VILNIUS GEDIMINAS TECHNICAL UNIVERSITY

FACULTY OF FUNDAMENTAL SCIENCES

DEPARTMENT OF INFORMATION SYSTEMS

Albert Folch

Interface development for Eucalyptus based cloud

Master thesis

Vilnius, 2011

VILNIUS GEDIMINAS TECHNICAL UNIVERSITY

FACULTY OF FUNDAMENTAL SCIENCES

DEPARTMENT OF INFORMATION SYSTEMS

Head of department

(Signature)

Doc. dr. D.Mažeika

(Name, surname)

2011 d.

(Date)

Albert Folch

Interface development for Eucalyptus based cloud

Master thesis

Advisor doc. dr. D.Mažeika

Vilnius, 2011

Vilnius Gediminas technical university

Fundamental Sciences Faculty

Department of **Information Systems**

ISBN ISSN

No. cop. **1**

Engineering Informatics study programme master's thesis

Title: **Interface development for Eucalyptus based cloud**

Author: **Albert Folch** Academic supervisor: **doc.dr. D.Mažeika**

Thesis language

X English

Lithuanian

Annotation

Nowadays the use of cloud computing is expanding exponentially. With a very short life has become a technology that will. Moving data and applications to Internet and access to them trough services gives some vantages to organizations. Although today there are still some doubts and reservations towards it.

The final thesis work describes and analizes cloud computing world and specially Eucalyptus, a software platform for the implementation of private cloud computing on clusters. As a practical work, it have been release a desktop client for interact with Eucalyptus that allow users to run and connect to virtual machines inside the system.

Keywords: Cloud computing, Eucalyptus, Amazon EC2, Java, Swing, Typica, Scilab.

1. Introduction ..... - 5 -
a. Short overview of cloud computing ..... - 5 -
b. Identify the problem ..... - 7 -
c. Objective of the thesis ..... - 8 -
d. Tasks of the thesis ..... - 9 -
2. Cloud computing ..... - 10 -
a. Historical view ..... - 10 -
b. Overview of cloud computing in details ..... - 12 -
I. Definition ..... - 12 -
II. Architechtures ..... - 14 -
III. Types of clouds ..... - 15 -
IV. Positions about cloud computing ..... - 15 -
V. Technologies related ..... - 19 -
VI. Implementations, providers ..... - 24 -
3. Practical part ..... - 29 -
a. Tasks of the thesis ..... - 29 -
b. Application, Eucalyptus ..... - 31 -
c. The application ..... - 42 -
d. Requirements for the cloud service ..... - 48 -
e. Architecture of cloud service (interface to the Scilab), UML diagrams ..... - 49 -
Conclusions and suggestions for further works ..... - 68 -
4. References ..... - 72 -

1. Introduction

a. Short overview of cloud computing

The use of Internet and new technologies nowadays, for business and for the current users, is already part of everyday life. Any information is available anywhere in the world at any time. That was not possible few years ago. Nowadays it have arisen a lot of possibilities of access to public and private information like internet speed access or the deployment of mobile dispositive that allow the connection to Internet from almost everywhere.

Today a lot of people are consulting their mail online through webmail clients, writing collaborative documents using web browsers, creating virtual albums to upload their photos of the holidays. They are running applications and storing data in servers located in Internet and not in their own computers. Something as simple as enter in a web page is the only thing a user needs to begin to use services that reside on a remote server and lets him share private and confidential information, or using computing cycles of a pile of servers that he will ever see with his own eyes. And every day its being used more this services that are called cloud computer services. That name is given because of the metaphor about Internet, as something than the user see like a cloud and cannot see what's inside.

This services can be offered by free or by paying by demand (pay for consume), can be simply like a function calling (like asking the temperature in some city in the world for include it in a web page) or complex (like the usage of a virtual machine with its own operating system, applications and storage space for running applications).

This means that many users and organizations can avoid install some applications in their computer or can have more computational power using cloud computer through internet, or they can make their own private cloud to manage it completely, or they can use both options for the moments of high demand of consume.

b. Identify the problem

Nowadays a lot of enterprises or users need to process a long amount of information data or need to do some complex operations, for example some mathematical models calculations, and they need a high amount of process power to resolve it. Maybe the power of a personal computer can be not enough to finish in a determinate time one task, and maybe this task is so punctual that is pointless to acquire the hardware necessary to do it. Also sometimes a user wants to use a program for few times, for example for making some practices in university, and maybe it's pointless to make install this user a program, like Scilab, so for both problems we see convenient to bring the possibility of use some programs without the needing of install in own computers and with the possibility of have more calculation power. So we have to bring to the user the necessary tools to be able of executing a problem in the cloud of the university.

As we will see after, Eucalyptus is a software that allows the management of a private cloud infrastructure, we are not going to enter now in technical explanations, but the truth is that a user that wants to run Scilab in the cloud don't need to know about the infrastructure that contain this program, we have to remember than users are working with applications, not with infrastructure. It's also true that for handling with eucalyptus user needs to know their commands and how to use them, so it is needed to deploy a friendly user interface to permit to users open Scilab in the cloud in a comfortable way, forgetting about what it's inside the system.

That is what user sees

USER

SCILAB

This is what really happens

c. Objective of the thesis

The objective of the thesis is develop a user friendly interface to access to eucalyptus private cloud system and run easily applications hosted inside. To achieve this objective we need to found a programming language able to communicate with eucalyptus. Eucalyptus works like Amazon EC2 so it means that all the modules of eucalyptus works like a web service. This web services are compatible with all languages because use standards (xml/soap), so searching a little we can find Typica, it's a Java API that allows communicate with Amazon EC2 and have been tested with Eucalyptus, in a very friendly way. Instead of reinventing again this communication in java we will use this API, its open source and also in deployment, so it means that there's a community of developers behind giving help and actualizing that API.

The communication will be deployed with Java, so to deploy the interface it will be some Java graphic library. Java Swing is a graphic library for Java that allows creating in very easy way graphic interfaces. The IDE NetBeans also have a Swing Gui Builder, also known as project Matisse, it is a graphic interface for the creation of graphic interfaces, simplifying the design deployment of this ones.

This will allow me to understand how it works internally a private cloud, how to interact with it, and understand witch kind of solutions can provide this technology. It's quite interesting the fact that Amazon EC2 and Eucalyptus use a compatible API

because I'm working with an open-source solution that is working like a very used one like Amazon EC2.

d. Tasks of the thesis

First of all it will be necessary to be on date about this entire world called cloud computing. Also its important to be familiar about the history of this product, all the technologies that are in contact with it, which are the conditions that encourage to use it, which are the problems that arise with it uses,… In other words be familiar with this new kind of computation.

Second task will be identify the problem and the requirements of the solution, having in mind who is going to use this and how.

Later it will be necessary design the solution and implement it, and in all this parts will be necessary to write in the documentation of the project what is happening. Finally it will be necessary to try the software created and obtain conclusions.

2. Cloud computing

a. Historical view

In the 60's John McCarthy said that "computation may someday be organized as a public utility". This is just a concept attached to cloud computing. Like if it was electricity supplier, user asks for how much it needs and pays as much as he uses, also there's an illusion of infinite supply. But let's see a little bit of history.

In the early 60's and 70 companies had large mainframe computers which provided services to workers who had access to them through dumb terminals. These computers stored all information and did all calculation. From a dumb terminal orders were sent to the mainframe and it returned the information to that terminal. But these mainframes were very expensive.

In the 80's, companies realized that servers based on normal computers could be installed at lower cost than mainframes. Also this gave users a sense of greater control over their actions. So gradually these mainframes were being replaced by computers for the users, also because of the decrease of the prices of the personal computers.

In 90's with begin of the global use of Internet the fashion of lots of computers accessing to one big server came back again. At that time were required web servers with plenty of power to hold the requests that were made from the Internet. From that time to today more services are offered in Internet and more storage need the users of this services. More and more the logic of the applications is being moved in servers in Internet from the personal computers thanks also to the increase of speed of the access to the net and to the facilities to access (mobile dispositive). Nowadays the massive computation needed can be solved by providers that are dedicated to this, in this way various users can share the same infrastructure maximizing the efficiency of this one and minimizing the coasts.

At the finish of the dot.com bubble, finish of the 90's, normally all data centers were using less than 10% of their capabilities because they wanted to reserve the rest for the occasional peaks, at that time Amazon made a great effort to solve this problem adding capabilities by demand to the users, one of the concepts of cloud computing, fast and easy.

At 1999 Salesforce.com began to delivery services to enterprises by their own website and pioneered the concept of software as a service. In 2002 Amazon launched Amazon Web Services (AWS), a suite that included storage, computation and others services. In 2006 Amazon launched Elastic Compute Cloud (EC2) to small companies and users to let users run their own computer applications in the cloud. In 2008, Eucalyptus was launched, being the first open source AWS API compatible platform for deploying private clouds. In 2009 Google began to offer enterprise applications based in browser as Google Apps.

b. Overview of cloud computing in details

I. Definition

The word cloud it's used to describe this kind of computing because of the metaphor used for describe networks, a cloud that underlie all the technology that is above and the user don't know it exists and don't need to know it. There's not an official definition about what is cloud computing. For answering this question we will make reference to some different definitions offered by important organizations.

The IT encyclopedia **whatis.com** defines cloud computing as:

> "A general term for anything that involves delivering hosted services over the Internet."… "A cloud service has three distinct characteristics that differentiate it from traditional hosting. It is sold on demand, typically by the minute or the hour; it is elastic -- a user can have as much or as little of a service as they want at any given time; and the service is fully managed by the provider (the consumer needs nothing but a personal computer and Internet access)."

For the IEEE **(Institute of Electrical and Electronics Engineers)** cloud computing is

> "A paradigm in which information is permanently stored in servers on the internet and cached temporarily on clients"

For the most famous encyclopedia in Internet, **Wikipedia**, cloud computing is:

> "Computation, software, data access, and storage services that do not require end-user knowledge of the physical location and configuration of the system that delivers the services"

The University of Berkeley, one of the most important technical universities over the world defines the term cloud computing as:

> "Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services"

So we can say that cloud computing is a computational paradigm based on consume of resources, applications, hardware or computation, offered by internet and consumed under demand. These services are public or private, for free or not, and have service level agreements that regulate them. Normally are focused in generalist solutions because are created to be useful for as much user as it can be possible.

II. Architechtures

Cloud computing it's based in the offer of services, we found 3 different kinds of services:

Software as a Service (SaaS)

These services are applications over Internet. Normally the user can run these applications using a web-browser. User abstract totally about the hardware and software that is using and simply access to a interface with a web browser and from there he have access to some information and functionalities. It's dedicated to current users; an example to this kind of services may be Google Docs

.

Platform as a Service (PaaS):

These services are focused on the deployment of applications or services online letting to the developer manage the hardware or software necessary, including also a solution stack. This service includes all the life-cycle of the deployment of application/ service such as design, implementation, testing, deployment, integrity with databases, etc

There are three characteristic points in this service:

- Services for deployment, testing and maintenance of applications
- Multi-user architecture, in other words, scalability.
- Collaborative tools.

An example of these services is Google App engine.

Infrastructure as a Service (IaaS):

These services are focused to offer a computer infrastructure. All the servers, connections, software and other resources are offered by the providers. And the users see it like an entire infrastructure hosted in the same organization.

III. Types of clouds

Public

Public cloud (also known as external cloud), is the traditional way, where services are provided by a third part via Internet, and they are visible to everybody (it doesn't mean that they have to be free). So in the cloud it's the information of lots of users but they can't access of course to the information of the others.

Private

This cloud consists on the hosting of private applications, storage, or computation in the same company emulating a cloud in Internet but only for private use (private networks). The coast of infrastructure and maintenance of it is the same that having it in normal way but the scalability and the sharing of the coasts is better...

Hybrid

It's a combination of public and private cloud. An organization can have a part of their services in its own infrastructure but also in public cloud. Or can use the public just when have peaks of usage. It's a good option when you want to have your data or application in local and don't want to invest too much in infrastructure.

IV. Positions about cloud computing

The most valuable points of cloud computing are:

Less initial investment

At the beginning, one organization needs to buy the entire infrastructure that needs for beginning to run a project. It means a lot of investment in computer infrastructure. If this organization has all the in-house it means that it should acquire some servers and personal computers powerful enough to assist all the needing of the organization. If this organization begins to use some services in the cloud, it can invest less money in this infrastructure and invest it in other areas of the project.

Costs reduction

Because of payment by demand, just pay what is being used, and because it‘s not necessary to have employers focused on the maintenance and adequacy of the infrastructure or software that is used by cloud computing.

New functionalities and actualizations

The software updates are controlled by the provider of the service, this provider will be interested in actualize all the products that they offer as soon as possible to attract more clients. So the organization don‘t need to be worried about this stuff and don’t need special workers focused in this topics.

Organization focused in business

With last points we can observe that the organization can focus their energies more in business area and not so much in the technical one.

Access to data

As this services are internet focused it‘s easy to access to all the data of one organization or to his information through every simple device with internet

connection, so it‘s very convenient for that organizations that have multiple access points.

Experts about cloud computing identifies next points as possible problems about the use of cloud computing:

Availability of Service

There‘s a big preoccupation in the users of cloud computing, it‘s how reliable is the service, because the enterprises needs data and other services 24 hours day. Providers cannot guarantee 100% of connectivity but their rates of availability of service are pretty high. Providers offers a contract, Service Level Agreement (a part of a service contract where the level of service is formally defined) but sometimes it’s difficult to know how critic can be lose a service for few hours.

