SOCIETIC
SOCiety as Infrastructure for E-Science via technology, innovation and creativity

Deliverable no.	D3.1
Deliverable name	Operational Platform for Experimental Research
Dissemination level	PU
WP no.	3
WP name	Infrastructure Operation and Deployment
Date	29/05/2013
Date of delivery	31/05/2013
Actual date of delivery	31/05/2013
Status	Final
Author(s)	Cândida G. Silva, Francisco Sanz García, Francisco Brasileiro, Rui M. M. Brito
Reviewer (s)	Eduardo Lostal, Francisco Brasileiro, Fermín Serrano

SOCIETIC is supported by the European Commission under Contract Number: RI-312902

Change log

Version	Date	Author/Editor	Reason for change / issue
1	29/05/2013	Cândida G. Silva	Creation
2	30/05/2013	Francisco Brasileiro, Francisco Sanz García, Rui M. M. Brito, Cândida G. Silva	Review and update
3	31/05/2013	Fermín Serrano	Final

Table of Contents

1. SUMMARY

This deliverable characterizes the operational platform for experimental research under development in the SOCIETIC project. This deliverable is under the responsibility of the WP3 leader and includes the contributions of all the other partners involved in WP3.

The basic aspects of the Infrastructure Operation and Deployment (WP3) related with the setup and operation of the hardware and software infrastructure are described and discussed.

2. INTRODUCTION

The principal objective of the WP3 is the setup and operation of the hardware infrastructure. This includes internal servers and the external resources.

Our first step was to analyze existing technologies and resources. Selected ones would be integrated in order to provide a fixed infrastructure to be used by WP4 applications. As technology evolves, new scientific problems arise, some problems related to the infrastructure itself are found and need to be solved. Therefore, this selection of technologies and resources presented here may change throughout the execution of the project, in order to provide the best solution to support scientific experiments as well as to find the most effective way to attract volunteers not only to help professional scientists to solve problems, but also to let general public to propose and create experiments.

After the selection of the technological components, these are deployed under the hardware infrastructure described below. We try to maintain a production branch and two testing branches of all the infrastructure. In one of the testing branch, we test new features while in the other WP4 deploys new experiments.

As mentioned above, while some technological components were selected to start experiments' deployment, we keep testing and evaluating all possible technologies susceptible to be used under SOCIETIC. This technology surveillance is continuously shared among all the partners of the project, thus we also provide some tools to support such communication.

We also need a way to describe the infrastructure and provide mechanisms for the connection with current and foreseen external resources. This will be achieved by publishing APIs for each element in the infrastructure, trying to use standards as much as possible. A major update of the infrastructure is expected to happen in the second year of the project.

The remainder of the document is structured as follows. In Section 3, Infrastructure description, we cover the software and hardware used to support the project. In Section 4, Maintenance and application porting policies, we present the procedures that need to be followed to update the system infrastructure as well as the application porting process. Next, in Section 5, called Interoperability requirements, we discuss extra developments required for providing a minimal level of interoperability for the infrastructure.

3. INFRASTRUCTURE DESCRIPTION

3.1 Hardware

Most of the hardware infrastructure (Figure 1.) of the project is provided by BIFI-UNIZAR, although other partners, mainly UC and UFC, provide their own hardware to install and test software related to the project.

We are providing OpenVZ[1] virtual machines in which we deploy different software components. OpenVZ is a container-based virtualization for Linux. It creates multiple secure, isolated Linux containers on a single physical server enabling better server usage and ensuring that applications do not conflict among them. Each container performance and execution is exactly like a stand-alone server. A container can be rebooted independently and have root access, users, IP addresses, memory, processes, files, applications, system libraries and configuration files. These virtual machines are hosted in 4 physical nodes and are described in Table 1.

Table 1. Description of the physical nodes supporting SOCIETIC infrastructure.

