

# **Written internship report**

Nicolas SOARES

soares\_n

Tek5 year 2016

## **Company :**

Société Générale, Grid department

## **Duration of the internship :**

6 months

03/03/16 to 03/09/16

## **Subject :**

The goal of the internship is to develop applications and tools to monitor and handle the computing grid used by other departments of Société Générale



# **Part I**

## **The company**

### **Description**

Société Générale is one of the main and oldest French multinational banking and financial services company.

The company group is a universal bank divided into 3 main pillars :

- Retail banking (Société Générale, Crédit du Nord and Boursorama)
- International Banking and Financial Services (IBFS)
- Corporate and Investment Banking (SG CIB)

Those 3 pillars are backed by 2 other core activities :

- Specialised Financial Services & Insurance
- Private Banking, Global Investment Management & Services

### **The department**

You are now part of the RESG/GTS/MKT/SSB/GRD department of Société Générale, each acronym represents a layer in the hierarchy of the group :

- RESG :

The direction of the group's resources and innovations, it gathers the teams in charge of information systems, technical infrastructures, purchases, estate operations and counsel activities of the group

- GTS (Global Technology Service) :

GTS ensures the proper functioning of computer infrastructures and their resilience and counsel and assists the business lines and functional divisions for the implementation of their projects

The computer infrastructures handle the workstations, the computing centers hosting the servers and the computer and telecom networks

- MKT (Market) :  
The wholesale bank division, its goal is to ensure the proper functioning of the computer infrastructures and the client relationship with its business partners SGCIB (Corporate and Investment Banking business of Société Générale) and GIMS (cover the private bank activities, asset management and securities services)  
This division assists its clients for their infrastructures choices, coordinates changes and ensures projects monitoring, service quality and respect of financial commitments.
  
- SSB (Servers, Storage and Backup) :  
This division is in charge of operations on storage et backup of the servers. The team assures the administration, maintenance and backup of all the Market infrastructure (physical and virtual) and handles several projects to ensure the proper functioning of the production
  
- GRD (Grid) :  
The Grid team is in charge of handling and monitoring the grid computing platform and infrastructure for various departments of Société Générale

## **The mission**

Your mission is to develop applications on the Grid computing scope.

## **Grid computing platform**

Grid computing is used for various purposes demanding high compute power including financial risk and pricing computing.

To achieve this high compute power, the grid computing platform is composed of a set of heterogeneous and affordable computing resources, working together to provide clients a unified compute power source.

The computing resources are glued together and handled by middleware programs.

The main middleware software used in the Société Générale Grid is the Symphony platform (IBM), designed for the financial services industry and working with various operating systems.

## **Work environment**

Each project have a script to setup its work environment, it will mount the virtuals X and Y disks, used by the projects so they have the same environment as in production.

The continuous delivery principle is used for the projects with CruiseControl, the success of the build of the project is tested when a new version is pushed to the versioning system (svn or Git). The builds status can be monitored using the CCTray tool.

## **The projects**

You will be working on the Grid monitoring projects, your mission is to continue the development of existing tools and develop new projects or features of projects, mainly using the C# language, according to the needs of the Grid team's clients.

The solution contains several projects and tools, in the first time you will continue the development of the GridAudit and GridSupervisor projects.

### **Overall**

Those projects have several dependencies common to other projects, therefore those dependencies are grouped in external projects :

- GTS-Common-Libs and GTS.GRD contains various helpers, class extensions and configurations
- GTS.GRD.Domain contains classes and configurations used to communicate with the Grid database with the ORM NHibernate
- GTS.GRD.Interfaces contains the interfaces principally used by the IoC Manager (see below) for the services
- GTS.GRD.Services includes several classes managing services (persistence, checks, WMI requests, notifications, compute nodes etc...)

Most of the classes of the projects use the Inversion of Control design principle (IoC), it uses dependency injection to handle the load and configuration of the dependencies of the classes at run time. This design principle allows a more modular approach and prevents tight coupling between classes.

