through the public internet to demonstrate a real scenario. Further enhancements to the testbed will consider interconnectivity options and additional resources and domains as described later in Section 3. The testbed is designed to be easily extended for

supporting future requirements from the involved members and new users. The different testbeds will include computing, storage and network resources, some of them located in dedicated data centre facilities. The testbed infrastructure will take direct advantage of the physical servers or can be virtualized on top of them according to specific needs. The hardware and administrative services needed to store applications and a platform for running applications are provided by each infrastructure owner. Figure 2-2 depicts an overall picture of the initial testbed facilities location.

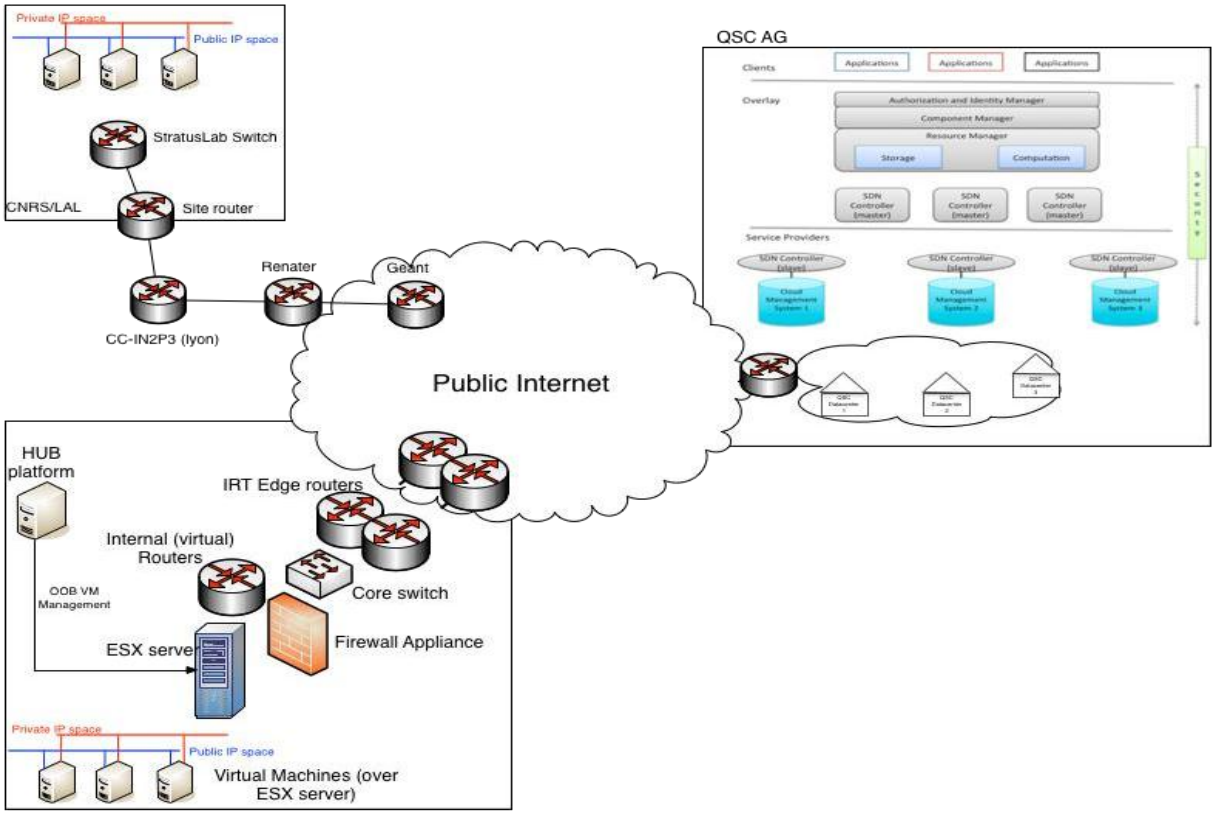

Figure 2-2 CYCLONE testbed overview

With the chosen testbed architecture, developers and application service providers in CYCLONE can provision processing, storage, networks, and other fundamental computing resources, as well as deploy and run arbitrary software, which can include operating systems and applications. The user does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications, and possibly limited control of selected networking components (e.g., firewalls and load-balancers).

In essence, the CYCLONE testbed will be composed of compute, storage and network resources.

- *Compute:* Compute servers are the power of the testbed. They are the servers on which the virtual machines are created. It provides resizable compute capacity in the cloud. It allows obtaining and configuring capacity with minimal friction. It gives complete control of the computing resources and lets the applications run on CYCLONE proven computing environment. The compute module encompasses physical assets such as processors, memory.
- *Storage:* Another component is the storage nodes. The format of storage is independent of the type of nodes and the controller can manage it. In order to consolidate the storage for maximum efficiency it is separated from the computation module. It is essential that storage capacity is not limited to storage devices that are directly connected to a particular server because this prevents

the ability to scale processing capacity on demand, to scale storage capacity beyond the limits of a few devices, to provide VM mobility to other facilities.

