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

Deliverable no.	D3.1.2
Deliverable name	Operational Platform for Experimental Research
Dissemination level	PU
WP no.	3
WP name	Infrastructure Operation and Deployment
Date	19/03/2014
Date of delivery	31/03/2014
Actual date of delivery	01/04/2014
Status	Final
Author(s)	Cândida G. Silva, Carlos Val Gáscon
Reviewer (s)	Eduardo Lostal, Adabriand Furtado

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

Change log

Version	Date	Author/Editor	Reason for change / issue
1	19/03/2014	Cândida G. Silva	Creation
2	20/03/2014	Eduardo Lostal	Completed SavingEnergy@home related sections
3	20/03/2014	Adabriand Furtado	Reviewed text about PyBossa-Facebook integration
4	24/03/2014	Carlos Val Gascón	Reviewed text about rate limiting feature
5	25/03/2014	Cândida G. Silva	Completed section on PyBossa major updates. Reviewed text on sections about virtual hosts, and porting and maintenance policies
6	31/03/2014	Eduardo Lostal	Review
7	01/04/2014	Cândida Silva	Final

Table of Contents

1. SUMMARY....4
2. INTRODUCTION....4
3. INFRASTRUCTURE DESCRIPTION....5
  3.1 Hardware....5
  3.2 Virtual Hosts....5
    3.2.1 Virtual Hosts related with SOCIENTIZE....6
  3.3 Software components....7
  3.4 PyBossa major updates....9
    3.4.1 Avoiding malicious users behavior....9
    3.4.2 PyBossa – Facebook Integration....9
4. MAINTENANCE AND APPLICATION PORTING POLICIES....10
  4.1 High Availability....10
5. INTEROPERABILITY REQUIREMENTS....11
6. INFRASTRUCTURE USAGE....12
7. CONCLUSION....12

1. SUMMARY

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

Basic aspects of the Infrastructure Operation and Deployment (WP3) related with the setup and operation of the hardware and software infrastructure are described. Major updates to the infrastructure are found at the software level in particular in what relates to added features to PyBossa, and will described further along the document.

2. INTRODUCTION

The main objective of WP3 is the setup and operation of the hardware infrastructure. The general configuration of the infrastructure was maintained to include a production branch and two testing branches of the Citizen Science infrastructure. In one of the testing branch, we test new features while in the other WP4 deploys new experiments. On the other hand, SOCIENTIZE website and the whole CMS used is maintained in a production branch with a testing branch for new features before moving to production.

Although technological components were selected to start experiments' deployment, we keep testing and evaluating all possible technologies susceptible to be used under SOCIENTIZE. Particular attention is given to open-source tools. This technology surveillance is continuously shared among all the partners of the project,. Additionally, SOCIENTIZE partners have been contributing to the development of new features to Citizen Science tools like PyBossa.

We also need a way to describe the infrastructure and provide mechanisms for the connection with current and foreseen external resources. This is achieved by the use of an API for each element in the infrastructure, making use of standards always that is possible.

The remainder of the document is structured as follows. In Section 3, Infrastructure description, we revise the software and hardware supporting the project development. 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, we address some interoperability issues. In Section 6, entitled *Infrastructure Usage*, we summarize how the infrastructure is used to support the applications being developed.

3. INFRASTRUCTURE DESCRIPTION

3.1 Hardware

BIFI-UNIZAR provides most of the hardware infrastructure (Figure 1) of the project, although other partners, mainly UC and UFCG, provide their own hardware to install and test software related to the project.

OpenVZ[1] is employed to create different virtual machines in which different software components are deployed. 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 described in Table 1.

Table 1. Description of the physical nodes supporting SOCIENTIZE 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. OpenVZ makes the processes of creating, destroying, cloning and backuping virtual machines 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. Another advantage of OpenVZ is that it allows to move the virtual hosts across the different physical servers, which confers the system with a great flexibility when doing, for example, maintenance tasks. Nineteen virtual hosts, not all of them related to SOCIENTIZE project, are currently running in these physical hosts.