Data Lock-in:

The APIs of cloud computing are still no standardized, so it‘s not easy to share data between providers in easy way. Also it‘s difficult to use in same way two different providers and also means that client see that providers have more power than themselves, because if an enterprise wants to change of provider it will be difficult to change all services and data and this creates, creating distrust in clients.

Low reliability in the security of data

Information is the one of the most valuable active in an enterprise, so it‘s a very important decision how to keep it. It‘s normal to think that have this information outside of the organization can be a problem. Also managers of organizations are usually conservative in this kind of decisions so it’s still a problem. Normally organizations decide to locate non critical data in the cloud and keep the confidential one hosted in the organization.

Low performance/Points of failure

The latency and speed of the networks can be a bottleneck easily. The performance of our system can be affected specially in the IaaS service, where we need big amounts of data transfer. Also we get two more different points of failure: the connection of the own organization and the connection of the provider.

Difficult to customize the application:

Services offered in the public cloud are focused to thousands of users, are not focused in particular problems, are just focused in general solutions and usually don't admit much personalization. It means that it's hard to find dedicated applications compared to the in-house software market where we can solutions to almost all necessities.

V. Technologies related

In this section are going to be explained some technologies that are attached to cloud computing.

Web Services

It‘s explained before that cloud computing it‘s based in the offer of services, but what is exactly a service and how it works? Normally two applications written in different languages and executed in different operating systems are not able to communicate between them, but exists a group of protocols and standards for exchanging information between different applications, no matter in which language are written or in which operating system are running. This is really useful, for example over Internet, when we don’t know how is implemented an application running on the web. This can be possible thanks to organizations like OASIS (Organization for the Advancement of Structured Information Standards) and W3C (World Wide Web Consortium). This communication is based in plain text. So it‘s not the same efficient as other kind of communication like CORBA or RMI. (Remote invocation method)

The different technologies used for this communication are:

XML (Extensible Markup Language)

It’s a metalanguage based on labels. It’s very structured and allows to different applications or systems to communicate between them. As it is said in the name is extensible, that it means that it allows defining new labels easily.

SOAP (Simple Object Access Protocol)

It’s a protocol that defines how two objects of different processes can exchange information through XML messages.

WSDL (Web Services Description Language)

Based on XML and it's the public interface in the world of web services. It describes all the services that exist in one location and the way to interact with them.

UDDI (Universal Description, Discovery and Integration)

It's a catalogue of all web services that exist in Internet, it was an industrial initiative but nowadays it's not really used. It's written in XML.

WS-Security (Web Service Security)

It's an extension to SOAP to apply security to web services.

Cloud computing security

It's a set of policies about the security in the cloud focused on computer, network and mostly information protection. These policies are about:

1. Data of one user have be totally isolate from data of another and it have to be moved securely inside the cloud.
2. Use of identities.
3. Physical infrastructure has to be completely secure and the access to the data needs identification.
4. Access to data have to be always available for the users, it can't happen that one user cannot access to his own data at one moment.
5. All the services delivered in the cloud have to be secure.
6. All sensible data have to be encrypted.

Datacenters

The enterprises that need to storage and process a lot of information use normally data centers. This data centers usually are racks inside rooms dedicated to that with special conditions. Datacenters must include lot of several security policies and redundancies of data, power and communication supply. And have to be in special conditions, like temperature, humidity, etc.

Cloud storage

It‘s the virtualization of the storage of data using the net and usually by third parts. There are many companies that have huge data centers that allow to others to store on them their date by using virtual servers and storage pools. So the client sees their files, which are organized among in different locations in the data center, as if they were located physically in the same place. This is very useful because enterprises don‘t have to be worried about the infrastructure of their data center but also they pay for how much data they are storing there.

Virtualization of computers

Virtualization is an abstraction of the resources of a machine between the hardware and the operating system, making a virtual version of a resource being possible to share it.

About virtualization of computers, software simulates an entire machine inside of another one. Normally is an operating system that is executed as if it was the only one running in the computer, but that‘s not real because it’s running together with other operation systems and don’t have all the resources of the machine, just a part of them. Many users can have access to one virtual machine, they feel that they are working alone on that, with access to everywhere but they are just using a part of the machine. That software have to manage the principal resources of computer, CPU, RAM, storage and network and the system has to be able to allow all users work together. This virtualization can be:

- Complete, in this case the hardware simulated is enough to run a normal operating system for the same machine as if it was running completely normal.
- Partial, It‘s a partial virtualization, share resources and host processes, so it‘s a virtualization of the space of addresses.
- Virtualization for O.S, install one O.S inside of another one by using a virtual machine. The O.S virtualized is not as potent as it could be if it was the only one running. With this option we can run different O.S in the same machine.

Green IT

It’s the fashion to maximize the efficiency of the computational resources to minimize the environmental impact. Also includes the deployment of ecologic products. Some technologies included in Green It are cloud computing, grid and datacenters.

Cloud computing is part of Green IT because with the usage of dynamic resources the enterprises minimize the energetic consumption.

Grid

Infrastructure that integrates different systems from different institutions like if it was only one (computers, networks, databases, etc) allowing the collective use. This is the difference with cloud computing, because in cloud all the infrastructure is managed by one organization although each cloud can use dynamically others clouds too.

Virtual Appliance

It‘s a virtual machine compressed in a file, with a program like Xen or VmWare it‘s possible to run this Virtual Machine inside the host. Generally this VM are created with an operative system and some applications installed for one of the next purposes:

- Virtual servers.
- Routers and Firewalls.
- Monitoring.

VI. Implementations, providers

Google Apps

It's a office suite offered as a service (SaaS) that everybody can use through a web-server. Includes the next applications:

- **Gmail**, it's one of the most used email (webmail, POP3 and IMAP) service all over the world. The stable version was released in 2007 and nowadays it has 193.3 million users.
- **Google Calendar**, a calendar service compatible with other standards.
- **Google Docs**, a service to write and edit documents in a collaborative way in real time, with the possibility of sharing them with other users.
- **Google Groups**, a service offered for discussion groups.
- **Google Talk**, a service about instant messaging and video chat.
- **Google Sites**, a service for creating and editing web pages and intranets in an easy way.

Google App Engine

It‘s a platform for the deployment and hosting of web applications (PaaS). These applications are hosted and running in Google Datacenter. It permits to deploy programs in Java (and all technologies that use JVM as Ruby) and Python, and the usage of different frameworks. This service has a limit of 1 GB of hosting (virtual file system) and 5 million pages visited per month for free. If the application exceeds this levels the payment is made by usage, user of App Engine pays per demand, totally dynamically. This service also gives a test environment.

Amazon EC2

It‘s a service (IaaS) that allows users to rent virtual computers. It brings the possibility to run any virtual image with any software installed. It‘s possible for the user to create his own image but also it brings the possibility of use some predefined ones with different operative systems (Red Hat, Suse, Windows, Ubuntu, Solaris,...) also with some software for databases (MySQL, Oracle,...), data processors (Hadoop), web hosting (Apache, ...), IDE‘s and application servers.

It’s a scalable server, user can ask for more capacity and the response is given in some minutes. It’s completely compatible with other Amazon services like S3 and brings the possibility of paying for hours or for running instances.

It offers 10 different kinds of virtual machines with different processors.

Amazon S3 (Simple Storage System)

It is a service that offers online storage through a web service. It uses standard interfaces like REST or SOAP for being compatible with all kind of systems. For download data the default protocol is HTTP but it‘s possible to use Bit Torrent also. It storage the data with fecundation and secure against intrusions and it checks periodically the integrity of the data. The coast of the usage is measured in Gigabytes/month, so the coast is totally dynamic and depends in how many data the client store.

Microsoft Online Services

Microsoft is also in the cloud offering *software plus services*, this is a variation that consists in the combination of some services hosted in the cloud with local applications also running in the client computing. This permits to the user some vantages like the possibility of running application online but keeping data in the personal computer of the client. The services provided are focused on the communications. They offer:

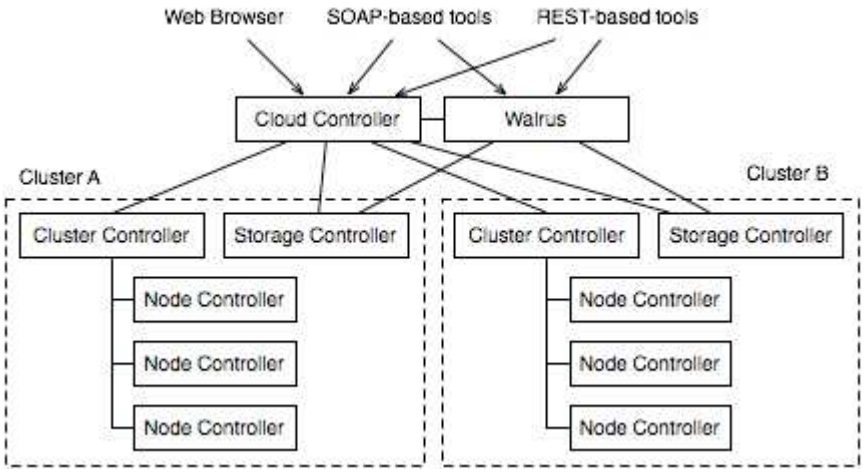
- **Microsoft Exchange Online***:* A service for sending messages with email included based on Exchange Server.
- **SharePoint Online**: It‘s a location in the net where users can collaborate together. Share resources and work in group.
- **Office Communications Online***:* Communication service with instant messaging (IM) and videoconference.
- **Microsoft Forefront**: Security software that provides security to networks, servers and other devices. Including antivirus, anti-malware, anti-spam and other protections tools.

- **Microsoft Office Live Meeting**: It's a web conference service, using a central server it allows to do conferences in easy way.

Eucalyptus

Eucalyptus **(Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems)** it's a open-source software framework for implement a private cloud (IaaS). It gives the possibility to the users to run and control different virtual machines using an entire physical infrastructure. The API of Eucalyptus is based in the API of Amazon EC2 and Amazon S3 giving compatibility with these important services.
Eucalyptus has a modular and hierarchical architecture and each module is implemented as a Web-Service to give more compatibility and security. The API is defined in a WSDL document and the security it's controlled by WS-Security.

We found four modules, the node controller, the cluster controller, storage controller and cloud controller.



Node Controller

This module is in the lowest level, its executed in each physic node where we want to run different virtual machines, it's the responsible to ask the queries of the Cluster Controller. Can make queries to the node to know about physical description of the host and control the state of each virtual machine. It's also the module that runs and terminates VM instances.

Cluster Controller

This module it's located between Node Controller and Cloud Controller, it is usually a cluster front-end machine. It is the responsible of managing the node controllers that are in the same cluster, basically it has 3 functions: schedule run requests, control the Virtual Network Overlay, and exchange information with the Node Controllers.

Storage Controller

Eucalyptus use Walrus, a service that is responsible of the storage for the Virtual machines. Have an interface compatible with Amazon S3 implementing REST (Representational State Transfer) and also is the responsible of stream data inside and outside the cloud.

Cloud Controller

It's a collection of web services grouped in three different categories:

Resources Services: Interaction with the CC to allocate physical resources.

Data Services: Controls movements of persistent data.

Interface Services: Generates user visible interfaces

Virtual Network Overload

All VM have to have connection with each other and also to Internet. But there are some limitations, if there are different users in the cloud; the VM of one cannot connect with the VM of the other.

3. Practical part

a. Tasks of the thesis

The project has 4 main tasks, and the each task have different subtasks, the main tasks are information about cloud computing, connection with Eucalyptus, graphic interface and final deployment and testing.

Cloud computing

This task consists in being familiar about cloud computing. Getting information about what is, understanding the way it works, which kind of technology is used, which kinds of cloud computing exists, identify providers and consumers of this technology, search information about infrastructure as a Service, about Amazon EC2 and Eucalyptus and other services providers.

Connection with Eucalyptus

This task consist in get information about the ways of accessing to Eucalyptus, search the existing APIs, and select the most appropriate to work with. Be documented about this API, design the solution of the problem and deploy it.

Graphic interface

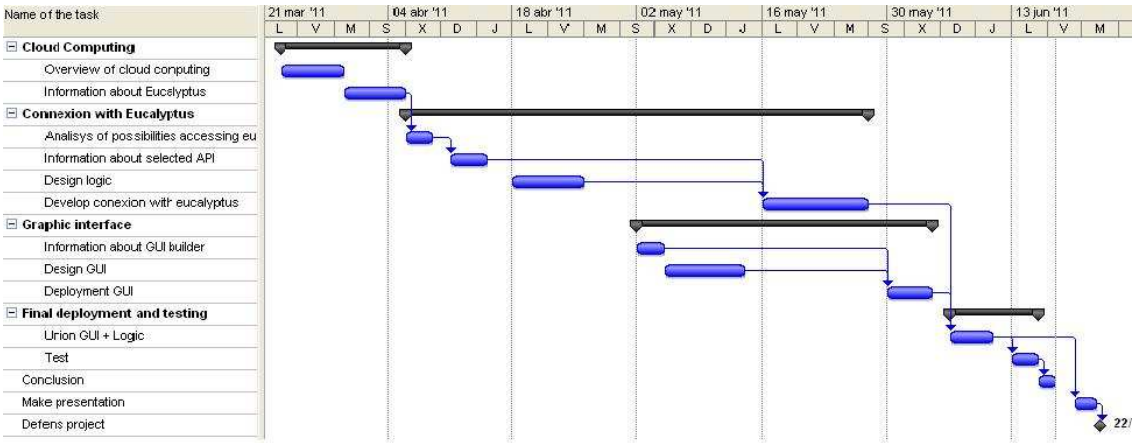
In this task is included the decision of selecting one graphic library and about one GUI Builder to make easier the deployment of the interface, Design it and finally deploy it.