- *Network*: Given that the testbed includes multiple servers, the ability for them to communicate with each other and with the outside world is crucial. Network nodes constitute another unit of the testbed. They are used to connect the cloud with the external world, such as the public Internet.

Crucial parts of the testbed are the different components that compose CYCLONE software and that will be deployed over the testbed infrastructure. These components provide the services supported by CYCLONE such as application management services or authentication and federation services. The deployment requirements of these components are briefly described in Section 4. During the progress of the project further testbed evolutions should be realized to be able to deploy and implement new use cases.

In the next sections each one of the domains forming the CYCLONE testbed is described.

2.2.1. QSC AG Testbed

The testbed infrastructure provided by QSC is located in their data centre in Hamburg. Their Infrastructure as a Service (IaaS) offering includes the delivery of hardware (server, storage and network), and associated software (operating systems, virtualization technology, file system) and allows users to provision resources on demand. QSC's main management tasks should be basically to keep the data centre operational, while users will deploy and manage the software services themselves just the way they would in their own data centre.

QSC's data centre is deployed as a layered architecture representing applications, a cloud manager and a network control layer. CYCLONE's QSC domain can be configured as an overlay on top of QSC's data centre infrastructure, which uses OpenStack as cloud manager.

The overall specification of the server's configuration used for the purposes of CYCLONE is as follows:

- 16 x 8 Core CPU
- 16 to 32 GB of Memory
- 42 TB Storage

Moreover, for the internet connection a 10 Gbps connection is provided for WAN and LAN.

2.2.2. LAL Testbed

LAL provides a testbed infrastructure based on StratusLab middleware.

StratusLab aims to deploy an IaaS cloud infrastructure where users can create and manage their own virtual machines (VM) and disks (VD), based on an image management infrastructure (the Marketplace).

StratusLab proposes a distributed architecture composed of a distributed client and a set of centralized services. This is the commonly used architecture for cloud infrastructures and is shown on figure 2-3:

- A Persistent Disk service: This service allows user to administrate and use disk volumes.
- A computing service: This service allows user to administrate and use virtual machines. This service is currently provided by 2 daemons, one_proxy to authenticate users and OpenNebula to run and manage VMs.
- One metadata management service: the *Marketplace* aims to provide a simple way to share and distribute Images. Each endorser pushes to the Marketplace a manifest, which is a metadata representation of his machine.
- Distributed clients deployed over the internet: Clients are installed by users (scientists, for example) on their own IT resources (cluster; personal computers). The client permits users to interact with the infrastructure (create VM and VD; start and stop VM etc.)

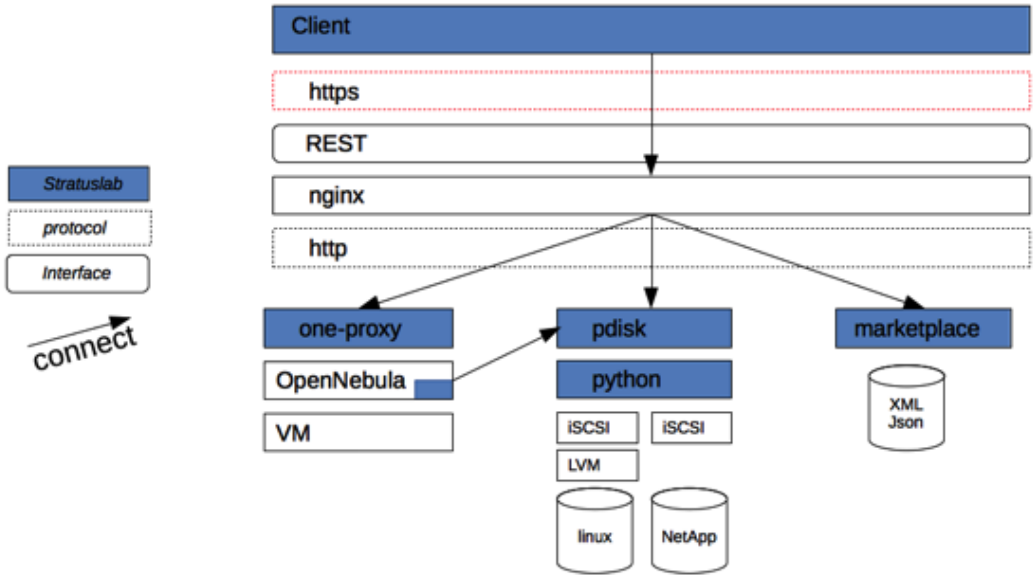

Figure 2-3 StratusLab architecture

The StratusLab deployment at LAL instantiates includes the three StratusLab services:

- *one_proxy*, the authentication service, is available at https://cloud.lal.stratuslab.eu/
- *PDisk*, the VD service management, is available at https://pdisk.lal.stratuslab.eu/
- Marketplace, the image management, is available at https://marketplace.stratuslab.eu/marketplace

The LAL testbed data centre will be composed of up to 32 machines with the following roles and characteristics (Figure 2-4):

Machine	Role	OS	CPU
onehost-2	test	CentOS 7	
onehost-4	Frontend VM (opennebula)	Scientific Linux 6	
onehost-5	test (Jenkins)	CentOS 7	
onehost-6	test (Jenkins)	CentOS 7	
onehost-7	Persistent Disk	Scientific Linux 6	About 10 TB
onehost-8	Marketplace	Scientific Linux 6	
onehost-10	test	Scientific Linux 6	
onehost-13 to 32	Hypervisor	Scientific Linux 6	About 600 cores

Figure 2-4 StratusLab Infrastructure at CNRS/LAL

2.2.3. IRT Testbed

Interoute testbed is derived from the production infrastructure though it is run isolated from production services in order to implement CYCLONE experiments without impact on existing services and customers.

The main location of the testbed is the Interoute Milan PoP in which a dedicated co-location area is made available to host dedicated servers for the CYCLONE software.

For the purposes of CYCLONE experimentation an initial setup of 2 servers hosted in the dedicated CYCLONE co-location area is planned.

Each server will have target specifications as follows:

- (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

It is assumed that the proposed specifications will be suited to cover the needs of CYCLONE software experimentations. Moreover, a number of test virtual machines could be run to emulate the distributed CYCLONE cloud IaaS. The proposed hardware configuration can offer up to 40 physical CPUs and 512GB RAM which thanks to hyper-threading technology of the processors (up to 4x vCPU per physical CPU) can raise the number of vCPUs made available for CYCLONE tests up to 160 vCPU. In an example scenario in which each VM will use 1 vCPU and 2GB vRAM, this testbed can host up to 160 VMs.

The servers can be equipped with KVM or QEMU hypervisor software (depending on the final choice of the consortium) and can run any free-license software in the VMs (OS, IaaS control system, etc.) made available by CYCLONE (i.e. StratusLab, Slipstream, Tresor, etc.). These software tools are different from the IaaS Cloud Management System platforms Interoute uses in its commercial infrastructure and are examined in this CYCLONE testbed to evaluate potentials and new features.

The CYCLONE servers will access the Internet through a dedicated connection. An initial 2Mbps uplink to the Interoute backbone is set up, which will be mostly used for control plane traffic among the multiple distributed software instances of the CYCLONE testbed. Should further connectivity requirements emerge from CYCLONE software and test plan, those will be analysed for a potential upgrade of the Internet connectivity.

Both CYCLONE servers and co-location area are under management of the central Interoute NOC, who monitors computing and network parts through the standard network management systems used by Interoute for commercial services.

The CYCLONE co-location space at Interoute Milan PoP is also available to host additional hardware (e.g. servers or network nodes) possibly provided by other partners in the consortium, in case of specific tests on configurations and hardware different from the provided one.

In addition to the dedicated servers and colocation described above, it could be possible to activate for the experimentation purposes of the project an Interoute VDC JumpStartUp Pack, which could allow to allocate up to 2 Virtual Machines per account with the following cumulative specifications:

  - 2 vCPUs
  - 2GB RAM
  - 60 GB of storage

These VMs can be deployed in two different VDC locations, namely Paris and Berlin DCs, and have access to public Internet via Interoute Network.

The activation of the VDC JumpStartUp Pack depends on the need in CYCLONE to allocate additional VMs out of the ones available through the dedicated hosting platform, e.g. for testing scenarios aiming at the integration of private and public Clouds. For the time being, the activation of the VDC JumpStartUp Pack is not foreseen for the initial use case testing in CYCLONE.

3. Network connectivity and requirements

3.1. Inter-Cloud network connectivity Testbed alternatives

Inter-Cloud network connectivity is part of the cloud federation environment as it connects the different distributed sites over which applications are deployed. Traditionally there have been two options when it comes to a multi DC connectivity service: the utilization of an IP service and accept whatever performance is achievable across the multiple network domains or construct a private network with dedicated capacity designed to meet specific performance needs.

However, as it has been pointed in D5.1 [1] inter-domain connectivity for federated Cloud services may need to rely on dedicated inter-DC connectivity services that can either be based on connection-oriented paradigms or QoS in controlled and deterministic way. It is therefore highly desirable to also consider, when possible, the inter-DC segment of the solution as part of the networking resources to be controlled, avoiding the plane IP service (usually "*Best effort*" services), since nothing could be controlled in case this was the inter-cloud connectivity solution.

