PROCEEDINGS OF SCIENCE

Prototyping a Cloud Ecosystem for a Regional Public Administration

Livio Fanò*
Università degli Studi di Perugia and INFN
E-mail: livio.fano@pg.infn.it

Gian Mario Bilei, Loriano Storchi, Andrea Valentini
INFN Perugia
Perugia, Italy

Enrico Fattibene, Matteo Manzali, Davide Salomoni, Valerio Venturi, Paolo Veronesi
INFN CNAF
Bologna, Italy

Hassen Riahi, Daniele Spiga
CERN
Geneve, Switzerland

Cinzia Amici, Serenella Carota, Francesco Cirillo, Maria Laura Maggiulli, Andrea Sergiacomi, Donatella Settimi
Regione Marche
Ancona, Italy

Claudia Diamantini, Domenico Potena, Giuseppa Ribighini, Emanuele Storti
Marche Polytechnic University
Ancona, Italy

Damiano Falcioni, Daniele Faní, Barbara Re
School of Science and Technology
University of Camerino
Camerino, Italy

In this paper we present the lessons learned in the deployment of a Cloud solution in the Marche Region Local Public Administration, which represents one of the pilot experiences at National level. The *MarcheCloud* (MCloud) pilot Project, started in mid-2012 as a joint collaboration among the Marche Region, National Institute of Nuclear Physics (INFN), University of Camerino and Polytechnic University of Marche.

International Symposium on Grids and Clouds (ISGC) 2014,
23-28 March 2014
Academia Sinica, Taipei, Taiwan

*Speaker.

PoS(ISGC2014)009

1. INTRODUCTION

MCloud is an hybrid Public/Private Cloud architecture based on the *anything-as-a-service* (XaaS) paradigm. The aim is to provide innovative digital services to enterprises and citizens through scale-economies, in order to assure a new business model starting from the Marche region, namely: the development of public, regional services for the regional e-government (MCloud.Gov) and, on the other hand, business support to foster a new ICT development model and innovative services based on public data (Business and Research Cloud component MCloud.B&R).

The proposed approach is also the basement of a new regional business model. Multichannel services (i.e. mobile or DTV) can be quickly realized with a low investment (as a consequence of the re-use and interoperability paradigm) and are supposed to be implemented in a pay-per-use scenario (i.e. self-provisioning, freemium, etc.).

The focus of the pilot project has been on the development of the infrastructural core of the MCloud.Gov component. The pilot has completed deploying a welfare service to the citizens related to the access of clinical analyses.

The deployment of the solution required the development of a basic infrastructure guaranteeing maximum adherence to national and international standards so to ensure interoperability and compliance with the rules regarding privacy and data security [1].

In addition, the choice of an open source software is consistent with the recommendations issued to Public Administrations described in art. 68 of the Italian Codice dell'Amministrazione Digitale (D.Lgs. 85/2005, in Italian) [2].

Among all the possible infrastructural solutions, the OpenStack framework [3] was chosen to implement the IaaS infrastructure, the core of the project. OpenStack, in particular, is an open source product that can be deployed on open source platforms (Linux); it has strong backing from the industry, with major ICT players directly supporting it [4]; it enjoys a steady growth in terms of both functionalities and developers [5]; it has an open and extensible architecture, mainly written in Python; it interoperates with other Cloud stacks and APIs (of remarkable importance for the MCloud use cases was the ability to interoperate with VMware vSphere clusters [6]); last but not least, there is significant experience with OpenStack deployment, configuration and extensions within the Italian National Institute for Nuclear Physics (INFN), which was the institution responsible for the MCloud IaaS infrastructure architecture and deployment.

Privacy and security are guaranteed by the integration into the infrastructure of the existing regional authentication system (FedCohesion) allowing a unified access to the services. This solution is particularly innovative with respect to the management of privacy and data access in the national and international scene, essential to the governance and security.