[1] http://openvz.org/

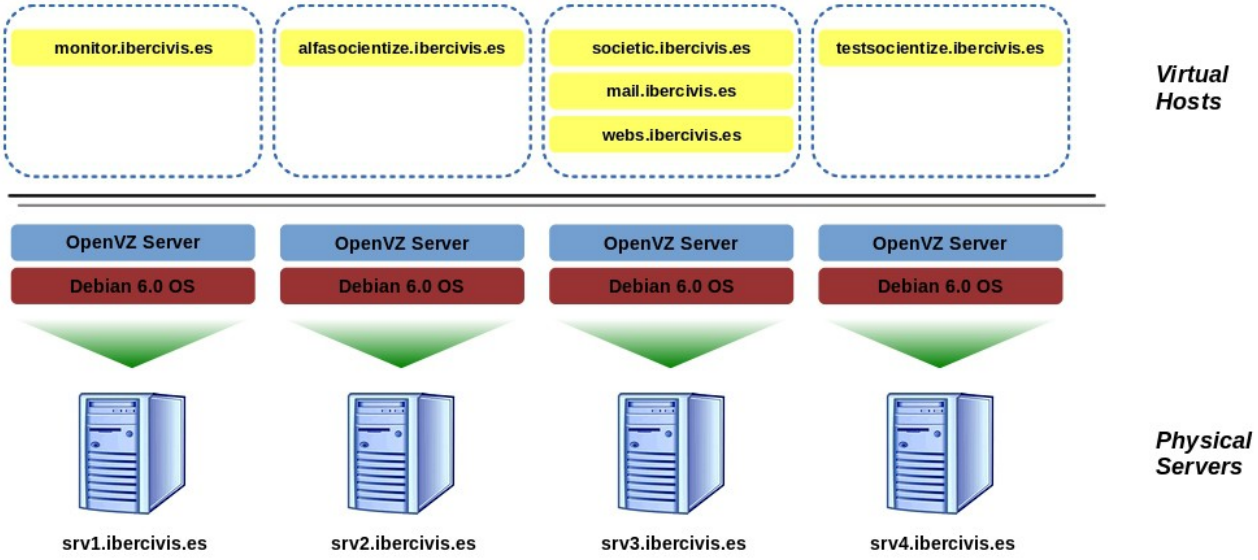

Figure 1. Diagram of the current infrastructure of the SOCIENTIZE project.

3.2.1 Virtual Hosts related with SOCIENTIZE

The virtual hosts (Figure 1) related to the SOCIENTIZE 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:** This server is used for the development of PyBossa[3] and Epiwork[4] applications. This server maintains the same configuration as the one present in societic.ibercivis.es, the production server. This server is hosted by srv2.ibercivis.es.
- **societic.ibercivis.es:** Hosted under srv3.ibercivis.es, this is our production server. Only stable and tested versions of applications, PyBossa and Epiwork are installed at this server. As abovementioned, this server and alfasocietic.ibercivis.es maintain the same software versions.
- **testsocientize.ibercivis.es:** This server is used for testing PyBossa and Epiwork middleware. We use this server to develop new middleware features, although our developers are moving to Vagrant[5]+ KVM[6], which allows server software to be easier developed using the developers personal computers. It is hosted under srv4.ibercivis.es.

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

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

[4] http://www.epiwork.eu

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

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

- **mail.ibercivis.es:** Using Qmail[7], this host is used to serve the emails under the SOCIENTIZE domain name. It is hosted in srv3.ibercivis.es.
- **webs.ibercivis.es:** In this host, we have installed the Drupal CMS that serves the main page of the SOCIENTIZE project[8]. We have also a MySQL server to support 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 SOCIENTIZE 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:

- **Management support**
    - **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.
    - **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 (that is an Apache2 VirtualHost directive under monitor.ibercivis.es) it is used to do daily incremental backups and weekly full backups of all of our virtual hosts.

- **Web and mailing list support**
    - **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 SOCIENTIZE project.
    - **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 societic.ibercivis.es.
    - **Mailman** is free software for managing electronic mail discussion and e-newsletter lists.

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

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

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 SOCIENTIZE project are supported by this software.

  - **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.

- **Database management systems**
  - **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. The Drupal CMS uses a MySQL database.
  - **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.

- **Citizen Science applications middleware**
  - **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). 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 SOCIENTIZE project.
  - **Epiwork** framework, a result of Epiwork project, is a web application for the generation and management of forms used for data collection. It provides the tools to easily create, deploy and manage surveys where users can enter the data gathered. Although it was originally developed addressing epidemic forecast issues, it can be modified to be used in alike projects. Epiwork is built on top of Django. Django is a high-level web application framework, written in Python, which follows the model-view-controller architectural pattern. Its primary goal is to ease the creation of complex, database-driven websites encouraging rapid development and clean, pragmatic design. Django emphasizes reusability and pluggability of components and the principle of do not repeat yourself.

3.4 PyBossa major updates

Two major features were included in PyBossa released in our production server. The first decreases PyBossa vulnerability to malicious users employing robot programs to perform tasks, while the second feature is related to interoperability.

3.4.1 Avoiding malicious users behavior

Following the launching of a contest for teenager students in which most active students in SOCIENTIZE applications would be rewarded with technological gadgets, a series tasks being executed hundreds to thousands of times by the same user or by multiple users have been identified. A careful analysis of the tasks creation timestamps and execution times led the team to believe that a group of users was employing robot programs or alike to perform these tasks. Although these users were notified of their wrong-doing and warned that malicious behavior of the platform was detected, as new students entered the contest the situation tend to repeat itself. These events motivated the consortium to include major updates on the server to prevent the acts of malicious users.