The class IoCManager handles it, it uses the Castle Windsor Inversion of Control container, one of the main IoC framework for .NET

The projects communicate with different databases and API to get informations about the Grid nodes :

- Grid database : A Microsoft SQL Server database
- Marley database : database listing all the elements of configuration SG CIB : servers, nodes, workstations, phones, printer etc...
- HAL API : REST API providing an interface between system administrators and the automation orchestrator performing automated tasks
- ICM API : REST API listing compliance informations about servers
- IKnow API : REST API listing informations on all exceptions affecting security and behavior of servers
- WMI (Windows Management Instrumentation) : Set of specifications which allow to request management tasks on a system

### **GridAudit**

GridAudit is a console program that will apply several tests and requests regarding a given list of server nodes to acquire and store informations about those nodes and detect inconsistencies among the different sources of those informations.

The list of server nodes is given to the program by parameter, either by giving directly a coma separated list of node name with "-Node" or by giving the path to a file containing the list of node with "-File".

The "-FromGridDB" or "-FromMarleyDB" parameters alone can also be used to select all nodes listed respectively in the Grid database or in the Marley database.

The parameter "-NoPostCheck" can be added to disable the execution of the post-checks.:

After parsing the parameters, the program call the Process method of the GridAudit class with a list of node name. For each node a thread is created, each thread will execute a list of check regarding a server node.

The check list is defined in the configuration file and the checks will be executed in the same order as listed in the configuration file.

The current checks are divided in 2 types:

- Parent checks used as conditions for the execution of their nested checks (they will only be executed if the parent has been executed successfully and return a positive value) :
  - o NodeWmiCheck: verify if the node checked can receive WMI requests
  - o NodePingState: verify if the node can be pinged
- Checks retrieving informations about the node and storing/updating those informations in the Grid database to be accessible in the GridHome platform:
  - o NodeExistsInInfos : test if the node is listed in the Grid database or/and in the Marley database
  - o NodeHallInfos : retrieve several informations about the node from the HAL API (server installation date, decommission date, bundle etc...)
  - o NodeHardwareInfos: retrieve several hardware and operating system informations about the node, mainly through WMI requests (Os version/architecture, model, memory informations, uptime etc...)
  - o NodeIcmInfos: get the node compliance ratio from the ICM API
  - o NodeIknowInfos: test if the node is listed in the IKnow API
  - o NodeMarleyInfos: retrieve several hardware and operating system informations about the node from the Marley database
  - o NodeSymphonyInfos: retrieve Symphony informations from the node

Once all the checks have been executed for each node and each thread is killed, a series of post-checks (defined in the configuration file like the previous checks) are executed (except if the "-NoPostCheck" option was passed to the program).

Contrary to the previous checks, the post-checks are not executed for each node, they use directly the full list of nodes.

The post-checks search for inconsistencies of result from the checks :

- CpuRepositoryDatabasePostCheck : list the entries in the table AuditCpuRepository (Grid database) created automatically that should be checked by a human to confirm the number of CPU/cores available
- LocationPostCheck: check location consistency in the AuditNode table of the Grid database
- NoMissingGridNodePostCheck : list all the symphony nodes that should have been listed in the AuditNode table
- RecentUpdatesPostCheck: lists the outdated entries from the AuditNode table

When a post-check found inconsistencies an alert is send by mail (the recipient is defined in the configuration file)

## **GridSupervisor**

GridSupervisor run as a service and execute, in an infinite loop, a list of compliance check on lists of server nodes.

Compliance errors detected are stored in the "GridComplianceAlerts" table of the Grid database to be displayed on the GridHome platform.

Nodes lists are loaded from a configuration file, each list has its own list of check.

The table "ref\_GridCompliance" of the Grid database contains the checks configurations : scope, delay between notifications, actions to execute if an alert is raised, templates to use on the GridHome platform to display the compliances error and informations to resolve those errors.

Each list of nodes passes several filters to remove nodes that cannot be checked (nodes unavailable, in maintenance, unreachable etc...).

Each checks associated with the node list is then executed for each node of the filtered list.