Final Deployment and testing

It's included the union between the logic and the interface of the program and some tests for detecting problems.

And also it is needed to write the memory of the program, as it will be written while the project is being done we can't assign a task for itself, so it means that each task have included the documentation part.

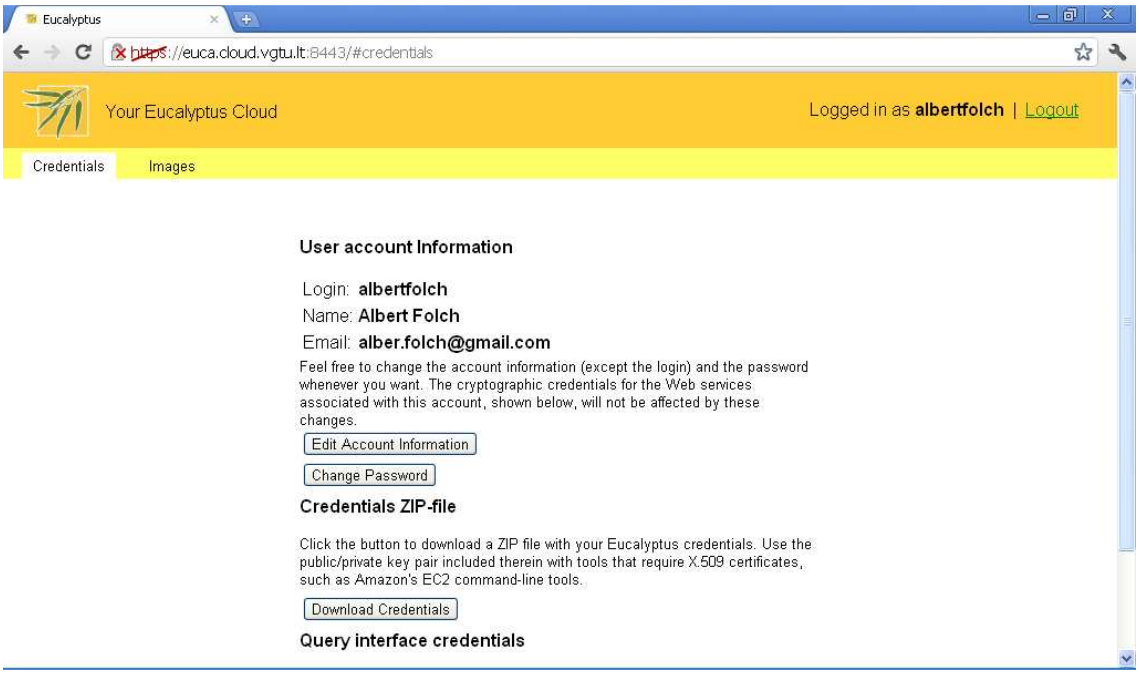
		Name of the task	Duration	Beginning	Finish	Predecessors
1		**Cloud Computing**	**10 días?**	**mié 23/03/11**	**mar 05/04/11**	
2		Overview of cloud computing	5 días?	mié 23/03/11	mar 29/03/11	
3		Information about Eucalyptus	5 días?	mié 30/03/11	mar 05/04/11	
4		**Connexion with Eucalyptus**	**38 días?**	**mié 06/04/11**	**vie 27/05/11**	
5		Analisys of possibilities accessing eu	3 días?	mié 06/04/11	vie 08/04/11	3
6		Information about selected API	4 días?	lun 11/04/11	jue 14/04/11	5
7		Design logic	6 días?	lun 18/04/11	lun 25/04/11	
8		Develop conexion with eucalyptus	10 días?	lun 16/05/11	vie 27/05/11	7;6
9		**Graphic interface**	**25 días?**	**lun 02/05/11**	**vie 03/06/11**	
10		Information about GUI builder	3 días?	lun 02/05/11	mié 04/05/11	
11		Design GUI	7 días?	jue 05/05/11	vie 13/05/11	
12		Deployment GUI	5 días?	lun 30/05/11	vie 03/06/11	10;11
13		**Final deployment and testing**	**8 días?**	**lun 06/06/11**	**mié 15/06/11**	
14		Union GUI + Logic	5 días?	lun 06/06/11	vie 10/06/11	12;8
15		Test	3 días?	lun 13/06/11	mié 15/06/11	14
16		Conclusion	2 días?	jue 16/06/11	vie 17/06/11	15
17		Make presentation	2,5 días?	lun 20/06/11	mié 22/06/11	14
18		Defens project	0 días?	mié 22/06/11	mié 22/06/11	17



b. Application, Eucalyptus

First of all there's the needing to demand to administrator of the system an account, it's easy to do because eucalyptus provide a login menu in its service to login or ask for an account.

Once we have access to our account we can download the credentials to access to the system and change the information of our user. Also we can see the images that the cloud can provide us. With these credentials we will have access to the cloud and ask for running images, and for accessing to these machines. We will have this menu like a part of the eucalyptus users where everybody can change their information and also check the images that are prepared for him.



Id	Name	Kernel	Ramdisk	State
emi-EFAF154F	CentOS-5.5-ansys-img/CentOS-5.5-ansys.img.manifest.xml	eki-0E781872	eri-541F196E	available
eki-24FB18C9	CentOS-5.5-matlab-kernel/vmlinuz-2.6.18-194.11.3.el5xen.manifest.xml			available
eki-240218B4	CentOS-5.5-scilab-kernel/vmlinuz-2.6.18-194.11.3.el5xen.manifest.xml			available
eri-6AF519A5	CentOS-5.5-matlab-initrd/initrd-2.6.18-194.11.3.el5xen.img.manifest.xml			available
eki-0E781872	CentOS-5.5-ansys-kernel/vmlinuz-2.6.18-194.11.3.el5xen.manifest.xml			available
emi-16D615C8	CentOS-5.5-matlab-img/CentOS-5.5-matlab.img.manifest.xml	eki-24FB18C9	eri-6AF519A5	available
emi-161B15C9	CentOS-5.5-scilab.img/CentOS-5.5-scilab.img.manifest.xml	eki-240218B4	eri-6A1B19A2	available
eri-6A1B19A2	CentOS-5.5-scilab-initrd/initrd-2.6.18-194.11.3.el5xen.img.manifest.xml			available
eri-541F196E	CentOS-5.5-ansys-initrd/initrd-2.6.18-194.11.3.el5xen.img.manifest.xml			available

For connecting with the virtual machines that are running in Eucalyptus (we have to remember that for the user are like normal machines) it's possible to connect via SSH and also via VNC. For connecting from windows we can use a program like PuttySSH for connecting via SSH and for VNC RealVNC.

SSH

It's a protocol for accessing to remote machines through a network using a secure channel. It was designed to apply more security to the connections between devices. It permits to use the machine through a command line terminal and it also permit execute graphic programs. Use a secure connection encrypted and also permits copy of data. SSH uses public-key cryptography for the authentication. It exist two keys, one public and one private, so we have to keep the private key. It means that when one message it's sent, we can only decrypt it applying the private key. This program is included in UNIX systems but for windows it's needed to install software. For example putty SSH.

VNC

It's a system that permits to share a desktop from one machine (server) to another (clients). It permits the uses of keyboard and mouse and others devices for the communication. There are no restrictions in this connection, the system don't need to use the same operating system, just it's necessary the installation of one application. It uses RFB protocol, it's a protocol for the accessing to graphical interfaces. In windows we can use for example RealVNC.

Eucalyptus API

Eucalyptus API is based in SOAP, and the requests and response of Eucalyptus services use standards. Any language that supports SOAP communication can use the

Eucalyptus API. Also all SOAP request have to be sent by HTTPS and also it's applied WS-Security standard in this service.

One example is the operation run instances. As we see the message that is send contains for each item of the set of instances, some information.

```
<RunInstances xmlns="http://ec2.amazonaws.com/doc/2011-05-15/">
  <instancesSet>
    <item>
      <imageId>ami-60a54009</imageId>
      <minCount>1</minCount>
      <maxCount>3</maxCount>
    </item>
  </instancesSet>
  <groupSet/>
</RunInstances>
```

All the queries are described in a WSDL document, the public interface of the service; we can download it from this web page http://s3.amazonaws.com/ec2-downloads/ec2.wsdl. For example the query in the top satisfies the calling specified in the WSDL.

```
<operation name="RunInstances">
    <input message="tns:RunInstancesRequestMsg"/>
    <output message="tns:RunInstancesResponseMsg"/>
  </operation>

<message name="RunInstancesRequestMsg">
  <part name="RunInstancesRequestMsgReq" element="tns:RunInstances"/>
 </message>
<xs:element name="RunInstances" type="tns:RunInstancesType"/>
    <xs:complexType name="RunInstancesType">
```

```
      <xs:sequence>
        <xs:element name="imageId" type="xs:string"/>
        <xs:element name="minCount" type="xs:int"/>
        <xs:element name="maxCount" type="xs:int"/>
        <xs:element name="keyName" type="xs:string" minOccurs="0"/>
        <xs:element name="groupSet" type="tns:GroupSetType"/>
        <xs:element name="additionalInfo" type="xs:string" minOccurs="0"/>
        <xs:element name="userData" type="tns:UserDataType" minOccurs="0"/>
        <xs:element name="addressingType" type="xs:string" minOccurs="0"/>
        <xs:element name="instanceType" type="xs:string"/>
        <xs:element name="placement" type="tns:PlacementRequestType"
minOccurs="0"/>
        <xs:element name="kernelId" type="xs:string" minOccurs="0"/>
        <xs:element name="ramdiskId" type="xs:string" minOccurs="0"/>
        <xs:element name="blockDeviceMapping" type="tns:BlockDeviceMappingType"
minOccurs="0"/>
        <xs:element name="monitoring" type="tns:MonitoringInstanceType"
minOccurs="0"/>
        <xs:element name="subnetId" type="xs:string" minOccurs="0"/>
        <xs:element name="disableApiTermination" type="xs:boolean" minOccurs="0"/>
        <xs:element name="instanceInitiatedShutdownBehavior" type="xs:string"
minOccurs="0"/>
        <xs:element name="license" type="tns:InstanceLicenseRequestType"
minOccurs="0"/>
        <xs:element name="privateIpAddress" type="xs:string" minOccurs="0"/>
        <xs:element name="clientToken" type="xs:string" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
```

And we can find the types, as we see it's not necessary to specify all the fields.

The response obeys the same WSDL so once we understand how works it's easy to deploy libraries for all the languages possible, we just have to make SOAP queries and

know which kind of response we will have. In general, the response data types are named according to the operation performed and whether the data type is a container. All the response has a unique ID for if there's any trouble with the response. This is an example of a response to the function describeKeyPairsResponse that returns all the keyPairs that the user have registered in Eucalyptus in form of list with all attributes inside.

```
<DescribeKeyPairsResponse xmlns="http://ec2.amazonaws.com/doc/2011-05-15/">
 <requestId>7a62c49f-347e-4fc4-9331-6e8eEXAMPLE</requestId>
 <keySet>
  <item>
   <keyName>gsg-keypair</keyName>
   <keyFingerprint>
     1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f
   </keyFingerprint>
  </item>
 </keySet>
</DescribeKeyPairsResponse>
```

Typica API

Typica is a client library, written in java, for the communication with Amazon EC2 and another Amazon's web services, as we know Eucalyptus it's compatible with Amazon EC2 so it's possible to use Typica API for communicate with Eucalyptus also. The code of Typica is open-source and it's possible to download from this web page http://code.google.com/p/typica/source .
These are the main part that has been used in the deployment of this project.

Connection to System

For connecting to Eucalyptus system you need one user and his key authorized to enter in the system. Of course we also need to specify the IP of the cloud controller

and the port where this controller is listening petitions. For this process we should create an object Jec2 that we find in the package com.xerox.amazonws.ec2.

Jec2(String awsAccessId, String awsSecretKey, boolean isSecure, String server, int port)

With this object we can send requests to system to get information and to send orders as we will see after. So we can say that Jec2 it's a wrapper for EC2 and is the main class for accessing to the system.

Images

Private cloud systems have their own images registered. Administrator can upload these images to the cloud and the users can run them in virtual machines creating then an instance of this image. We can get Information about the available images with some filters, for example we can specify a list of user and get information of all the images that this user is owner, but basically we can get information of these images with the next procedure of the Jec2 object.

public List<ImageDescription> describeImages(String[] imageIds)
throws EC2Exception

We get a List of ImageDescription with all the information about the images, like the id, the owner, location, the state (if it is ready or not), the platform in which one is running, architecture, assigned kernel, the type of image, etc

Keys ssh

For having secured access to the instance we need to connect for example with SSH, so we need to create users and their private and public keys. This is easy to do with the next function

createKeyPair(String keyName)

deleteKeyPair(String keyName)

This function return a class called KeyPairInfo (com.xerox.amazonws.ec2.KeyPairInfo) that contains the name of the user, the fingerprint(private key), and the material (public key). Now we can access with this user to the instances that we create. Also the system keeps this information.

Security Groups

A security group is a collection of access rules; these ones specify the network traffic access policy of the instances. These rules can be modified in any time and you can add to the instances all the rules that you want, when a rule is changed all the instances that are running with this rule also notice the change. In these rules is specified the accepted traffic, for one IP, one initial port to one finish, and the protocol (tcp, udp).