In this section we propose a number of potential alternatives that could be adopted by the CYCLONE testbed and that would enable a certain level control and management, so that the OpenNaaS network management platform can provide this inter-DC network segment "*as a service"* while describing applications' requirements in terms of QoS thresholds (i.e. Bandwidth required, latency, etc.).

3.1.1. GÉANT connectivity services

The GÉANT community has been largely devoted to promote network services across Europe, facing the challenges that NRENs (National Research and Education Network) providers have experienced while building a global European inter-connected network mainly focused on research projects. There are a number of well-known GÉANT multi-domain connectivity services that could be considered as an alternative to connect CYCLONE sites. These services conform to the cloud federation.

It must be pointed that at this stage of the project it is not clear whether CYCLONE partners will be able to access the PoPs (Points of Presence) of any of these connectivity services, since not all partners may be able to connect to GÉANT. This will be investigated during the following months, of the project.

A point in favour of using GÉANT services lies in the collaboration between the GÉANT community and I2CAT for the automated provisioning of some of these services using OpenNaaS. Two of the reference multi-domain connectivity services developed in the context of GÉANT community are presented below.

3.1.1.1. *GEANT Bandwidth on Demand (BoD) connectivity service – AutoBAHN service*

GÉANT BoD allows users to provision multi-domain circuits through a web interface in minutes rather than weeks. BoD constitutes a flexible network option for situations where users need to reliably transfer large amounts of data between two end points, for a short period of time, and with guaranteed bandwidth. GÉANT BoD is ideal for international research collaborations in fields such as radio astronomy, high-energy particle physics, bioinformatics, life sciences and arts, e.g., collaborative audio-visual performances [2]. All in all it constitutes a suitable inter-DC connectivity service to satisfy CYCLONE cloud federation services. Most relevant BoD key features include:

- **Multi-domain**: Bandwidth-on-demand services so far have only been available within a single domain and, although ad-hoc multi-domain circuits have been created before, by the time BoD was proposed, it was the world's first multi-domain bandwidth-on-demand service.
- **Dynamic provisioning**: Traditionally, creating multi-domain point-to-point circuits has been a time and resource-intensive process. The GÉANT BoD service's advanced provisioning tools now enable circuit paths to be found and created dynamically with just a few clicks of the mouse.
- **Flexible connections**: GÉANT BoD offers all of the benefits of a dedicated point-to-point circuit – guaranteed capacity, no congestion, and deterministic performance – but with the flexibility of an IP service.
- **Tool-independent**: Being based on the IDCP, and soon the NSI protocol, GÉANT BoD offers the option of choosing a provisioning tool that best suits the network. GÉANT offers its AutoBAHN provisioning tool but equally other in-house tools can be used.
- **Co-provisioned:** The BoD service is built on a trust model that is unique to the NREN community. Each NREN will offer and deliver part of a jointly provided service to users across Europe [2].

These features make from GÉANT BoD a good option that may satisfy CYCLONE's requirements (Dynamic, multi-domain, flexible) while provisioning inter-DC connectivity services. Moreover, in the past collaboration between i2CAT and the GÉANT community has been implemented to automate the provisioning of network resources between DC sites by means of the OpenNaaS network management platform. Figure 3-1 shows the overview of the testbed set up: Basically, OpenNaaS models the BoD service in order to control the provisioning features from the PoPs to which the remote sites were connected. Although being a prototype implementation, it could be used as a good starting point if this option was chosen.