The new code developed follows the same guidelines from Twitter Rate Limiting[9] Calls, and it is based on the code from Armin Ronacher[10] licensed as public domain. The code has been modified to support the Redis Sentinel mode, so the master node is configured via it. Basically, the idea is that we limit a view for a certain period of time and increment a counter. It has been enabled for the endpoints to avoid malicious users submit a large amount of tasks without completing all the required steps in a task. Now a windows of 15 minutes to do a maximum of 300 request per endpoint is allowed.

3.4.2 PyBossa – Facebook Integration

Citizen Science projects are based on *networks* of volunteers. Thus, working on the integration of PyBossa with Facebook not only increases PyBossa interoperability but it also enables to explore social networks when Citizen Science applications are presented, for example, in a gaming scenario, as it allows users to share the scores and ranking profiles with other connections. Furthermore, it may also increase the projects' visibility.

UFCG team has worked on the integration of PyBossa with Facebook. To achieve this integration, it was necessary to perform modifications on the PyBossa Rest API to enable its applications to run inside Facebook interface. These modifications added to PyBossa the capability to authenticate and identify requests to its API from a specific Facebook user.

With this integration, users on Facebook can now contribute to Citizen Science projects that also run on the PyBossa site. The Mind Paths application was used as a case study for this integration and it is now available for Facebook users to play with under https://apps.facebook.com/mindpaths/.

[9] https://dev.twitter.com/docs/rate-limiting/1.1

[10] http://flask.pocoo.org/snippets/70/

4. MAINTENANCE AND APPLICATION PORTING POLICIES

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

SOCIENTIZE infrastructure comprises three distinct servers, each with its own purpose:

- Production server, hosted at societic.ibercivis.es, is used to deploy the validated SOCIENTIZE applications, and make them accessible to the general public through the project's web page;
- Alpha server, hosted at alfasocietic.ibercivis.es, is used to develop new applications and test their correct functioning before being deployed at the production server;
- Test server, hosted at testsocietic.ibercivis.es, is used to test new versions of the PyBossa and Epiwork. Our developments on the middleware features are also made on this server 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 PyBossa and Epiwork. 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-verse. 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.

4.1 High Availability

Users tend to feel themselves frustrated when an online service is not available whatever is the reason: either because of a system failure or a denial of service given too many accesses at a time. Eventually, the result is the same, user is unable to access the service.

Availability is the grade in which an application or service is available when and how users expect to. Main features to be considered are:

- **Reliability:** Both hardware and software are critical elements in order for the system to work properly.

- **Recovery:** Is there a plan to make our application to keep working in case of a failure? How long would it take to restore the system in case of a disaster? These are some of the points to be studied and planned in order to minimize consequences of an unexpected event.
- **Error Detection:** It is necessary to know the status of an element (i.e. failed, saturated, etc) in order to fix it in case of failure. Monitoring is a key point to figure out that status.
- **Constant Improvement:** Maintenance tasks must be transparent for the end user.

In order to provide high availability of our services, we are using Keep Alive, HA Proxy and Redis tools. We changed the engine for caching objects from Memcache to Redis[11] because it is better at managing caching objects and speed up the site. Actually, we had to enable this feature because some elements of the PyBossa servers, like the statistics pages, broke down after a few queries by the users. One example was the querying of the leaderboard.

The tool Keep Alive is being used so that the public IP is always up although even if one server is down; with HA Proxy, we provide load balancing which allows us to scale the system as needed. The third one decreases the number of queries to our database servers, caching them. We have also an active-active database system, thus queries are balanced between them. The infrastructure can continue to function properly even if one of this database servers fails.

5. INTEROPERABILITY REQUIREMENTS

Much of the work of WP3 is to analyze and integrate existing technologies and resources with SOCIENTIZE 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 SOCIENTIZE components, and whenever possible by addressing standardize 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 schema. 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 SOCIENTIZE allow the participation of registered and unregistered volunteers. Currently, a volunteer can create an account by providing a valid e-mail address, or he/she can sign in using a Facebook account.

[11] http://http://www.redis.io/

6. INFRASTRUCTURE USAGE

Currently, SOCIENTIZE infrastructure supports four applications in production phase: semantic maps (*Mind Paths;* mindpaths.socientize.eu), cell images analysis (*Cell Spotting;* cellspotting.socientize.eu), sun images analysis (*Sun4All*; sun4all.socientize.eu), and analysis of households energy consumption (SavingEnergy@Home; savingenergy.socientize.eu (in Spanish and English)).

Different versions of the first three applications have been developed using HTML5 + JavaScript on the client side and Python on the server side running on top of PyBossa. On the other hand, the SavingEnergy@Home application has been developed using Epiwork framework on top of Django. That means that while the client side the application is rendered using HTML and JavaScript, server side is completely working on Python.

7. CONCLUSION

SOCIENTIZE infrastructure is fully operational and supports several Citizen Science projects. Although, at the hardware level no major updates have occurred, at software level some changes have taken place to make it more reliable and to increase its interoperability. Most relevant updates relate to PyBossa integration with Facebook and the decrease of its vulnerability to malicious users by the use of robot programs.