If an Instance is running without any security group eucalyptus assigns the default group to that instance. This group allows all traffic between instances of the same group but discard all datagram from other ip. These are some functions.

createSecurityGroup(String name, String desc)
deleteSecurityGroup(String name)
describeSecurityGroups(String[] groupNames)
authorizeSecurityGroupIngress(String groupName, String ipProtocol, int fromPort, int toPort, String cidrIp)

GroupDescription class (com.xerox.amazonws.ec2.GroupDescription) have information about the security groups and you can get it with describeSecurityGroups function, this class contains the group name, the description, the owner, and a list of permissions.

Instances

The images are ready to be running in virtual machines, now we need to initiate an instance. For running an instance it's necessary to launch that function

ReservationDescription runInstances(LaunchConfiguration lc)

Launch configuration it's a class used for capture the necessary parameters to initiate one instance. This class contains information about the image, the state of the instance, the name of instances launched, the key name for accessing, the private and public (if it have associated one) IP address, the security groups that have and other characteristics. These parameters are used for the launching of the instance. We can also specify which groups of users will use this instance for apply all the security policies for the exchange of traffic.

We have also functions for start, stop reboot and terminate instances, and also function for getting information about the instance.

terminateInstances(String[] instanceIds)
stopInstances(String[] instanceIds)
This function gets a list of the running instances and their state and configuration.
describeSecurityGroups(String[] groupNames)

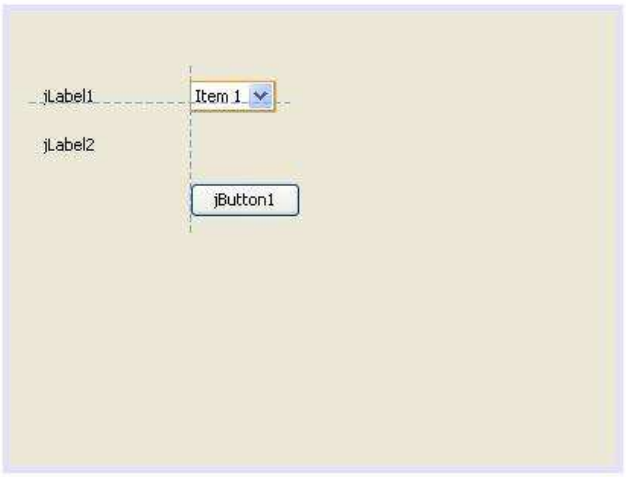
Typica dependencies

For run Typica API there are some associations of other libraries that may been included in the project:

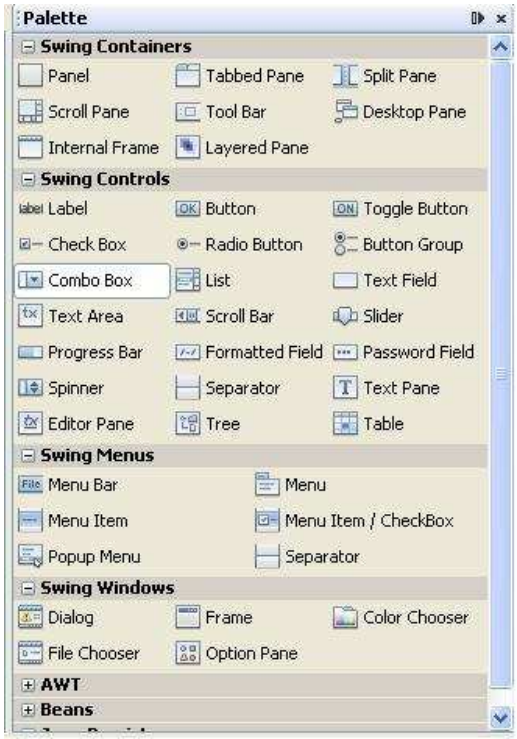
- commons-logging (http://commons.apache.org/downloads/download_logging.cgi)
- JAXB https://jaxb.dev.java.net/servlets/ProjectDocumentList?folderID=6746&expandFolder=6746&folderID=3952)

- httpclient 4 (http://hc.apache.org/downloads.cgi)
- commons-codec (http://commons.apache.org/downloads/download_codec.cgi)

Java Swing Netbeans

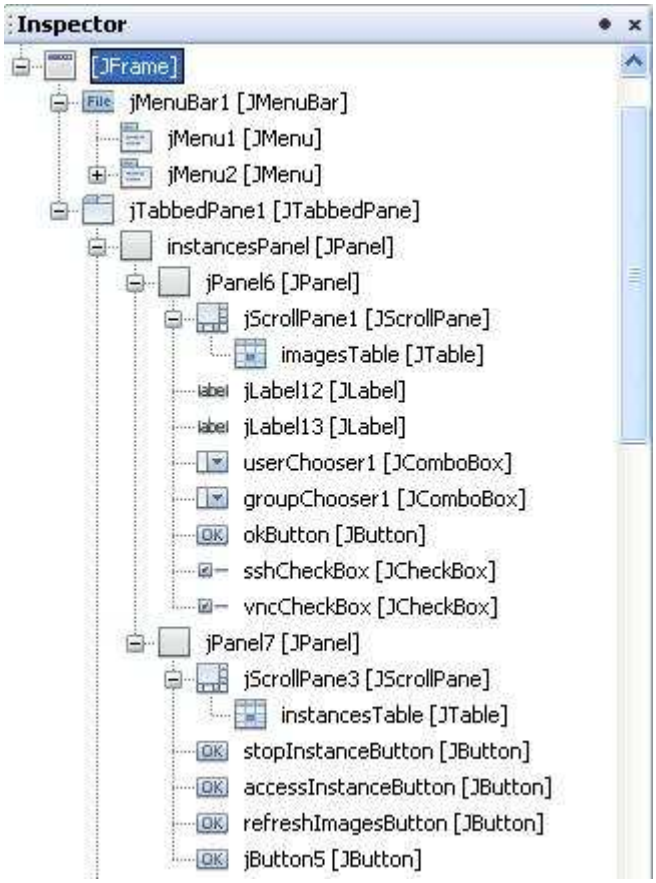


For the designing of graphic interfaces as we remind is going to be used the IDE Netbeans with the Swing GUI Builder (Also known as project Matisse). Java Swing is an API for Java that includes a lot of predefined graphical widgets easy to modify and locate in the interface, as buttons, labels, text areas, emergent messages,… It's a renovation of the old system AWT (Abstract Window Toolkit) for making it more sophisticated. It uses the model view controller pattern.



With this builder it's really easy to design an interface and add listeners to each action that it's important to capture. It also permits to modify the objects (widgets) with a lot of facility. And also it's possible to check the source code and change things there; also we can customize almost all the code created.

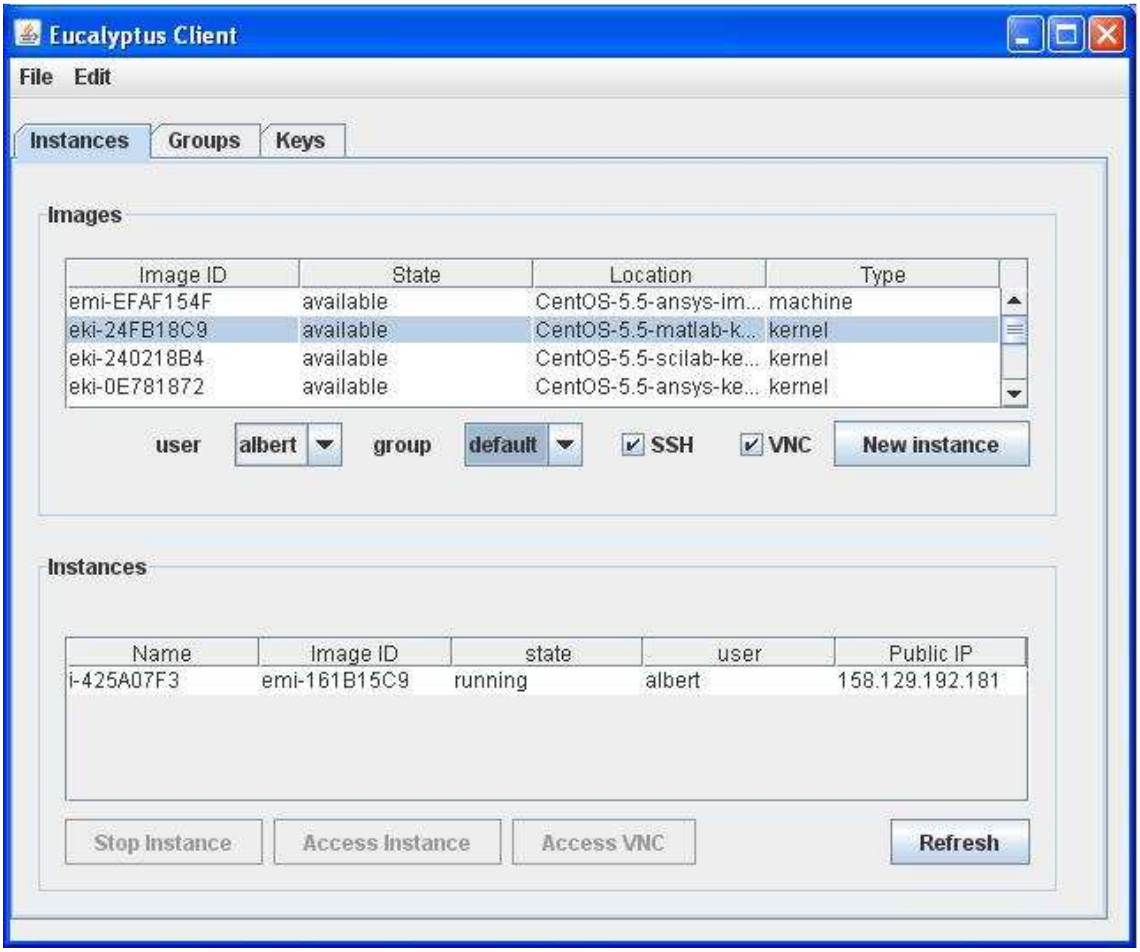
Also we have an inspector of the view where we can see the hierarchy and the elements we have inside one Frame, for example relating to the view of the Images and Instances that exist in the System and we can see in the next page we have the following Inspector capture.



As we can see the main container it's an object of the class JFrame, to this object are added the MenuBar in the top of the interface and the TabbedPane that is the container that allow us to change of Panel. This TabbedPane have three different panels, one for tab. The images and instances one then have two more jPanel inside. It is made like this to differentiate inside the same view the two main sections that we have: instances and images. And in each one we can see all the elements like buttons, labels, textfields, checkbox, table,… It's important to make a hierarchy like this and try to add elements to JPanels and not directly to the view because if we want to make changes any day and we want to move an entire view we only have to move the panel that contains this view.