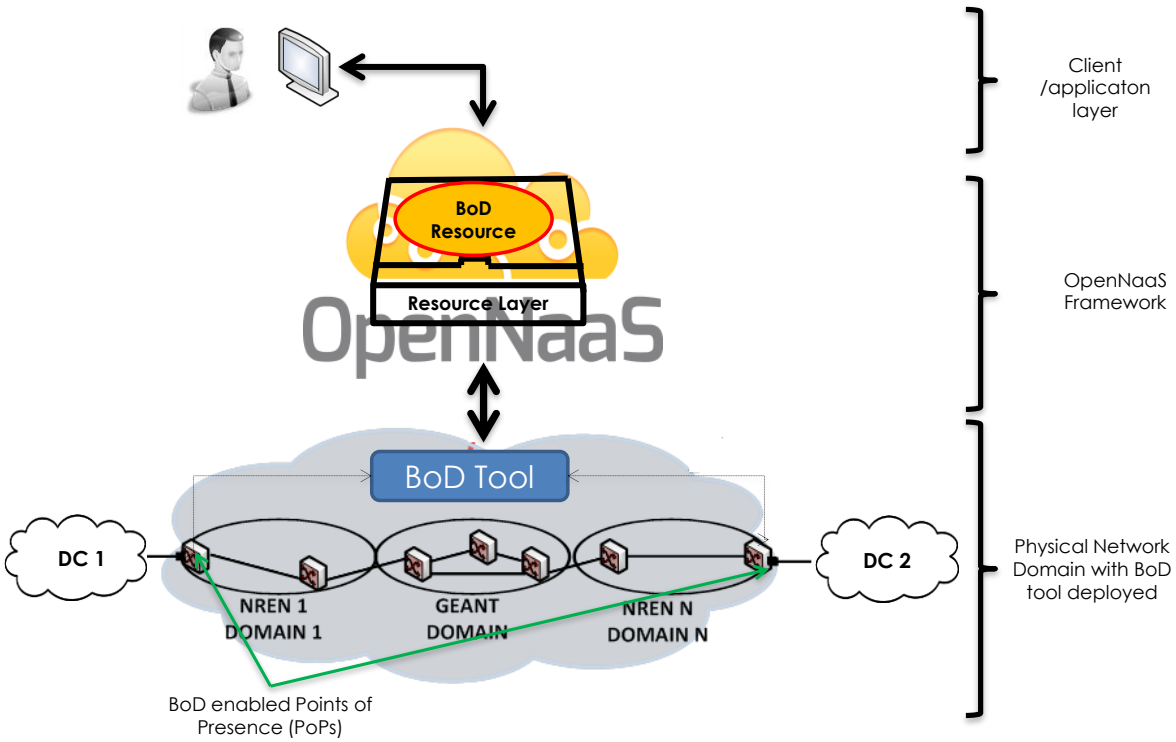

Figure 3-1 OpenNaaS use case experience for automating BoD connectivity service

3.1.1.2. *GEANT Multi-Domain Virtual Private Network (MDVPN) connectivity service*

Multi-Domain VPN is a joint service delivered by the NRENs and GÉANT backbone that provides with VPN transport connectivity at different levels. It is based on MPLS (BGP/MPLS IP VPNs – RFC 4364) [3] and BGP-LU (BGP Labelled Unicast – RFC 3107) [4] and it is already available in many routers of different national providers across Europe to which some of CYCLONE partners could be connected to make use of it. For instance, MDVPN routers are available at RENATER (French NREN) and DFN (German NREN). Figure 3-2 shows the topology footprint of the MDPVN connectivity service. It can be seen that it adopts a hierarchical infrastructure: the GÉANT backbone acts as "carrier of carriers" and NRENs act as peers. A more detailed description on the service approach and principles as well as already carried out tests and use cases can be found in [5].

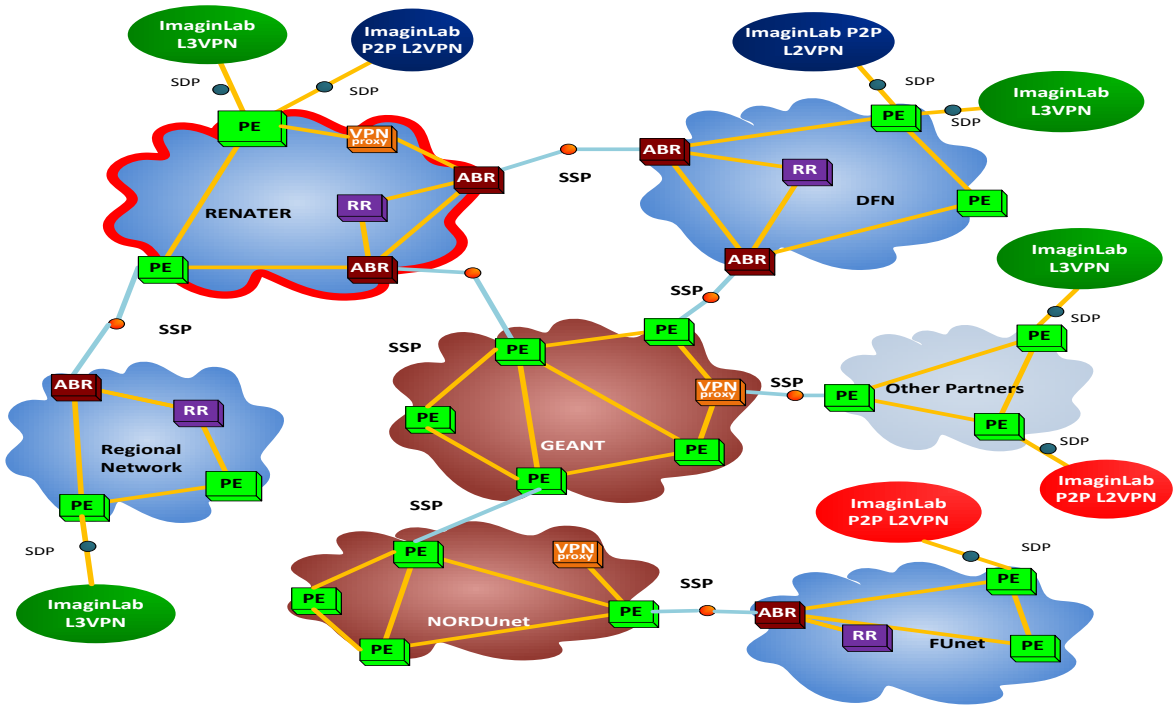

Figure 3-2 MDVPN footprint deployment

The most relevant MD-VPN connectivity characteristics are:

- **Multi-domain**: The MDVPN service is utilized across Europe and already comprises a large number of Points of Presence.
- **OPEX saving**: Thanks to its VPN multiplex feature and avoiding manual configuration between the PoPs.
- **No CAPEX**: No CAPEX is required as it relies on the reuse of standard features already available in PoP routers.
- **Large number of services**: MD-VPN providers a bundle of connectivity services:
  - L3VPN
  - P2P-L2VPN (L2 P2P circuit)
  - MP-L2VPN (VPLS)
- **Not only MPLS based**: The MD-VPN service is ready to cooperate with non-MPLS domains and regional/metro networks.
- **Flexibility**: The service offers a new way to propose a bundle of useful services that covers a wide scope of their user needs:
  - Reduce lead time to provision services assists ad hoc and short period projects.
  - All types of site can be connected using multiple access and solutions [6]

It is important to note that this multi-domain connectivity service has gained popularity within the NREN community during the last year and also constitutes an attractive option to implement inter-DC connectivity among CYCLONE sites.

3.1.2. Other studied alternatives

The availability of previous GEANT-NREN services comprise a number of L2 and L3 connectivity options made available by means of the points of presence enabled by the NRENs at the edges of their respective domains. Nevertheless, in case it is not possible to some of the CYCLONE partners that provision IT facilities to connect to those points of presence, other L2 and L3 VPN options could be configured among the different domains to build an E2E cloud federated network.

It is not clear at this stage of the project which option will be eventually implemented and the ongoing task 7.2 (Testbed specification/definition) will further analyse the possibilities and select the most suitable one considering the deployment and networking requirements of the two pre-selected use cases.

3.2. Networking requirements

The following technology areas are required features of the Inter-cloud network: address management, routing management, security, network infrastructure management, cloud VPN management, and system collaboration with reference to the general technology areas of a distributed system. In what follows, basic networking requirements have been identified for the deployment of CYCLONE's testbed infrastructure.

Address management

- Network shall be able to avoid private address collision between cloud systems.
- Network shall be able to reroute traffic without IP address collision when using the same private IP addresses among multiple cloud systems, and recovering cloud system shall be able to properly handle the traffic.

Routing management

- Routing management shall be able to distribute accesses to servers.
- Routing management shall be able to relocate the user access to the distributed cloud system according to workload.
- Network shall be able to configure a communication with minimum latency.
- Network shall be able to dynamically reroute a communication accessing to the cloud system.
- Network shall support flexible network routing control to distribute user access for load balancing.
- Network shall be able to monitor workload autonomously, and control route and bandwidth according to the result.
- Network shall be able to reroute the communication which transits multiple network operators.

Security

- Single sign on.
- Cloud system shall be able to execute the access control to the cloud VPN with its own control policy independently.

Network Infrastructure management (mainly performance management)

- Network shall be able to identify service quality degradation for user access through autonomous performance measuring that is caused by increased load and congestion.
- Network shall be able to measure latency between user and cloud system.
- Network shall be able to provide end-to-end network performance monitoring and control network to guarantee network quality during recovery period.

VPN management

- Network shall be able to create, delete, and modify cloud VPN between cloud systems.
- Network shall be able to isolate cloud VPN between specific combination of cloud systems, from cloud VPN used by other cloud systems.
- Cloud system shall be able to monitor network quality for the cloud VPN.
- Cloud system shall be able to monitor cloud VPN autonomously to adjust bandwidth according to the usage situation.
- Network shall be able to temporarily increase bandwidth at cloud VPN beyond its contract.
- Provider shall be able to set up multiple networks for the monitoring purpose.
- Network shall be able to create, delete, and modify a cloud VPN dynamically through the control interface.
- Network shall be able to classify incoming traffic to transfer to a corresponding cloud VPN.
- Network shall be able to dynamically create a cloud VPN which may transit multiple network operators.
- Network shall be able to temporarily increase bandwidth of cloud VPN which transits multiple network operators.
- The common service interface shall be defined to specify various cloud VPN configuration technologies.