2. The MCloud Platform

2.1 The OpenStack-based Infrastructure

When the project started, the most stable OpenStack release was Folsom [8], so the initial IaaS infrastructure has been based on that release. Given the goals of the Marche Cloud project, it was focused on a subset of the OpenStack components, namely:

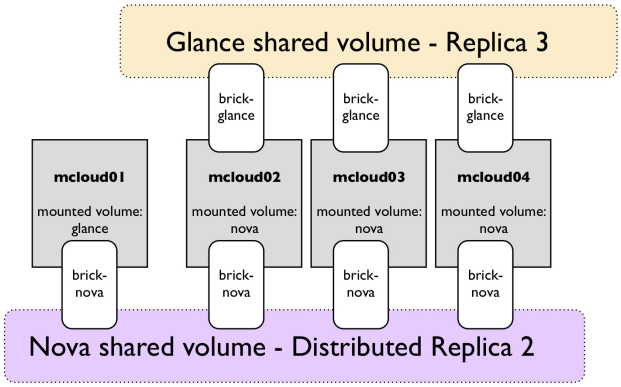

Figure 1: Storage areas within the MCloud project.

- Dashboard: OpenStack provides a default user interface through a module called Horizon. This module was extended to provide the monitoring and accounting systems, as described below.
- Compute: the standard OpenStack module, named Nova, providing compute capabilities (e.g. the ability to manage the virtual machines, or VMs) was used.
- Storage: OpenStack provides block storage functionalities through its Cinder module, and this was extensively used in MCloud. OpenStack also provides an object storage module called Swift to manage files seen as single objects; since there was no immediate need for such a functionality, it was not deployed in the MCloud infrastructure.
- Network: the OpenStack network module, at the time called Quantum (now renamed Neutron), was deployed and replicated across all the OpenStack compute nodes through the Nova-network component.
- Image repository: the standard OpenStack Glance component was used.
- Authentication: the OpenStack Keystone module was used, extended to support integration with SAML-based authentication mechanisms already in use by the Marche Region, as described below.

The shared file system chosen for the MCloud infrastructure is GlusterFS [7] by Red Hat, because of the strong support expected from the huge community of developers as well as its native integration in OpenStack. It was configured to provide a high-available repository for virtual images and a common storage area across all compute nodes. The GlusterFS deployment was also configured to implement automatic fail-over in case of problems to one of the GlusterFS servers. While initially the disks local to the compute nodes were used to define the IaaS storage volumes, in a second phase GlusterFS was re-configured to point to SAN-based storage, available in the Regione Marche computing center.

Figure 1 shows how GlusterFS was configured so that the Glance volume (image repository) was replicated three times for redundancy, and how the Nova volume (storage space shared across

compute nodes and used to store running VMs) was set up in replicated-distributed mode, so that live migration could be possible with both high availability and good I/O performance.

Along the course of the Marche Cloud project, OpenStack released two new versions, code-named Grizzly and Havana. The first evolution of the MCloud IaaS infrastructure, then, involved the migration from OpenStack Folsom to OpenStack Grizzly. The architecture was reworked to add higher flexibility especially at the network layer, where a mix of per-tenant, private networks, together with external networks (used to masquerade private VMs and make them accessible to the outside world) and shared networks (used to connect the OpenStack-based MCloud deployment with existing legacy systems like VMware vSphere and Proxmox clusters) was defined. In addition, the MySQL database used by OpenStack was made redundant creating a MySQL cluster; finally, the GlusterFS cluster was connected to volumes derived directly from the existing Storage Area Network of the Regione Marche computing center.

Further evolutions of the MCloud IaaS infrastructure will exploit OpenStack features available in the Havana release, starting with the definition of a Load Balancing as a Service to scale websites and applications, and with the implementation of a fully redundant set-up for all the components used by OpenStack (Keystone, MySQL, AMQP, Dashboard, Glance, Cloud controller) based on Pacemaker and Corosync [10].

2.2 Performance test

Several tests have been performed on an equivalent infrastructure configuration realized at the CNAF-INFN facility. Instances are running on the same hypervisor (24 cores), the network is switched at 1 Gbps and the storage system is GlusterFS in replica. The IPERF tool is used for CPU idle and networking measurement while IOSTAT is used to monitor the networking activity.