c. The application

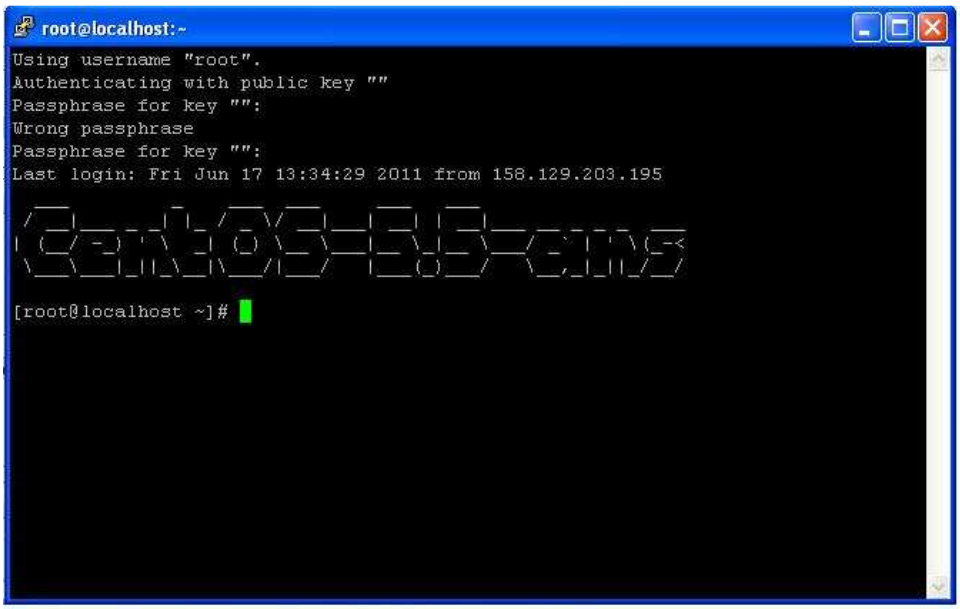
Instances and Images



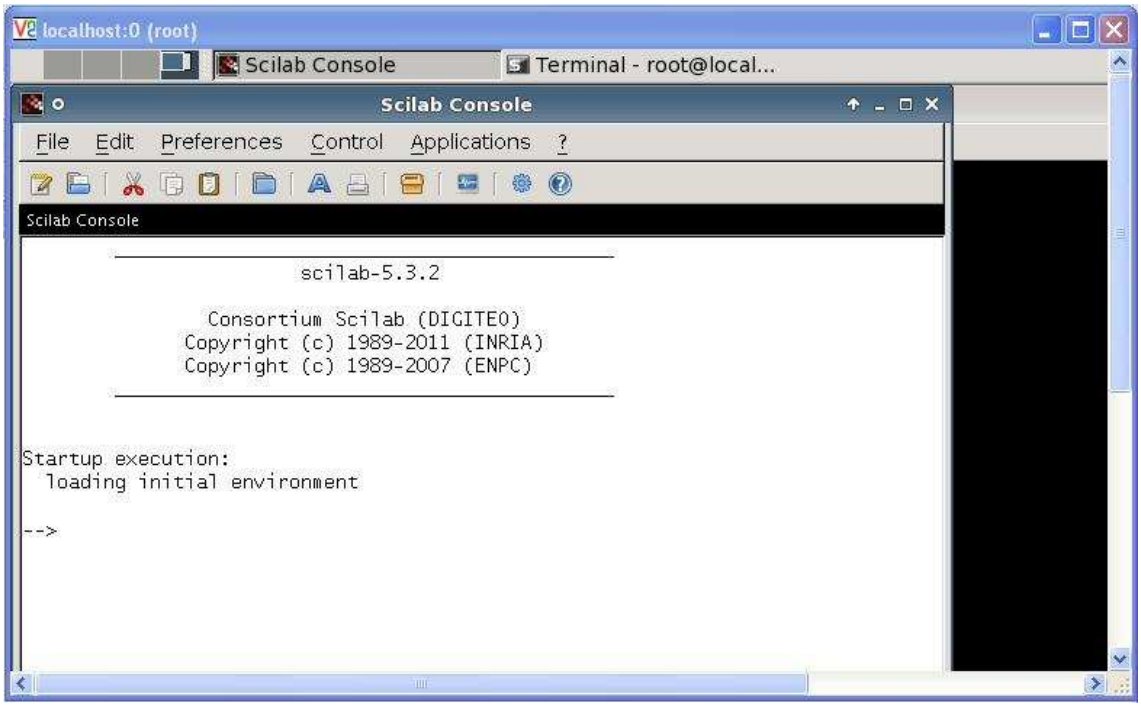
The application is a graphic interface where the user can open instances of virtual machines and execute them. User has to specify his user code and password for being able to connect with the system. There is a list of images and a list of instances. Each image has the name of the program that is installed in. Then easily they can click in access instance and a SSH console will be open, like if it was the same computer. But if the user wants to open with graphic interface, should open with VNC. It is easy to select a key or to select a group of policies.

In this view there are some restrictions to stop some wrong petitions and don't send them to the system.

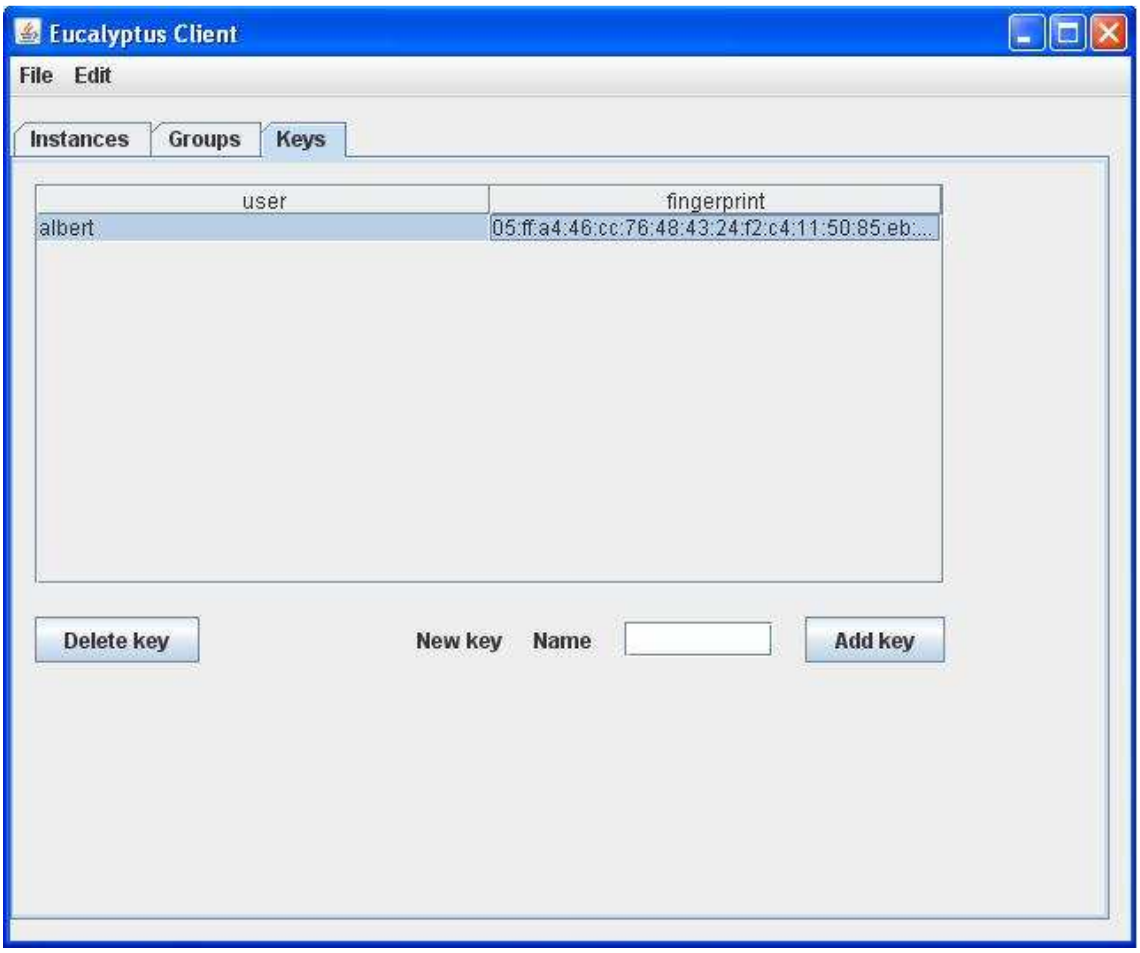
- If there aren't instances running or there's no one selected on the instances table; stop instance, and access instance buttons are disabled.
- If new instance button or stop instance button is selected the two tables are refreshed immediately.
- If there is no image selected new instance button is disabled.
- Refresh button refreshes the two tables, the user selector and the group selector.



If we want it's also posible to open that terminal with VNC in this case we will enter in the machine in graphic mode and we can interact with it and open the program.


localhost:0 (root)
Scilab Console
Terminal - root@local...
Scilab Console
File Edit Preferences Control Applications ?
Scilab Console
scilab-5.3.2
Consortium Scilab (DIGITEO)
Copyright (c) 1989-2011 (INRIA)
Copyright (c) 1989-2007 (ENPC)
Startup execution:
loading initial environment
-->

Management of Keys

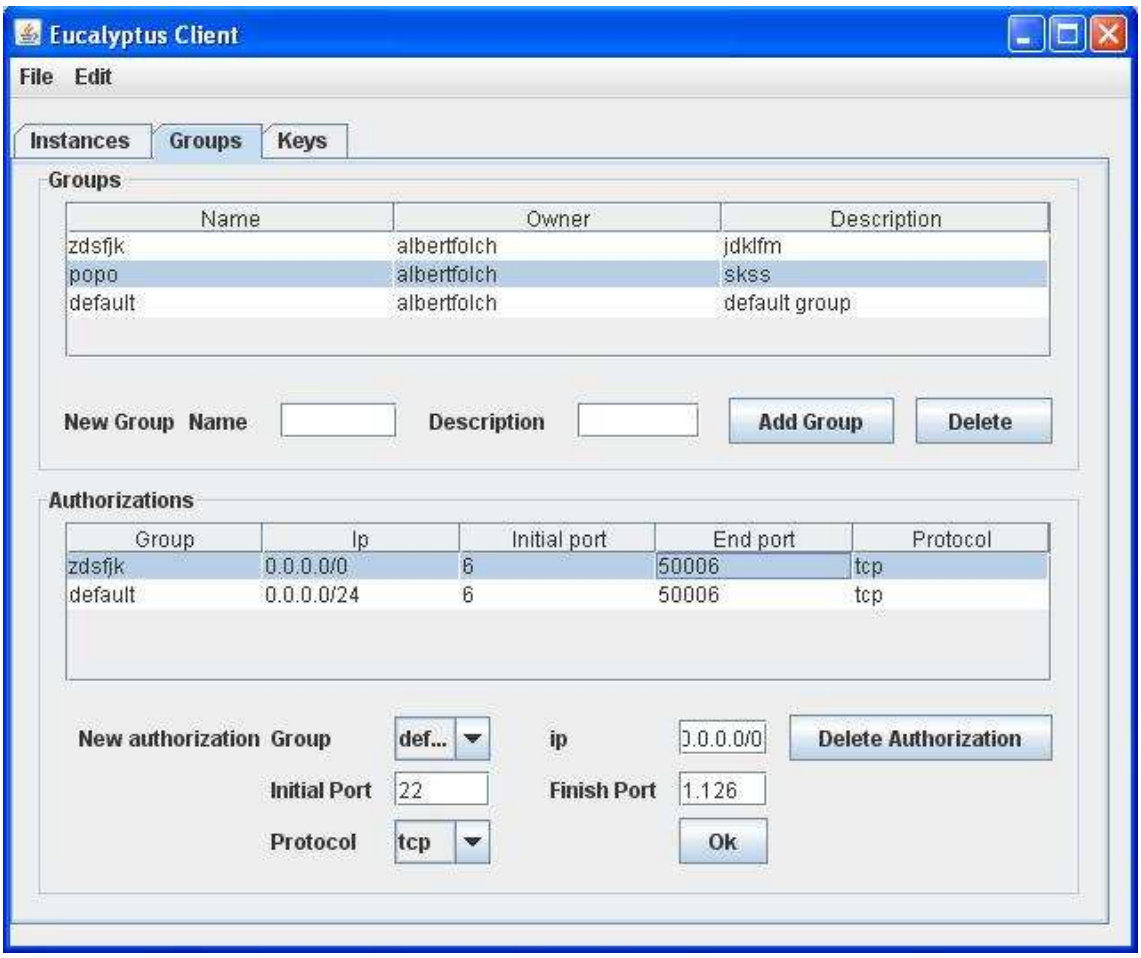


Here we can add keys to the system, if the user want to allow more person to enter in an instance with different keys it is very easy to create new ones, the ssh private key will be automatically saved in one directory by default but can be saved in another one if user specify it in the options. This certified it's personal and necessary for accessing via SSH to the instance.

Here are the restrictions of this view:

If there is no key selected the delete key button its disabled

Management of policies



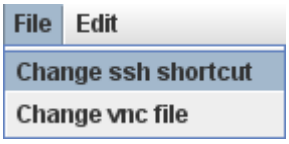
In this other capture we can see the different groups of policies that there are in the system and the policies that are applied for each one of these groups, we have to remember that we can apply to a machine more than one group of policies but for make it more easy to the user this quantity is restricted to one with this program. The user can add and delete easily these policies although is not necessary for him to make it.

The restrictions for this view are:

- If there's no selected group or authorized group the delete buttons are disabled.
- When a group is added or deleted Group selector is actualized and also the two tables.

Options

Also it's necessary to specify where PuttySSH and VNC are located inside the disk, because the program has to know the route for open them.



d. Requirements for the cloud service

Design a friendly user graphic interface for interacting with Eucalyptus

The user will be not familiar with cloud computing so it's important to make as easy as possible the interaction with eucalyptus.

Make feel the user like if he was running the application at home.

The system has to connect with Eucalyptus server with security.

Inform to the user if the connection has been lost

Due problems of connection it can exist sometimes disconnections, make know to user when the connection is lost.

Give necessary tools to the user for adapt Eucalyptus to his needs

Make an advanced menu for advanced users that want to get better performance. (Creating access policies).

Use open-source solutions and follow standards in the deployment.

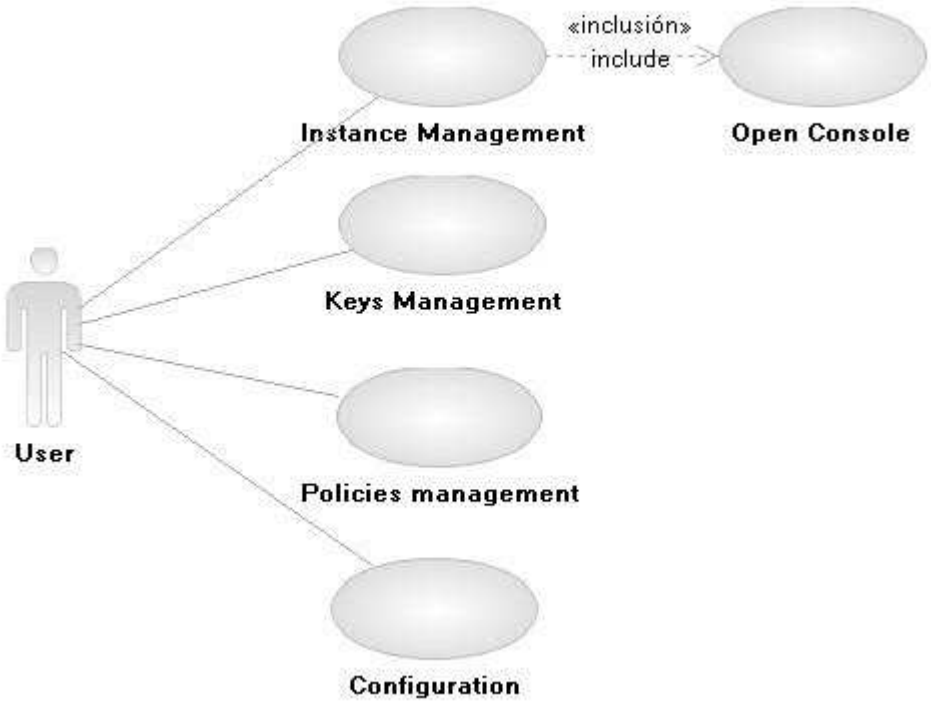
Deploy a solution using 3 layers pattern.