Cloud system to cloud system

- Network shall be able to securely transfer data for cloud collaboration.
- Cloud-access network shall be able to securely transfer data between user and cloud system.
- Cloud provider(s) shall be able to search other cloud provider(s) who can lease sufficient resources (such as Web servers) to guarantee performance.
- Cloud system shall be able to search and discover a cloud system which meets with user requirements (performance, etc.).
- Cloud system shall be able to autonomously monitor other cloud systems to detect the necessity of recovery.
- Cloud system shall be able to autonomously transfer resource configuration information and service data toward cloud system of other provider for backup the services.
- Cloud system shall be able to autonomously search resources in other clouds for leasing.
- Cloud system shall be able to identify quality degradation autonomously by measuring the performance of the providing service(s).
- Cloud system shall be able to search, discover, and secure a cloud system(s) which can lease resources to create service environment.

4. Software and Tools

The project aims at integrating existing cloud management software to allow a unified management of federated clouds. For achieving this target the CYCLONE project integrates and improves existing, production-quality tools. The provided platform with the open source tools facilitates the deployment and management of complex cloud-based applications. Thus, the testbed will include apart from hardware elements, software and networking elements that will allow the testing of all the components and rely on frequent exchanges about requirements, technical capabilities, and general solution ideas between the involved partners the CYCLONE software consists of StratusLab, Slipstream, Tresor and OpenNaaS. The first feedback and the requirements from the selected use case are already integrated in the software selection.

In fact, on the base of the conceptual testbed architecture introduced on the previous section, it is important to identify as components of the testbed both the hardware and software resources required to offer an appropriate run-time environment to application service providers that develop, deploy, and maintain complex computing platforms within multiple cloud infrastructures.

CYCLONE's software components need to be tested and improved to meet the identified requirements such as providing enhanced functionality for cloud providers that agree to federate their resources or enabling dynamic allocation of bandwidth between cloud providers and common authentication mechanisms. From the integration of the proposed tools the application developers should be able to take advantage of features like VM coordination within deployments, automated placement of service components, and scaling of service components, essentially providing them with the means to develop a Platform-as-a-Service (PaaS) offering. Likewise, the CYCLONE software has as objective to allow developers to ensure the end-to-end, secure use of data within their application as well as secured access to remote data sources. In this way the confidence of application developers in using cloud infrastructures for services that handle sensitive personal data will be gained and the requirements of concrete academic and commercial use cases will be met.

More specific features and deployment requirements of the components are provided in other deliverables in WPs 4, 5 and 6. Moreover, the possible combinations of them and the integration in the testbed will be specified in the next steps of the project and will be described in future documents.

4.1. StratusLab

StratusLab [7] is a software distribution that allows to deploy complete, IaaS clouds in their own data centre. The distribution aims to be as simple to install, maintain, and use as possible. The software is sufficiently extensible to allow other CYCLONE developments, like networking and identity management, to be added with reasonable effort.

Default deployment assumes two types of machines: (a) Front-End (cloud management server) and (b) Node (host in which the virtual machines will be instantiated. By default the compute and disks management services will be deployed on the Front-End. A set of packages will be installed on the Node(s) and then, the Node(s) will be configured and added to the manager of the compute resources on Front-End. By default KVM is used the Node(s).

4.2. SlipStream

SlipStream [8] is a multi-cloud application management platform for the full lifecycle of the cloud applications, including the definition of the application, its automated deployment, scaling its resources, and its eventual termination. The model covers well the basic requirements needed for the initial CYCLONE prototypes and can be used immediately.

The Slipstream component will be deployed as part of the CYCLONE testbed. Depending on the needs of the collaboration and the details of each of the available testbed resources, the deployment can be carried out as a dedicated SlipStream instance deployed on testbed resources, a dedicated SlipStream instance using SixSq resources, or integrated as part of SixSq standard SaaS offering (https://nuv.la).

4.3. Tresor

Tresor [9] enables the secure management of cloud service consumption, and cloud service mediation and brokering. The provided components include the Identity Federation Provider for Web Single Sign On, the Distributed Logging System for multi-cloud logging, and the Service Compendium for allowing CYCLONE cloud application developers to assess different deployment options in combination with SlipStream. This software provides end-to-end security for cloud applications using the Trusted Cloud Transfer Protocol [9].

TRESOR includes an Identity Federation Provider for Web Single Sign On, a Distributed Logging System for multi-cloud logging, and a Service Compendium for allowing CYCLONE cloud application developers to assess different deployment options in combination with SlipStream.

All components can be deployed via Docker as well as Slipstream. TUB will provide enough instances of all components for the development phase. These will be hosted on TUB resources (VMWare VSphere & Microsoft Hyper-V). Integration testing has already started by deploying TUB components through SlipStream on the StratusLab Cloud hosted at LAL.

Based on the usage scenarios we currently do not foresee a high system load in production which would require additional resources not currently available in the initial testbed. All components communicate via the Internet and there is no need for special connectivity besides unfiltered Internet access.