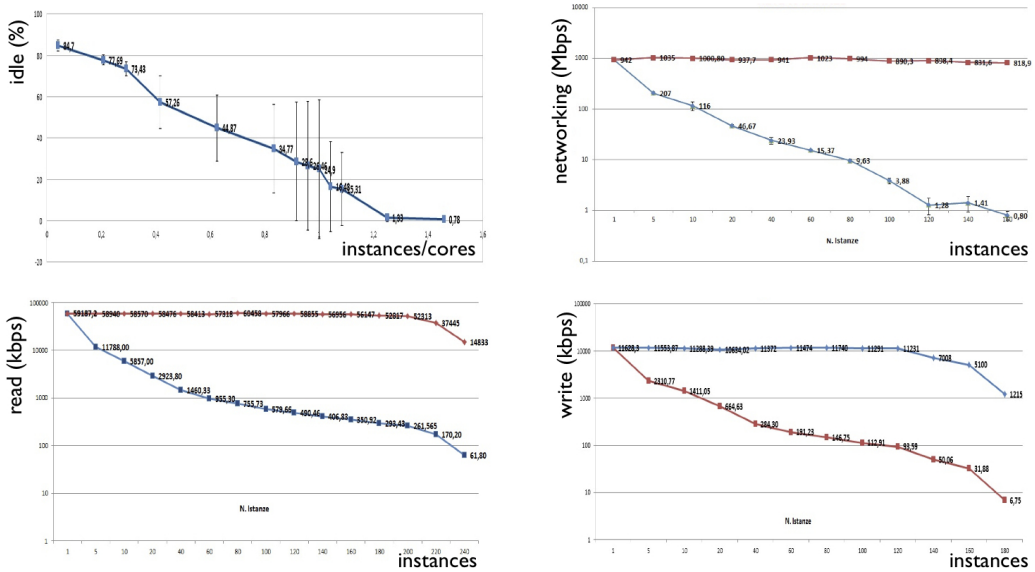

Figure 2: Stress test results. a) top-left: cpu idel VS instancescores ratio b) top-right: networking activity VS instances c) bottom-left: storage read activity VS instances d) bottom-right: storage write activity VS instances

Results are summarized in Figure 2. The CPU idle index has been used to investigate the scalability related to fraction of running instances per CPU-core. Results are summarized in Figure 2a where a linear behavior observed increasing the instances / CPU-cores ratio; as expected the idle is $\sim$0 when the core is shared among several instances (cooperation mode). Figure 2b summarizes networking activity: the full bandwidth is correctly saturated up to 160 instances without any loss in the overall performance. In Figure 2c and d the storage performances are summarized in terms of read and write metrics (*c* and *d* respectively). The speed limit is driven by the disks capabilities, the network bandwidth is not yet saturated, and the performances of the infrastructure are stable up to $\sim$200 instances (read) or 160 (write). As a general conclusion, the performed tests tend to indicate a very robust infrastructure which scale as expected increasing the number of instances up to more than 160 instances per CPU (24-core).

2.3 Cloud Identity as a Service

Generally speaking, the management of federated identity in the cloud allows reduction of capital and effort expenses that are required to maintain the identity management system, and reduces the effort required to port applications to the cloud. The Marche Cloud project aims to satisfy identity management requirements using a cloud-computing scenario. We aim to design and develop a Cloud Identity as a Service (IDaaS). The identity issues were explored according the different levels of cloud from service to infrastructural level. To do that we evaluate the Open Stack infrastructure in term of identity management and we study the goodness of the regional authentication framework, named FedCohesion, toward the cloud.

During the years, the Marche Region has made substantial investment on the Cohesion framework, so the decision to re-engineer prevailed over substitution with others framework SAML 2.0 compatible like Shibboleth [11, 12]. For the Marche Cloud project FedCohesion results as an enabling infrastructure, realized that its integration in term of IDaaS makes possible the porting of most of Regional services. The first step was the integration of the regional authentication framework at infrastructural level (IaaS) so that different cloud federations can be trusted and integrated in term of identity.

Extending the Keystone component to use federated authentication, beyond the advantages of externalizing the authentication process, allows the addition of multiple Identity Providers without additional modification of the underlying platform. This makes possible the creation of a community of reference in term of identity. The identity provider we introduced is FedCohesion. The main advantages are the increasing of security (stronger then weak name and password), the federation identity model that among the others avoids asking users to create additional account.