If the view has to be changed it doesn't mean that code of logic has to be changed.

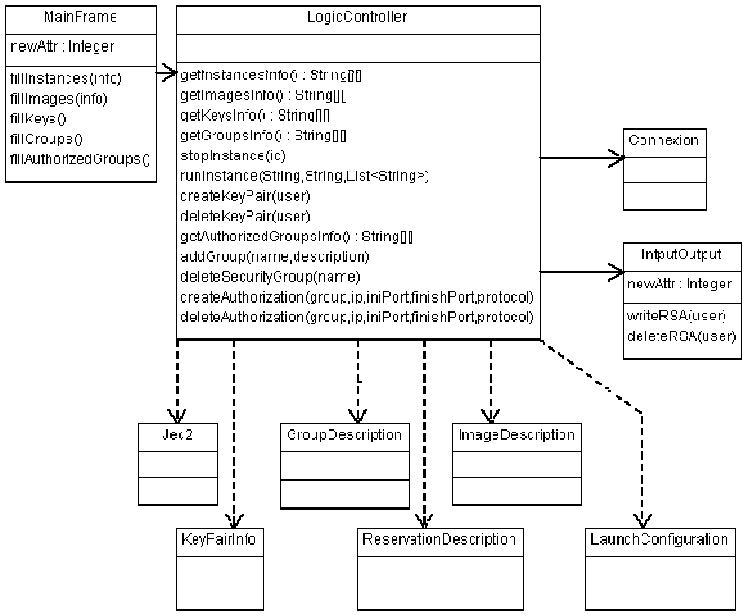
Explain how code is working for make it easier to change or improve in the future.

e. Architecture of cloud service (interface to the Scilab), UML diagrams…

Use case diagram



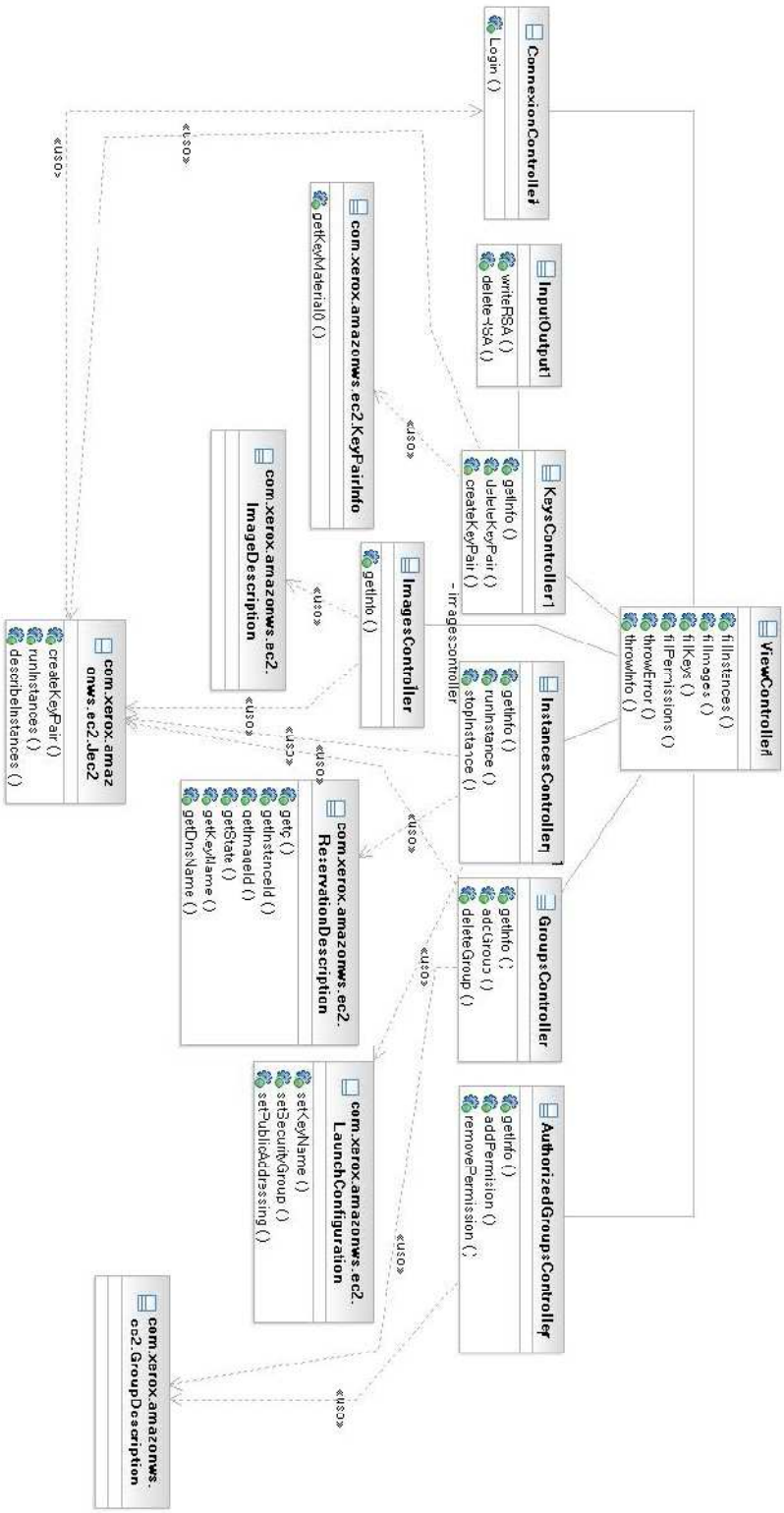
In this project there are clearly differentiate 4 different main use cases for each process of the system. As we see there's only one actor in this system, cause it's a personally program to manage the Eucalyptus system. He can interact with the system getting information about the available images on the system and the instances running on them, user can create and delete keys and specify and manage all the policies of accessing to these instances. Also user can specify the configuration of the program. Also there's a special use case, open console is included inside the instance management.



This was the first class diagram designed for the system, we can differentiate 4 main classes and all of them are singleton, only one object of each one. There's the main frame where is the design of the interface and also the listeners of the events that the user activate with the interaction with the interface. Then these events are calls to functions or actions that are in the LogicController. This way we separate completely the interface with the logic. The view wait for the response of the logic controller, this responses are not in special classes, the information that the view receive is always in arrays or matrix and not in the structures that Typica use, because if one day Typica changes their API will be not necessary do any change on this class.

There is a special class focused on working with the input/output of data called "Input Output". It's the one who have to deal with the registration of private keys in computer.

Also there's the connection class that is the one who deals with the connection with Eucalyptus. And finally we found the classes that Typica API offers and are used.



But after I decided to change this diagram of classes because the main controller was carrying with too much work and i separate it in differents classes, one for the

instances, another for the images, another for the keys, another for the groups and finally another for the authorized groups. This way we can separate all the work and we can get better uncouple between the classes of the system and the ones of typica. This way if something change on typica is easier to change it in the system, and for future changes its more easy to identify where to implement new funcionalities or change them. Also we keep inputOutput class and Connexion one working.

Definition of operations of the system

InstancesController

public String[][] getInstancesInfo() throws EC2Exception

Gets information about the existent Instances and returns it.

returns a matrix containing the information of each image (row) hosted in the system and the next attributes per column: the id of the instance, the name of the image the state the name of the key that owns it and the public ip for accessing

ImagesController

public String[][] getInstancesInfo() throws EC2Exception

Gets information about the existent Instances and returns it.

returns a matrix containing the information of each image (row) hosted in the system and the next attributes per column: the id of the instance, the name of the image the state the name of the key that owns it and the public ip for accessing public void

public void runInstance(String amiName, String user,LinkedList<String> securityId) throws EC2Exception

Runs an instance of a determinate image, with a determinate key like owner and with a determinate security group.

stopInstance(String id)

Stops an instance that is running in that moment on the system with the given id

KeysController

public String[][] getInfoKeys() throws EC2Exception

Gets from the system information about the existent Keys and returns it.

Returns a matrix containing the information of each key (row) in the system with the same information per column: keyName and fingerprint (public key)

public static void createKeyPair(String user) throws EC2Exception

create a key in Eucalyptus and save the RSA in disk

public void deleteKeyPair(String user) throws EC2Exception

Deletes an existing key in the system and form the pool of RSA in disk

GroupsController

public String[][] getInfoGroups() throws EC2Exception

Gets information about the existent groups and returns it.

returns a matrix containing the information of each group (row) hosted in the system with the next attributes per column: name of the group, key owner of the group and description of that group.

public void addGroup(String name, String description) throws EC2Exception

creates a securityGroup in Eucalyptus with a given name and description

public void deleteSecurityGroup(String name) throws EC2Exception{

Deletes an existing securityGroup in the system

AuthorizedGroupsController

String[][] getInfoAuthorizedGroups() throws EC2Exception

Gets information about the existent authorized groups and returns it.

returns a matrix containing the information of each group (row) hosted in the system with the next attributes per column: ip, initial port, final port, protocol

void createAuthorization(String group, String ip, int iniPort, int finishPort, String protocol) throws EC2Exception

creates an authorization in Eucalyptus with the group that contains it, the ip that is applied to, the initial port, the finish port, and the protocol

void deletePermission(String user, String ipAddress, String portIni, String portFin, String protocol) throws EC2Exception

Deletes an existing authorization in the system with a given group that has it, the ip address the initial port, the finish port and the protocol

ViewController

public void fillInstances() throws EC2Exception

private void fillImages() throws EC2Exception

private void fillKeys()throws EC2Exception

private void fillAuthorizedGroups() throws EC2Exception

private void fillGroups(String[][] groups) throws EC2Exception

Each one of these functions asks to one of the controllers about the state of the system and refills one of the informationTable or selector. For example the fillInstances ask to the InstancesController about the instances that are running in the system and it tooks this informatn and refill the table of instances.

public void throwError(String message)

public void throwInfo(String message)

We have also some listeners of buttons that call functions of the Controllers, we can see some of them.

private void instancesTableMouseClicked (java.awt.event.MouseEvent evt)

for example all tables have a listener for when we clicked on them it means they are active and contain some information and a row is selected. In this case we can enable and unable some of the buttons to allow or not the user to click them.

public MainFrame()

The creator of this class call all the fillingTable functions that we saw before, when it finish all the tables are ready and the selectors also with the current information of the system.

All this functions are the ones that are activated when user clicks one of the buttons. Each function is activated by a specific listener. They took information of the view (for example if runInstanceButton is clicked the view will ask to the table of images wich one is selected and will send this information to the InstanceController)

private void stopInstanceButtonActionPerformed(java.awt.event.ActionEvent evt)
private void refreshImagesButtonActionPerformed(java.awt.event.ActionEvent evt)
private void accessInstanceButtonActionPerformed(java.awt.event.ActionEvent evt)
private void addKeyButtonActionPerformed(java.awt.event.ActionEvent evt)
private void deletteKeyButtonActionPerformed(java.awt.event.ActionEvent evt)
private void deleteGroupButtonActionPerformed(java.awt.event.ActionEvent evt)
private void addGroupButtonActionPerformed(java.awt.event.ActionEvent evt)
private void deleteAuthorizedButtonActionPerformed(java.awt.event.ActionEvent evt)
private void newInstanceButtonActionPerformed(java.awt.event.ActionEvent evt)
private void newAuthorizationButtonActionPerformed(java.awt.event.ActionEvent evt)
private void AccessVNCButtonActionPerformed(java.awt.event.ActionEvent evt)