Name	CPU	Mem	HD	OS
srv1.ibercivis.es	Intel(R) Xeon(R) CPU E5520 @ 2.27GHz (x16)	24GB	1TB (Raid 1)	Debian 6.0
srv2.ibercivis.es	Intel(R) Xeon(R) CPU E5520 @ 2.27GHz (x16)	48GB	1TB (Raid 1)	Debian 6.0
srv3.ibercivis.es	Intel(R) Xeon(R) CPU E5520 @ 2.27GHz (x16)	48GB	1TB (Raid 1)	Debian 6.0
srv4.ibercivis.es	Intel(R) Xeon(R) CPU E5520 @ 2.27GHz (x16)	24GB	1TB (Raid 1)	Debian 6.0

3.2 Virtual Hosts

On top of the physical nodes, several virtual machines can be deployed as needed.The processes of creating, destroying, cloning, backuping, and so on are quite straightforward. There are some templates corresponding to different Linux flavors that are used by some shell commands (like vzcreate, vzdump, etc.) to create, dump, etc. these hosts.

One of the advantages of OpenVZ is that it allows move the virtual hosts across the different physical servers. This allows us a great flexibility when doing, for example, maintenance tasks. Nineteen virtual hosts, not all of them related to SOCIETIC project, are currently running in these physical hosts.

[1] http://openvz.org/

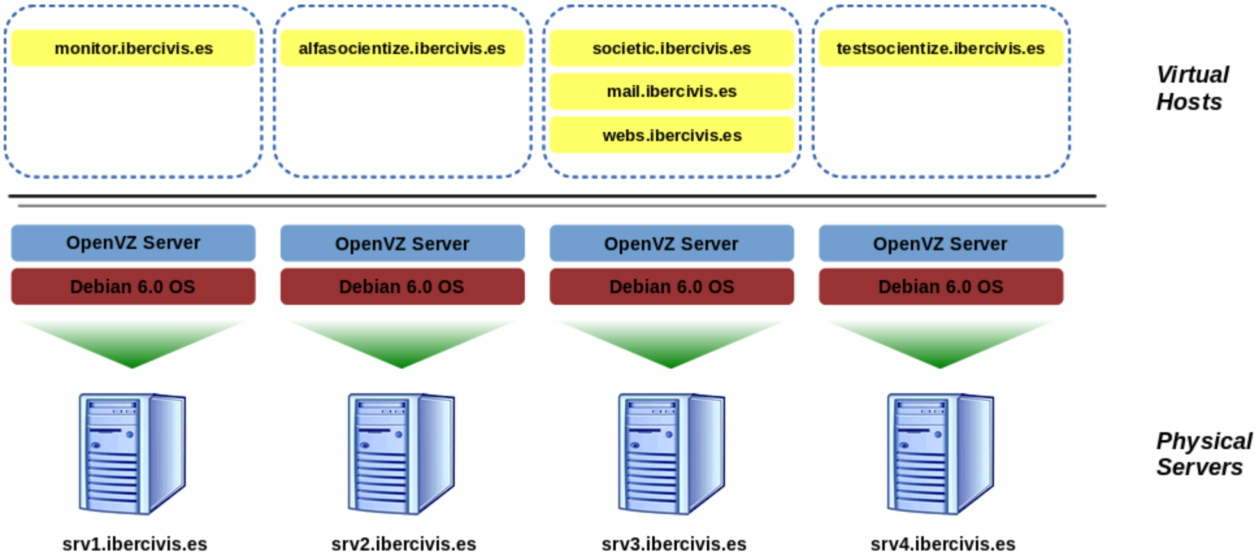

Figure 1. Diagram of the current infrastructure of the SOCIETIC project.

3.2.1 Virtual Hosts related with SOCIETIC

The virtual hosts (Figure 1.) related to the SOCIETIC project are:

- **monitor.ibercivis.es:** This host, running in srv1.ibercivis.es, does daily incremental backups and weekly full backups of the other hosts. This is performed through a software called BackupPC[2] under a 2TB file system mounted using RAID5. We do also a weekly snapshot of the virtual host using the vzdump tool that is stored in the same file system.
- **alfasocientize.ibercivis.es:** At this host we develop PyBossa[3] apps. This server maintains the same configuration as the one present in societic.ibercivis.es, the production PyBossa infrastructure as the. This server is hosted by srv2.ibercivis.es.
- **societic.ibercivis.es:** Hosted under srv3.ibercivis.es**, t**his is our PyBossa production server. We install only stable experiments and stable and tested versions of PyBossa at this server. As abovementioned, this server and alfasocientize.ibercivis.es maintain the same software versions.
- **testsocientize.ibercivis.es:** PyBossa middleware is tested at this server. We use this server to develop new features of this middleware, although our developers are moving to Vagrant[4]+ KVM[5], which allows server software to be easier developed using the developers personal computers. It is hosted under srv4.ibercivis.es.
- **mail.ibercivis.es:** Using Qmail[6], this host is used to serve the emails under the SOCIETIC domain name. It is hosted in srv3.ibercivis.es.

[2] http://backuppc.sourceforge.net/

[3] https://github.com/PyBossa/pybossa

[4] http://www.vagrantup.com

[5] http://www.linux-kvm.org

[6] http://www.qmail.org

- **webs.ibercivis.es:** In this host, we have installed the Drupal CMS that serves the main page of the SOCIETIC project[7]. We have also a MySQL server to support some different applications. This is hosted also in srv3.ibercivis.es.

All hosts described above run under Debian 6.0 operating system.

3.3 Software components

We use several software components to support the SOCIETIC project. Notice that this is not a fixed list. It is expected some updates during (and after) the project. The main components are described in the following:

- **Apache2:** Apache HTTP Server Project is an effort to develop and maintain an open-source HTTP server for modern operating systems including UNIX and Windows NT. The goal of this project is to provide a secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards. We use it, in conjunction with PyBossa, Drupal, etc. to serve almost all the web pages provided by the SOCIETIC project.
- **BackupPC** is a high-performance, enterprise-grade system for backing up Linux, WinXX PCs and laptops to a disk server. BackupPC is highly configurable and easy to install and maintain. Installed under backuppc.ibercivis.es it is used to do daily incremental backups and weekly full backups of all of our virtual hosts.
- **Drupal** is a free and open-source content management system (CMS) written in PHP and distributed under the GNU General Public License. It is used as a back-end system for at least 2.1% of all websites worldwide. Under webs.ibercivis.es our Drupal CMS serves socientize.ibercivis.es.
- **EpiCollect** provides a web application for the generation of forms and freely hosted project websites (using Google's AppEngine) for many kinds of mobile data collection projects. We're starting to use it to create Android and IOS form-like apps.
- **Mailman** is free software for managing electronic mail discussion and e-newsletter lists. Mailman is integrated with the web, making it easy for users to manage their accounts and for list owners to administer their lists. Mailman supports built-in archiving, automatic bounce processing, content filtering, digest delivery, spam filters, and more. Two mailing lists of the SOCIETIC project are supported by this software.
- **MySQL** is the world's most widely used open source relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. Some of our projects (like HappyUp) are using this database. Also, the Drupal CMS uses a MySQL database.
- **OpenVZ**, as aforementioned, is a container-based virtualization for Linux. OpenVZ creates multiple secure, isolated Linux containers on a single physical server enabling better server utilization and ensuring that applications do not conflict. Each container performance and execution is exactly like a stand-alone server; a container can be rebooted independently and have root access, users, IP addresses, memory, processes, files, applications, system libraries and configuration files.

[7] http://www.socientize.eu

- **PostgreSQL** is an object-relational database management system (ORDBMS) available for many platforms including Linux, FreeBSD, Solaris, Microsoft Windows and Mac OS X. It is released under the PostgreSQL License, which is a MIT-style license, and it is thus free and open source software. PyBossa middleware uses this database.
- **PyBossa** is an open source platform for crowdsourcing online (volunteer) assistance to perform tasks that require human cognition, knowledge or intelligence (e.g. image classification, transcription, information location, etc). PyBossa was inspired by the BOSSA crowdsourcing engine but it is written in python (hence the name). It can be used for any distributed task application but was initially developed to help scientists and other researchers crowd-source human problem-solving skills. At this moment, PyBossa is one of the most important software of the SOCIETIC project.
- **Qmail** is a mail transfer agent (MTA) that runs on Unix. It is a more secure replacement for the popular Sendmail program. Qmail's source code is in the public domain.