3. Monitoring and accounting: Evaluation of needs and data-type identification

Two categories of data, and the corresponding producers, have been identified into the Openstack architecture:

- infrastructural: derived from Nova component and populated by the Openstack message service
- from the virtualization system: information that is accessible by API, plugins and libvirt

Moreover, the main components of the system have been identified as:

- *Data Producer*. The natural source of information can be classified as the following:
  - Facilities and Services monitoring;
  - Activities monitoring.
- *Data Flow*. The flow can be divided, for monitoring or accounting purposes, as follow:
  - Data Transport;
  - Data Storage;
  - Monitoring and View.
- *Data Format*. The design of the format has to assure the maximum flexibility in integrating new applications into the framework.
- *Architecture*. A new module related to monitoring and accounting activities has been integrated into the Dashboard, it can be summarized as composed by 3 different layers (Figure 3):

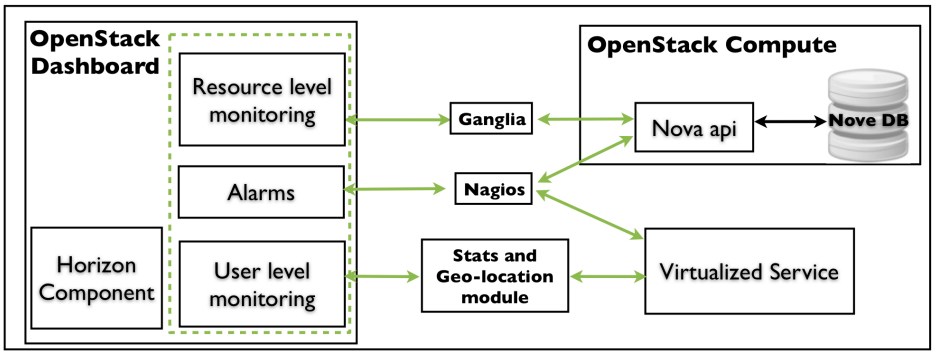

Figure 3: Monitoring and Accounting components architecture.

  - Resource level monitoring;
  - Alarms;
  - User level monitoring.

The most common applications that already integrates the three activities have been deeply investigated: Ganglia, Collected and Zenoss as performance providers while Nagios and Zenoss as notification system. Furthermore, a python based solution that interfaces directly with the internal messaging system (AMQP) and with the virtual engine (KVM) via libvirt has been analyzed. Figure 4 summarizes the comparative results between the different applications based on the supported functionalities.

	Performance monitoring	User-friendly Web App	Notifications	Log monitoring	Libvirt plugin	Support of Windows	Plugin for OpenStack	Plugin based metrics	Richness in existing metrics	Popularity
Collectd	X		X	X	X	X		X		
Ganglia	X	X				X	X	X	X	X
Nagios		X	X	X	X	X	X	X		X
Zenoss	X	X	X	X	X	X	X	X		
Own libvirt-based script					X			X		

Figure 4: Monitoring tools, screening results.

The final choice is Ganglia as performance monitor and aggregator; Nagios has been adopted as notification system; different specific tools and plugins for the application layer. The Ganglia/Nagios choice, even if it requires a customization of the virtual image, is motivated by the following:

- They are largely used in combination with Openstack;
- A reduced integration effort due to the possibility to use specific plugins already available, such as the one at ref. [9], which allows to monitor the Nova services with common libraries;
- Easily expandable.

On the contrary, Zenoss, apparently a complete solution, is less diffused in combination with Openstack because of higher resource consumption.

3.1 Integration within the Dashboard

Several ad-hoc plugins have been developed in order to monitor the infrastructure and the application selected for the MCloud pilot project (access to electronic medical records). The server components of Ganglia and Nagios have been integrated into the Dashboard creating three custom menus related to:

- *Resource Level Monitor* and *Alarms*, to directly access the Ganglia and Nagios web front-end;
- *User Level Monitor*, which aggregate in specific views the resource consumption related to a specific user of the Openstack infrastructure.

Several Ganglia and Nagios plugins have been integrated into the virtual images used as reference to instantiate the Virtual Machines. One specific Ganglia plugin has been configured in order to monitor a specific virtual machine, collecting and reporting all the information from the internal MySQL database of OpenStack. In addition, two Nagios plugins have been created:

- apache_usage, it is able to monitor the number of requests per second reaching an Apache server;
- check_log, it is used to have an incremental summary by IP of the apache_usage output.