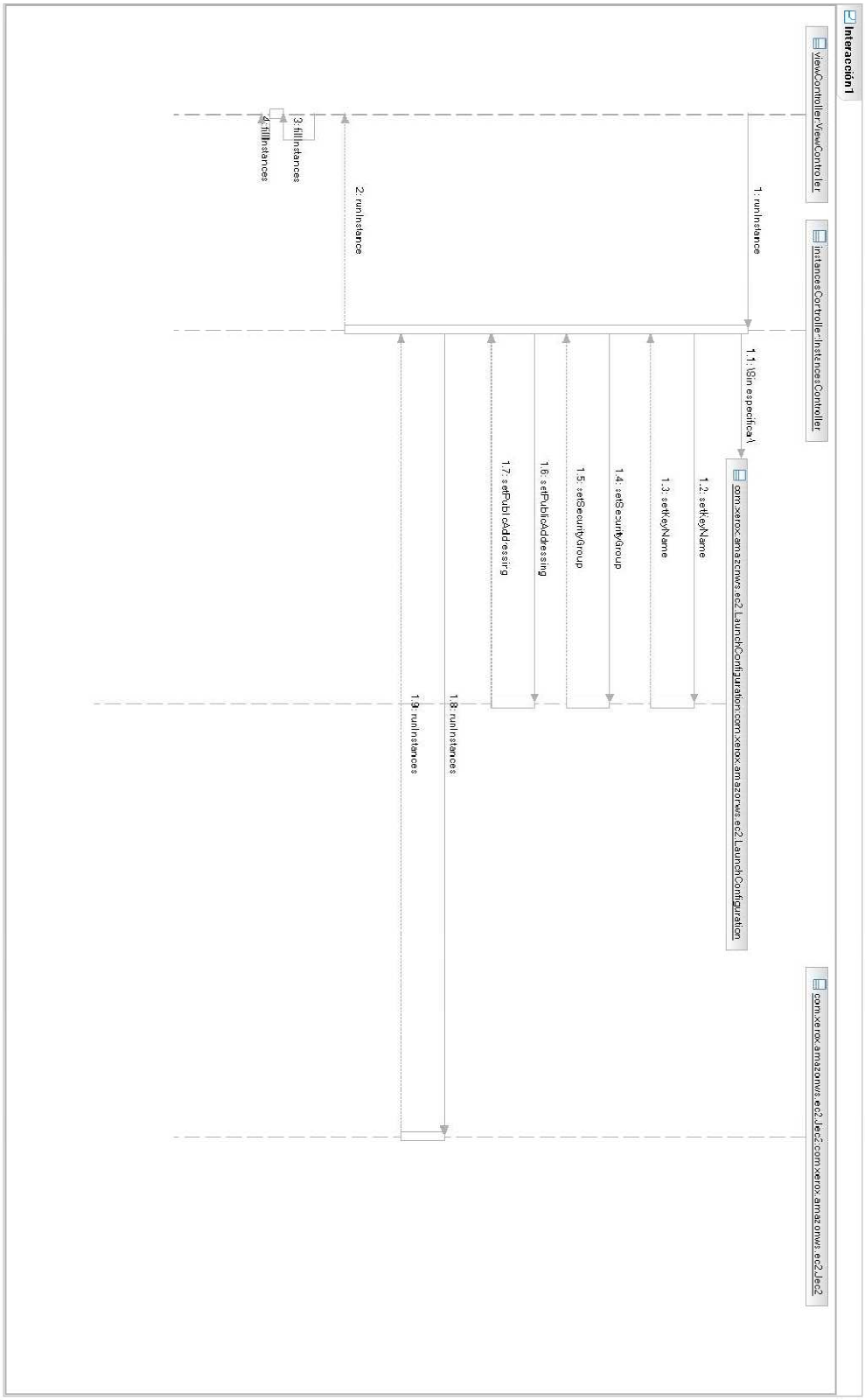
When the instance table is active, user can click buttons like stop instance or access instance, but if this table is not active user shouldn't be able to do it, we control these with actions like this one.

```
private void instancesTableMouseClicked(java.awt.event.MouseEvent evt) {

    stopInstanceButton.setEnabled(true);
```

```
        accessInstanceButton.setEnabled(true);
        AccessVNCButton.setEnabled(true);
}
```

Below there are some seqüence diagrams for understanding the interactuation between the classes of the system

Run Instance



Code of this operation:

Class MainFrame

Button RunInstance has a listener that runs the next function when it's activated:

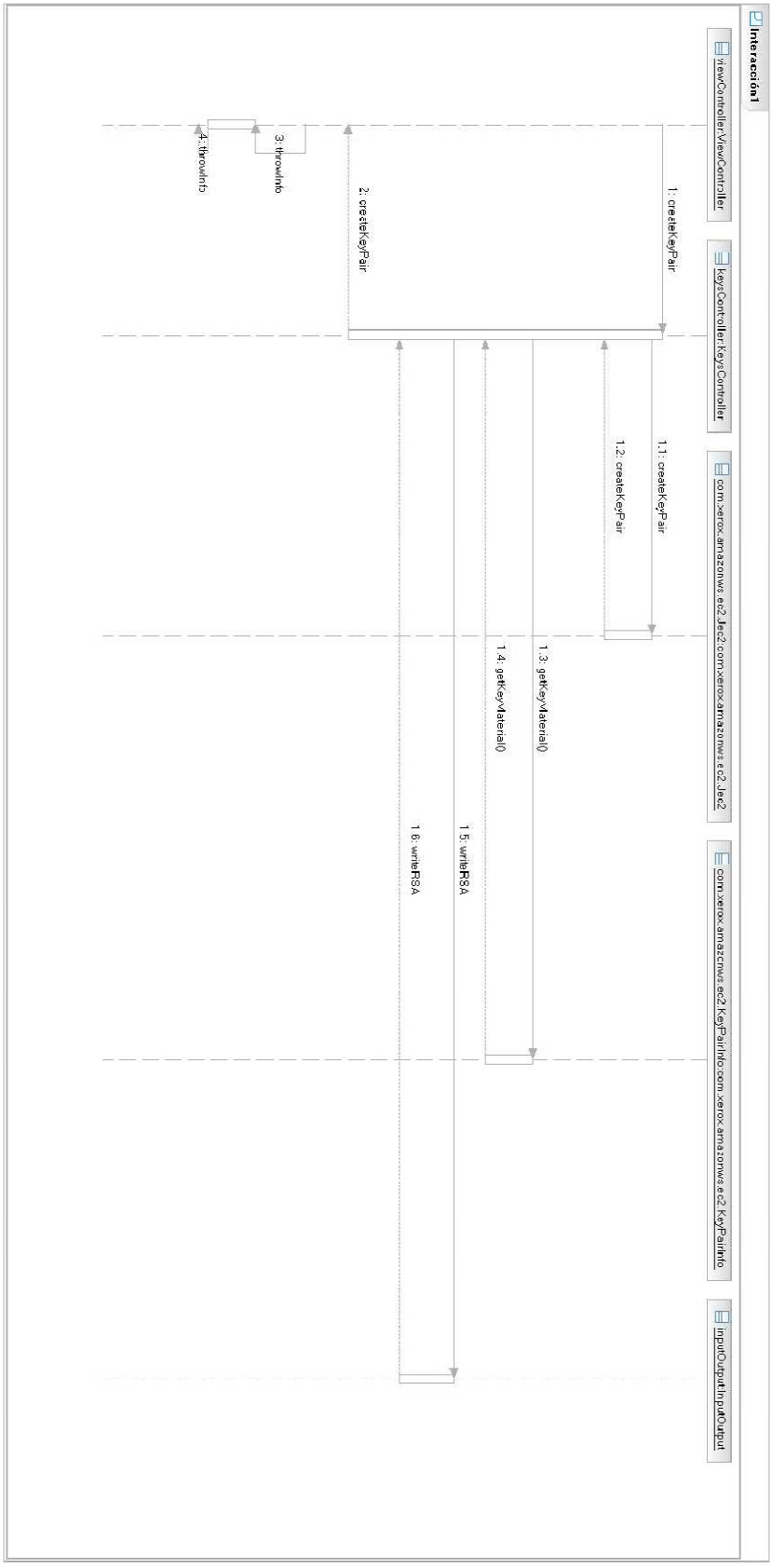
```
private void newInstanceButtonActionPerformed(java.awt.event.ActionEvent evt) {
    LinkedList groups=new LinkedList<String>();
    groups.add(this.groupChooser1.getSelectedItem().toString());
    try {

InstancesController.getInstance().runInstance((String)this.imagesTable.getValueAt(this.images
Table.getSelectedRow(), 0), (String)this.userChooser1.getSelectedItem().toString(), groups);
      this.fillInstances();
      throwInfo("New instance is running correctly");
    } catch (EC2Exception ex) {
      throwError("Error running a new instance");

    }

  }
```

In instanceController

```
public void runInstance(String amiName, String user,LinkedList<String> securityId) throws
EC2Exception{

    LaunchConfiguration lC = new LaunchConfiguration(amiName);
```

```
    lC.setKeyName(user);

    lC.setSecurityGroup(securityId);

    lC.setPublicAddressing(true);

    Connexion.getConnexion().runInstances(lC);

}
```

Create Key Pair



Class MainFrame:

```
private void addKeyButtonActionPerformed(java.awt.event.ActionEvent evt) {
      try {
         KeysController.getInstance().createKeyPair(nameField.getText());
         throwInfo("User added correctly");

      } catch (Exception ex) {
          throwError("Error adding a user");
      }
      try {
         this.fillKeys();
         this.fillImages();
      } catch (EC2Exception ex) {
         Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE, null, ex);
      }
   }
```

Class keysController

```
public static void createKeyPair(String user) throws Exception{
      try {
         KeyPairInfo kp=Connexion.getConnexion().createKeyPair(user);
         InputOutput.writeRSA(user, kp.getKeyMaterial());
```

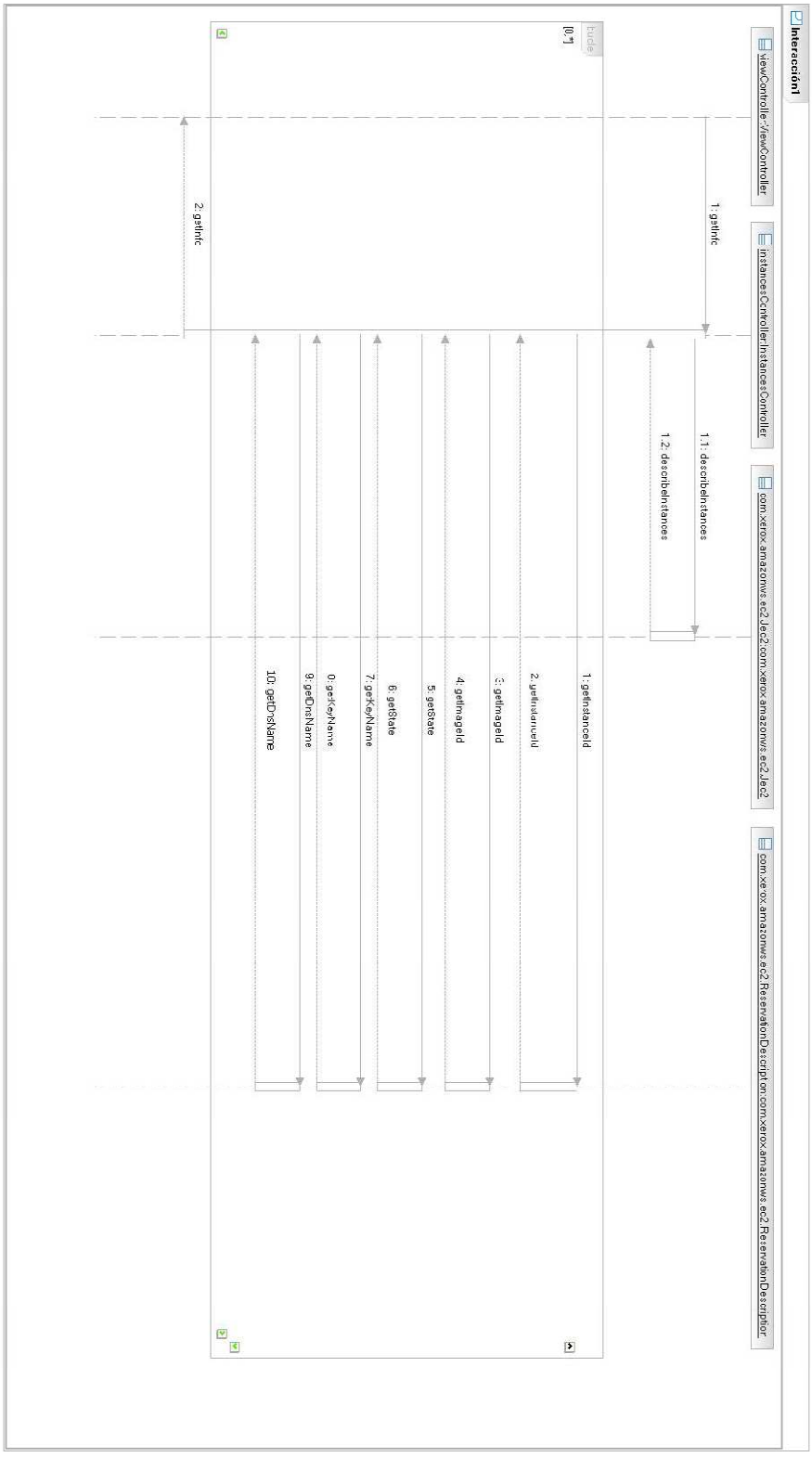
```
    } catch (IOException ex) {
        Connexion.getConnexion().deleteKeyPair(user);
        throw new Exception();
    }
  }
```

Connexion class

```
public static void writeRSA(String user,String material) throws IOException{

    FileWriter fichero = null;
    PrintWriter pw = null;
    try
    {
        fichero = new FileWriter(path+""+user+".txt");
        pw = new PrintWriter(fichero);

        pw.print(material);

    }  finally {

       if (null != fichero)
         fichero.close();
        }
```