The currents checks are :

- DiskSpace : Using WMI request, verify if the diskspace available on the node is between thresholds defined in the check configuration (in the "ref\_GridCompliance" table)
- EgoHostTypeCheck : verify if the node's OS architecture match the node's host type
- FileSpace : verify that the size of logs and paging does not go over the threshold defined in configuration
- Ping : verify the ping of the node
- ServiceState : through WMI request, verify the state of the node
- SymphonyFileReference : verify various symphony informations, the version in particular

A check can have an action associated that will be executed if it detects an error, those actions are defined with the rest of the check configuration.

Currently the only check with actions is the DiskSpace check, it can either call the CloseNode or OpenNode actions. They close or open a node and list the node as in maintenance or remove it from the maintenance list.





## **PART II**

### **Context**

The project GWiM, Global Windows Master, is an interface used by all the departments of GTS to administrate the deployment of servers, from OS installation and configuration to software installation and execution. The front office part of this project is a web application made in C# but a new project will soon be created, a native mobile version of the front office.

A new developer (Eric Fakename) experienced with GWiM is already assigned for this task but here is an email trying to convince my supervisor (Loïc Smith) to let me take part of the project and assist the new developer.

**To :** loic.smith@socgen.com

**From :** nicolas.soares@socgen.com

**Subject :** Request about the mobile GWiM front office project

Dear sir,

The development of the mobiles GWiM applications is starting soon. The development will take time but, as I perceive from the many mails from various department teams, this project is eagerly awaited.

Eric Fakename, who is already assigned to this project, will be alone to achieve it and, even if he is very experienced with GWiM, has not a lot of experience in mobile development.

So I ask you to let me take part of this project too to help him.

I think my help will be useful, indeed, I already have many experiences in native mobile application thanks to my previous internships and school projects and I think my participation on this project will save precious time.

I also, through the different projects I participated in and the people of many departments I met since the start of this internship, acquired various knowledges about the grid computing platform and infrastructures. And this project will be a great opportunity for me to apply those knowledges and acquire even more.

Working with Eric will allow me to discover more deeply GWiM and I am sure it will be rewarding for both of us and the project.

I have been working with you for 3 months now and I think you are satisfied of my work.

I really want to help and this project is a great opportunity for me.

By allowing me on the project we will save a precious amount of time.

Hoping on a positive return, I wait for your answer.

Yours sincerely,

Nicolas SOARES



## **Part III**

### **Context**

The project GWiM, Global Windows Master, is an interface used by all the departments of SSB to administrate the deployment of servers, from OS installation and configuration to software installation and execution.

My experience with this project was participating in the development of the mobile front office.

GWiM working only with the Windows operating systems, a new project is planned : GLiM (Global Linux Master), an administration platform like GWiM but for Linux operating systems.

The development of this project will involve a lot of contact with the other departments of the SSB department which, in this case, can be considered the clients of the Grid team.

Here is an email destined to the manager of the Grid team (Nahim Doe) trying to convince him to let me be in charge of this project.

**To :** nahim.doe@socgen.com

**From :** nicolas.soares@socgen.com

**Subject :** Request about the GLiM project

Dear sir,

As the GLiM project is planned next month, yet no one is still assigned for it, I propose myself to be in charge of this project.

Having already participated in the development of the mobile front office of GWiM I had to acquire a good understanding of the components of the back office therefore I now know every part of the project which will allow me to be quickly efficient.

As each SSB team will have several particular needs for GLiM that will differ from GWiN, I think the communication with the other departments will be an essential part of this project which put me in a good situation.

Indeed, thanks to the previous projects I participated in and the people I met and exchanged with through those projects, I now have good working relations with coworkers from almost all departments of SSB.

Being in charge of this project is also an incredible opportunity for me, it will make me use skills and knowledges I acquired not only from my current internship but from the beginning of my IT scholarship and will also make me practice my social and communication skills.

It is a great challenge for me that I know I can overcome.

Hoping on a positive return, I wait for your answer.

Yours sincerely,

Nicolas SOARES