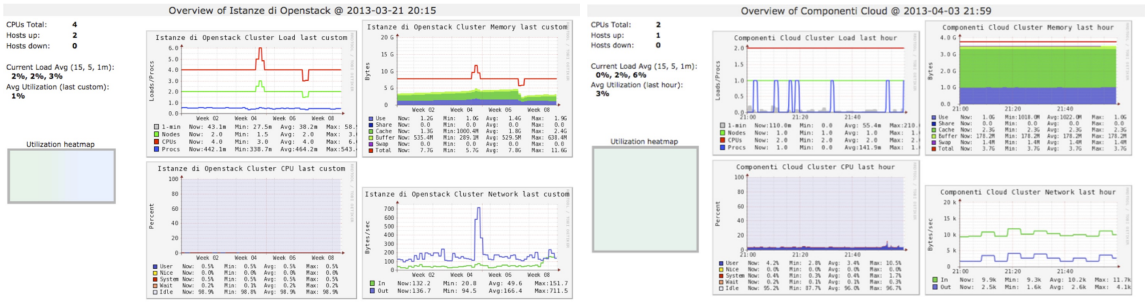

Figure 5: Virtual and Real resources consumption.

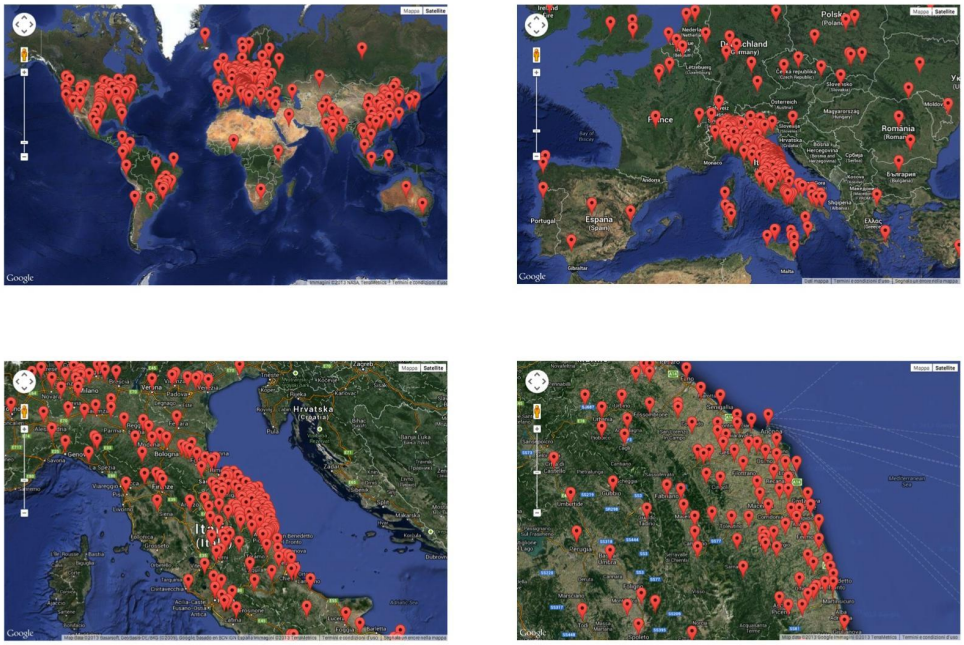

Figure 6: Web application users access, geographical summary

At this point the *monitoring and accounting* module is fully integrated and can be used to access all the functional details of the infrastructure. In Figure 5 is shown an example of the output, respectively for virtual and real resources.

The infrastructure is very stable, with very low consumption related to the considered load metrics (CPU load, storage, memory usage and networking). Nagios is also used to directly view in a googlemap the geographic information related to the Apache incoming contacts (Figure 6), thanks to the following plugins:

- check_webserver_log. It inherits from the check_log plugin and produce a notification when a connection to the Web application is active;
- check_nagios_log. It collects dates and IPs from check_webserver_log. Consequently a time range can be applied to filter the output.

4. Clinical Report as a Service

The core of the pilot has focussed on deploying a welfare service to the citizens related to the access of clinical analyses. A service that allows people to view the results of their clinical tests, performed at laboratories distributed throughout the region, was then developed. Indeed, before starting the MCloud project, around 20 laboratories already put at disposal of citizen their own web-based applications for consulting reports. The isolation of these web applications leads to an environment where data and services are not interoperable, it does not allow the administrator (the Region) to have an unique insight into costs and benefits of the system. Usually citizens consult reports during daytime of working days; hence the number of accesses varies considerably over time and geographical location.

The web application is a monolithic tool and does not allow users to access in parallel to the various functionalities. In order to overcome these limitations, it has been chosen to transform the (functionalities of) web applications in various services, and logically centralize them into the MCloud environment. Furthermore in addition to the web interface, mobile and smartTV applications have been developed in order to allow people to interact with services everywhere and everytime.