4. MAINTENANCE AND APPLICATION PORTING POLICIES

In order to avoid long downtimes of the production PyBossa server that supports the applications that are made available by the SOCIETIC project, we have defined a set of procedures that must be followed for the maintenance and upgrade of PyBossa server, as well as for the development and deployment of new applications.

The PyBossa infrastructure is comprised of three distinct servers, each with its own purpose:

- PyBossa production server, hosted at societic.ibercivis.es, is used to deploy the validated SOCIETIC applications, and make them accessible to the general public through the project’s web page;
- PyBossa alpha server, hosted at alfasocientize.ibercivis.es, is used to develop new applications and test their correct functioning before being deployed at the production server;
- PyBossa test server, hosted at testsocientize.ibercivis.es, is used to test new versions of the PyBossa middleware, and our own developments that are potential contributions to the middleware, before they can be deployed in the production and alpha servers.

The procedures that must be followed in order to avoid unnecessary downtimes at the production server, as well as erratic behavior of applications used by the general public are the following:

- The test server is used only to test upgrades in the PyBossa middleware. For a middleware update to be performed at the either the production or the alpha servers, this must be preceded by a successful deploy at the test server. Only after the update is tested and validated at the test server, a middleware upgrade in the production or alpha servers is allowed to occur. The test server has some simple applications deployed that are used to check if everything is working as expected.
- The alpha server must always have the same software version that is deployed at the production server. Upgrades in this server imply upgrades in the production server, and vice-versa. This guarantees that an application that works in the alpha server will also work at the production server. Thus, before being deployed in the production server, applications must first be deployed and tested in the alpha server. Only after this validation is performed, can

an application be deployed at the production server.

5. INTEROPERABILITY REQUIREMENTS

Much of the work of WP3 is to analyse and integrate existing technologies and resources with SOCIETIC components in order to set up the technological ecosystem of this project. This scenario raises the issue of interoperability which needs to be addressed at different levels:

- **Technical interoperability**, associated with hardware and software components, can be accomplished by adapting external infrastructures APIs to fit with the SOCIETIC components, and whenever possible by addressing standardised protocols. The technological solutions explored within the scope of projects like EDGI or DEGISCO by some of the partners of this consortium are to be considered.
- **Syntactic interoperability** is related to data sharing and analysis. The fundamental goal in data interoperability is to facilitate and make transparent to end-users the extraction of information from multiple heterogeneous data sources residing in different locations. The design and management of schema mappings are the standard way to achieve data interoperability. A schema mapping is a specification of the relationship between two distinct file formats (XML, HTML, ...) or database schemas. Approaches for enhancing data interoperability are crucial when considering a collaborative environment of multiple citizen science projects .
- **Authentication** is required or recommended in most of citizen science projects. As in many other projects, applications running on SOCIETIC allow the participation of registered and unregistered volunteers. Overall, we can consider three common ways of registration: (1) by creating an account providing a valid e-mail address (currently implemented on SOCIETIC), (2) by using a federated identity, and (3) by using OpenID credentials. While the first option only guarantees a volunteer's profile closed to a particular citizen science project (or platform such as in SOCIETIC, Zooniverse, and so on), the last two options offer greater advantages in what concerns interoperability and will be taken under consideration. For example, the popular social login (an example of federated identity) allows users to log onto third-party websites, applications, mobile devices and gaming systems with their existing identity in one of the social platforms such as Facebook, LinkedIn, Google, Microsoft, and so forward. This type of authentication mechanism may be interesting to explore when applications on a gaming scenarios are present as it permits users to share the scores and ranking profiles with other connections in their social networks. On the other hand, OpenID intends to offer a unified "web identity" to each internet user, allowing web sites and other people to connect the different accounts the user has created online into a more cohesive persona. Clearly oriented towards interoperability, both options could allow a volunteer to use the same identity across multiple (even if independent) citizen applications, thus allowing him/her the creation of a citizen scientist badge/profile.