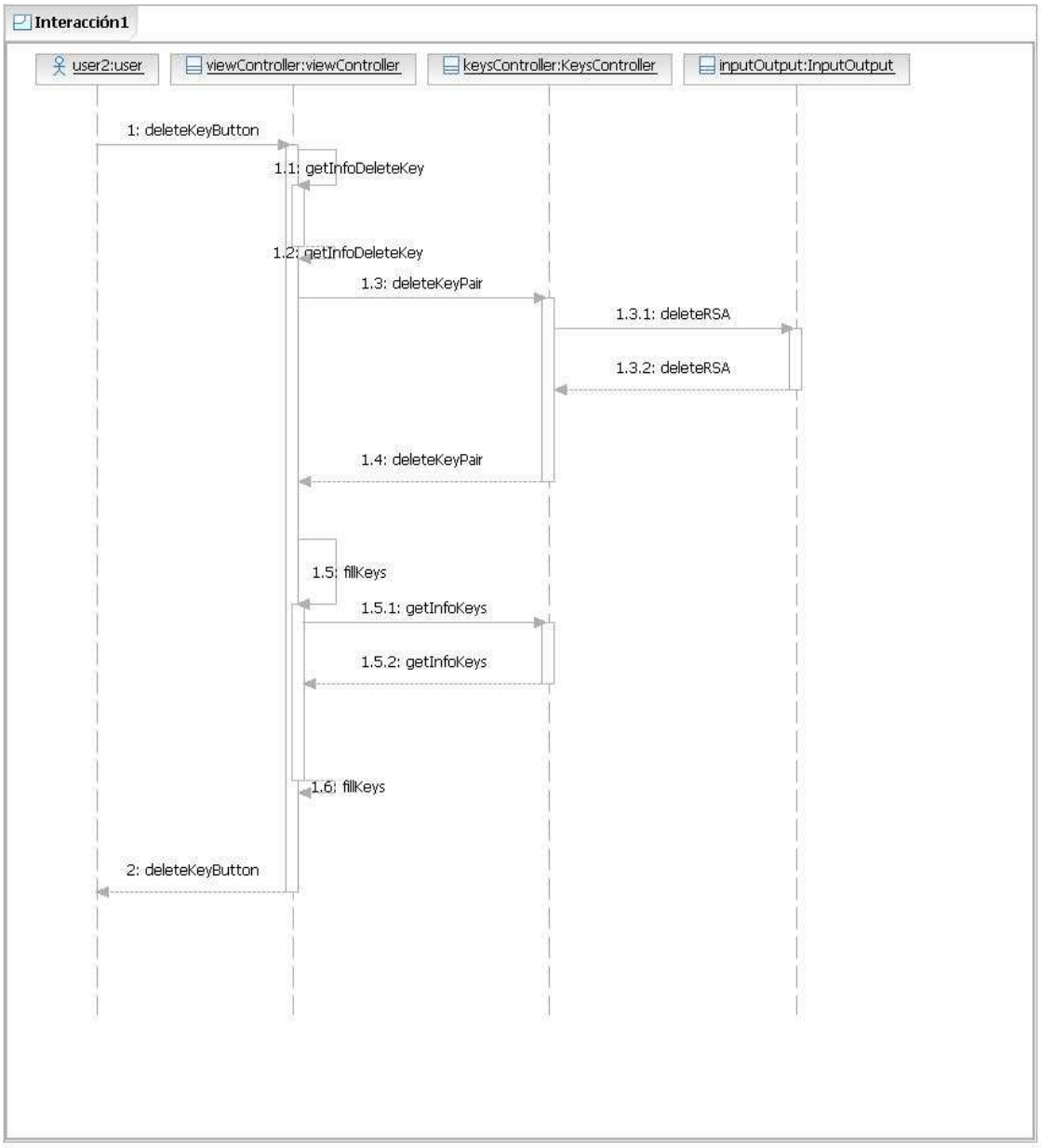
}

Get Interfaces Info



```
Class MainFrame

public String[][] getInstancesInfo() throws EC2Exception{


    String [][]instances=null;

    List<ReservationDescription> reservations =
Connexion.getConnexion().describeInstances(new String[]{});

    int i=0;

    for (Iterator it=reservations.iterator();it.hasNext()&&i==0;){

      ReservationDescription o=(ReservationDescription)it.next();

      instances= new String[o.getInstances().size()][5];

      for (Iterator<Instance> it2=o.getInstances().iterator();it2.hasNext();){

        Instance  iD=it2.next();

        instances[i][0]=iD.getInstanceId();

        instances[i][1]=iD.getImageId();

        instances[i][2]=iD.getState();

        instances[i][3]=iD.getKeyName();

        instances[i][4]=iD.getDnsName();

        i++;

      }

    }

    return instances;

  }
```

Delete Key

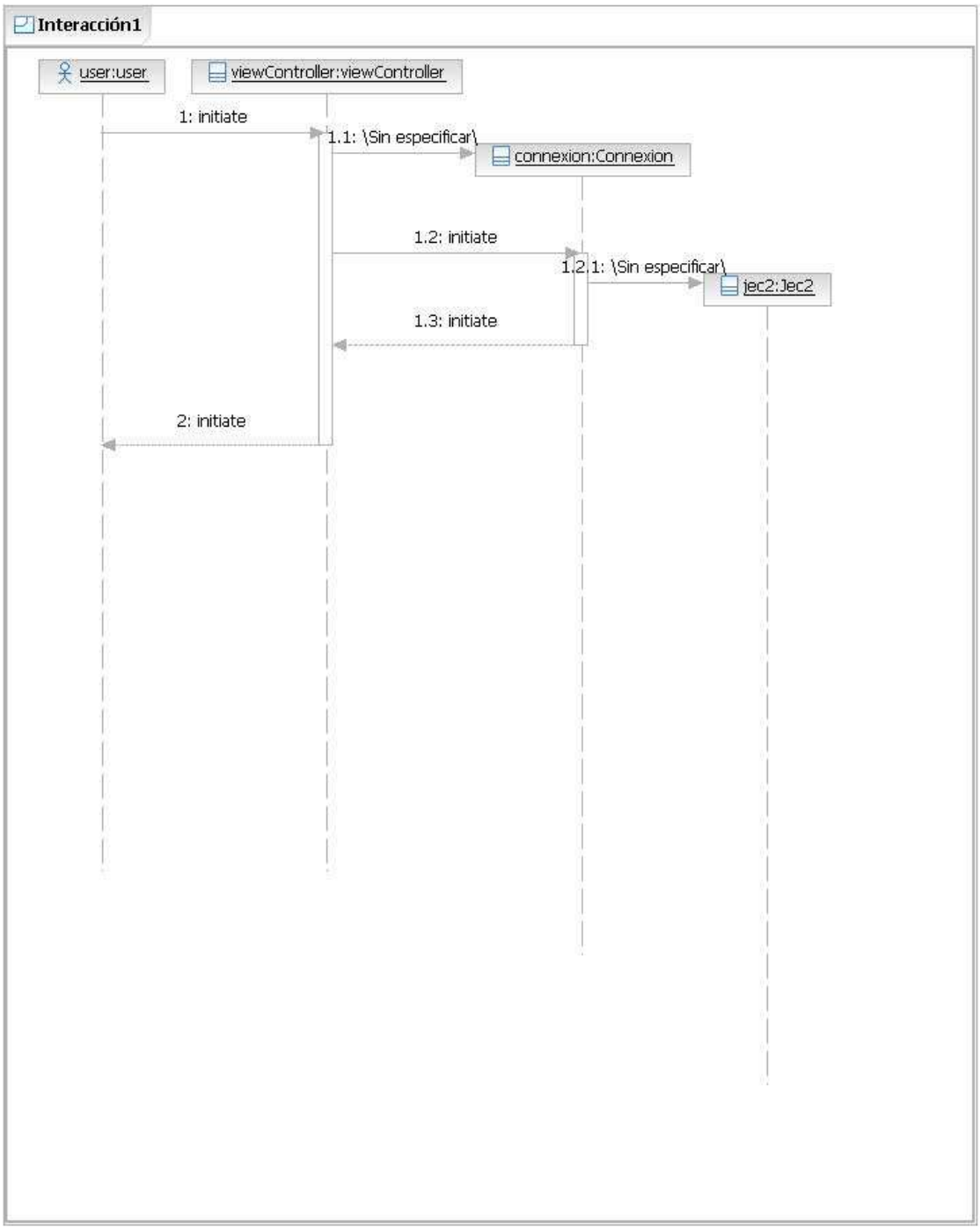

Interacción1
user2:user
viewController:viewController
keysController:KeysController
inputOutput:InputOutput
1: deleteKeyButton
1.1: getInfoDeleteKey
1.2: getInfoDeleteKey
1.3: deleteKeyPair
1.3.1: deleteRSA
1.3.2: deleteRSA
1.4: deleteKeyPair
1.5: fillKeys
1.5.1: getInfoKeys
1.5.2: getInfoKeys
1.6: fillKeys
2: deleteKeyButton

Class MainFrame

```
private void deletteKeyButtonActionPerformed(java.awt.event.ActionEvent evt) {
      try {
         if(keysTable.getSelectedRow()>=0)
KeysController.getInstance().deleteKeyPair((String)keysTable.getValueAt(this.keysTable.getSel
ectedRow(), 0));
      } catch (EC2Exception ex) {
         throwError("Error deleting a user");
      }
      try {
         this.fillKeys();
      } catch (EC2Exception ex) {
         throwError("Error connecting with server");
      }
   }

private void fillKeys() throws EC2Exception {

      this.infoKeys=KeysController.getInstance().getInfoKeys();
     keysTable.setModel(new javax.swing.table.DefaultTableModel(
         infoKeys,
         new String [] {
            "user", "fingerprint"
         }
      ){
         @Override
   public boolean isCellEditable(int rowIndex, int columnIndex) {
      return false;
```

```
    }
});
      if (infoKeys.length==0) this.deletteKeyButton.setEnabled(false);
      else {
          if(keysTable.hasFocus())
          this.deletteKeyButton.setEnabled(true);
      }
    }

Class KeyController
public void deleteKeyPair(String user) throws EC2Exception{

          Connexion.getConnexion().deleteKeyPair(user);
          InputOutput.deleteRSA(user);

    }
Class inputOutput
public static void deleteRSA(String user){

         File f= new File(path+""+user+".txt");
         f.delete();
     }
```

Connexion


Interacción1
user:user
viewController:viewController
1: initiate
1.1: \Sin especificar\
connexion:Connexion
1.2: initiate
1.2.1: \Sin especificar\
jec2:Jec2
1.3: initiate
2: initiate

Conclusions and suggestions for further works

If we look back to the objectives of the work we can differentiate some points, the user friendly interface was easy to design and deploy thanks to the GUI builder of Netbeans that work really well. Also the API JSwing it's very intuitive to use and really well documented. Also there are not many difficulties because Java Swing (now part of Oracle) has very good tutorials online in the same official page and also Netbeans page have some tips and helps for using it. Also programming with Java for this kind of desktop application is always comfortable.

The deployment of the communication with Eucalyptus it was not so easy. At the beginning have been a little bit complicated to understand how Eucalyptus works. Although there is some information the official web page is not really well documented and it's necessary to check the Amazon EC2 documentation to understand some things. Also Typica framework doesn't have a very good documented API so it's necessary to check Amazon documentation a lot of times. Also I found this API not difficult but a little bit uncomfortable to use but just in some punctual moments. Also are some bugs like when an instance is terminated always an exception is thrown. And also the exceptions are not very helpful cause some time there are some errors not specified. But it's also true that the communication with this API is intuitive and becomes easier when you are familiar with it. It's also true that for having such a small cluster sometimes the response of the virtual machines take a long time, it will be good to check why this happens if some day the university wants to use Eucalyptus in some project. Maybe for run an instance we need one minute more or less.

The part of cloud computing in general have been very interesting, it's a world that is becoming bigger and bigger each day and there are a lot of huge organizations (as we saw before) that are betting for it. And only with some years of history, it looks like is going to be a future tendency if things don't change too much, but it's also true that is a little bit scaring that some enterprises can have all information hosted inside and all the computation power. If this are educational institutions everything it will be no

problem, but the fact that some huge organizations can have all this information can be a handicap for this technology.

And checking how powerful can be I made some tests comparing the performance of Scilab in my computer and in the cloud. I tried two examples of calculation. And I calculate this performance just with one instance (we should remember that in cloud it's possible to ask for more than one instance running an image). This test has been done with a Samsung NC10 notebook.

Code	Time host (seconds)	Time cloud (second)

stacksize('max');
IM=zeros(1000,1000);
IM=matrix(IM,1,1000000);
IM = string(IM);
timer();
mputl(IM,"essai.txt");
Time=timer() timer();
	1.984375	0.620038

timer();
mat=rand(1000,1000);
for i=1:1:10
det(mat);
end;
Time=timer()
	19.625	8.072504

These are two random examples, the first one calculates time in writing in disk, and as we see it's almost 4 times faster in the cloud. The second calculates 10 times the determinant of a matrix and the result in cloud is calculated in half time. Although these results are not real significant we can observe that for high computation this works really well, it would be interesting for the future to add more capacity to the cloud and try to execute the same with all the nodes and for sure the results are going to be more specific.

Before doing this thesis I knew a little about cloud computing but basically about SaaS, I didn't know such a quantity of possibilities that this technology offer, it's really interesting that it's possible to build applications on the cloud, including all the phases of the creation of software.

Also have been interesting, but not very easy, to write this entire thesis and defend it in English. It is first time that I have to deal with it and have been a little bit difficult sometimes to search the exact words to describe some things, but finally my valuation is positive.

For further works I would like to go further in this topic, it will be real interesting to try to create my own images and install eucalyptus by my own. I think it can be real interesting also to check other possibilities like Open Nebula or Amazon EC2, or maybe try to develop some applications with Google App Engine.

4. References

- http://www.kurzweilai.net/memorandum-for-members-and-affiliates-of-the-intergalactic-computer-network
- http://www.computerweekly.com/Articles/2009/06/10/235429/A-history-of-cloud-computing.htm
- http://www.servermotion.com/blog/2010/07/history-of-cloud-computing-yesterday-today-and-tomorrow/
- Above the Clouds: A Berkley View of Cloud Computing

Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia (February 10, 2009)

- http://www.datati.es
- http://twitter.com/business_ready
- http://www.cloudsecurityalliance.org/
- http://www.idg.es/idgconnect/Default.aspx
- http://cacm.acm.org/magazines/2010/4/81493-a-view-of-cloud-computing/fulltext
- http://www.saasmania.com
- http://aws.amazon.com/ec2
- http://www.elastichosts.com
- http://www.gogrid.com