From the MCloud point of view, we have implemented and deployed the following services:

- *Login*, which checks the credentials of the user and enables them to consult their own reports and also manage their account password.

- *ReportList*, which returns the clinical history of the user, i.e. the list of reports of all clinical tests regardless the laboratory in which she has done tests. This service returns for each report just a triple: the code of the reports, the laboratory and the date of the test.

- *ReportData*. This service takes as input the code of the report and provides as output the list of all exams composing the report; e.g. a blood exam is formed by various tests: creatinine, glucose, sodium, and so on.

All services are RESTful and the output is returned in JSON, which is a lightweight protocol allowing to limit the workload of the user device. With the same goal, both computation and data management activities are assigned to services. Furthermore, the granularity of services, as opposed to monolithic applications, leverages the performances of the overall system by using at best the scalability and elasticity of a cloud platform. All communication from and to the services guarantee privacy and security, by using personal access and communication over secure http.

Figure 7-left depicts an example of interface where data provided by ReportData service are shown. In particular, the example is related to a blood test, where details of a complete blood count are exploded; in red anomalous values. The SmartTV app provides the same functionalities of the smartphone one, but they have been redesigned in order to fit the wider screen and the limited interaction capabilities offered by the TV remote control; in order to overcome the latter issue a keyboard has been simulated to support data entry. An example of the smart TV interface is shown on the right hand side of Figure 7, where the same data are reported in a different fashion. The app

Figure 7: An example of interface of the smartphoneand smartTV app.

has been deployed for the 2012 version of the Samsung smartTV operating system. At present is available at the Marche Region web site[1], soon on the Samsung marketplace.

5. Conclusions

The pilot project MCloud has been the first deployment of a cloud solution based on the OpenStack framework in the Italian public administration system. In this paper the motivation and the architectural design to put in production a welfare service to the citizens related to the access of clinical analyses, has been presented.

The high stability demonstrated during the production operations (from January 2013), where the overall downtime period has been measured to be less than 0.02% serving a volume of several tens of access requests by day, represents a key point for an open, low cost and resilient solution for a transparent public administration, especially where important services (like the access to clinical reports) need to be guaranteed with high availability. The project represented one of the pilot experiences at National level and allowed the collaboration to acquire significant competence on the actual issues, feasibility and potential impact of Cloud as the backbone of all the regional ICT infrastructure, both for E-Government and Business services. The experience and positive feedback serves as a foundation towards further developments and extensions of the pilot project also outside the administrative boundaries of the Region. The expansion of the MCloud infrastructure will then focus beyond the pure IaaS layer, and address more PaaS- and SaaS-related solutions, possibly in cooperation with other Cloud-related Italian projects of interest to the Marche region and national level, such as the recently financed Open City Platform (OCP) project [13]. From the application point of view, future attention will be devoted to the development of novel advanced services for business and research to foster a new development model based on innovation and advanced technological solutions in the ICT field based on public open data.

[1] http://www.ecommunity.marche.it/AgendaDigitale/MCLoud

References

[1] D. Lathrop, L. Ruma (Eds.), “Open Government: Collaboration, Transparency, and Participation in Practice , O’Reilly Media Inc. 2010.

[2] Codice dell’Amministrazione Digitale, http://www.camera.it/parlam/leggi/deleghe/05082dl.htm

[3] OpenStack, http://www.openstack.org/

[4] OpenStack Foundation, http://www.openstack.org/foundation/companies/

[5] OpenStack vs OpenNebula vs Eucalyptus vs CloudStack, http://www.qyjohn.net/?p=3432

[6] VMware and OpenStack, http://docs.openstack.org/grizzly/openstack-compute/admin/content//vmware.html

[7] GlusterFS, http://www.gluster.org/

[8] Folsom Architecture, http://ken.pepple.info/openstack/2012/09/25/openstack-folsom-architecture/

[9] https://github.com/ganglia/gmond_python_modules/tree/master/openstack_ monitor.

[10] Pacemaker and Corosync, http://clusterlabs.org/

[11] B. Pfitzmann, and M. Waidner, “Federated identity-management protocols," Security Protocols Workshop, 2003, pp. 153-174.

[12] R. Morgan, S. Cantor, S. Carmody, and W. Hoehn, “Federated security: The shibboleth approach," EDUCAUSE Quarterly, vol. 27, no. 4, 2004, pp. 12-17.

[13] Open City Platform (in Italian), http://goo.gl/JWLMWv