Every component could be run on a modest Ubuntu VM with 1 vCPU, 1024 MB RAM and 5 GB HDD. The Logging System could get somewhat bigger in terms of HD if it would receive a heavy load of logs (which we currently don’t assume), but most certainly not more than 100 GB HDD.

4.4. OpenNaaS

OpenNaaS [10] is an open-source platform for provisioning network resources. It provides tools for managing the different resources present in any network infrastructure. It allows the deployment and automated configuration of dynamic network infrastructures. This software allows them to contribute and benefit from a common Network as Service (NaaS) software oriented platform for both applications and services.

OpenNaaS works in a centralized way, requiring a single machine for its deployment and offering a single point of communication through its API. The physical servers foreseen in the initial testbed specification are sufficient for supporting the correct performance of this component

5. Use case View

The CYCLONE project has identified two initial leading use cases: an academic cloud platform and associated services for bioinformatics research and a commercial deployment for future energy management in the energy sector.

One of the main objectives of the testbed is to show that the implemented tools in CYCLONE target the developer requirements for each selected use case. For this purpose the testbed architecture will support a high degree of flexibility in the architecture. From the discussions with the partners and the description of the use cases it is clear that diverse systems and hardware is necessary to be installed in order to reach this flexibility. The selected use cases aim to retrieve general elastic cloud federation requirements and specific features to enhance current cloud infrastructure processes and services provisioning mechanisms.

The requirements derived from the use cases should be identified in the testbed. On the side of the testbed components, they should enable the deployment of the use cases. This resulted in the definition of a common set of requirements. These common requirements should be considered in the testbed toward the assistance of the design of federated cloud with security features and conceal the allocation and management of cloud resources. For example, the main requirement to the cloud infrastructure for running the Energy use case is to run UNIX-like systems on non-dedicated infrastructure.

The setup for the testbed was done in close connection with WP3 “Use Cases” since it involves activities that concern the requirements and analysis, deployment and support, testing and validation phases.

6. Conclusions

This document specifies the testbed that will be used for the purposes of the CYCLONE project. The initial architectural design for the testbed is achieved. The current testbed is composed of three sites provided by the following partners:

- QSC AG
- CNRS/ LAL
- IRT

The required software tools for collaboration, development and integration of the CYCLONE solution into the testbed are determined taking into consideration the requirements from the initially selected use cases from the field of bioinformatics and energy. The combination and the increase of the cloud should be realized depending on the needs that will arise. During the lifetime of the project new use cases will be incorporated and deployed, bringing new requirements that will extend the testbed infrastructure.

References

[1] CYCLONE Deliverable D5.1: Functional specification of the E2E Network service model. Available at: http://www.CYCLONE-project.eu/deliverables.html

[2] http://services.geant.net/bod/Resources/Documents/BoD-product-brief-18.5.12.pdf

[3] https://tools.ietf.org/html/rfc4364

[4] https://tools.ietf.org/html/rfc3107

[5] https://www.terena.org/activities/netarch/ws2/slides/141113-psnc-MDVPN.pdf

[6] http://issuu.com/danteprm/docs/md-vpn_white_paper

[7] Charles Loomis, Mohammed Airaj, Marc-Elian Bégin, Evangelos Floros, Stuart Kenny, and David O'Callaghan. "StratusLab Cloud Distribution" in European Research Activities in Cloud Computing, Dana Petcu and Jose Luis Vasquez Poletti, Eds. Newcastle upon Tyne: Cambridge Scholars Publishing, 2012, pp. 260-282. Available from http://hal.archives-ouvertes.fr/hal- 00676252.

[8] SlipStream is a product from SixSq, the core of which is released under the Apache2 open source license. Information available from: http://sixsq.com/products/slipstream.html.

[9] TRESOR. "Trusted Ecosystem for Standardized and Open cloud-based Resources". Available: http://www.cloud-tresor.com

[10] Slawik, M. (2014): The Trusted Cloud Transfer Protocol. In: Institute of Electrical and Electronics Engineers (IEEE) (pub): 2013 IEEE 5th International Conference on Cloud Computing Technology and Science (CloudCom).

Abbreviations

B2B	Business to Business
DC	Data Centre
E2E	End to End
HW	Hardware
IaaS	Infrastructure-as-a-Service
ICT	Information and Communication Technology
IT	Information Technology
NaaS	Network-as-a-Service
PaaS	Platform-as-a-Service
PoP	Point of Presence
SaaS	Software-as-a-Service
SCI	Smart Core Interworks
SDN	Software Defined Networks
SP	Service Provider
SW	Software
TCP/IP	Transmission Control Protocol/Internet Protocol
TCTP	Trusted Cloud Transfer Protocol
VD	Virtual Disk
VDC	Virtual Data Centre
VM	Virtual Machine

<END OF DOCUMENT>