

Projet it de gestion des obligations sécurisées cbs

Rapport de stage de fin d’études



Ecole d’ingénieur Polytech Paris UPMC

Spécialité Electronique Informatique – Système Embarqués

*Maitres de stage* : *Tuteur académique* :

M. Damien DENIZART M. Andrea PINNA

M. Amen Allah BHAR

Réalisé par : mlle. thin-hinane younsi

Mars – aout 2016

Entreprise d’accueil : GBIS - SOCIETE GENERALE

Autorisation de diffusion de rapport

Je soussigné, M. Damien DENIZART, représentant de la banque « Société Générale », autorise la stagiaire Mlle. Thin-hinane YOUNSI à reproduire ce rapport comme support électronique et papier en vue d’une diffusion dudit travail universitaire dans la communauté académique et de recherche. Cette diffusion peut notamment avoir lieu par le réseau internet ou par un autre type de réseau comme un intranet universitaire. Dans le respect de la politique de sécurité interne de diffusion des documents de fin d’étude, ce rapport ne contiendra aucune capture d’écran de code ni de noms des clients.

Fait à la Défense, le 22 août 2016

Signature

Tuteur et représentant de l’entreprise :

La stagiaire :

Résumé

Dans le cadre de ma dernière année dans l’école universitaire Polytech Paris UPMC en vue de l’obtention du diplôme d’ingénieur en Électronique Informatique – Systèmes Embarqués, j’ai eu l’opportunité d’effectuer un stage de 6 mois dans le département informatique de la Société Générale (ITEC). Celui-ci gère le maintien et l’évolution du système d’information de l’entité assurant les activités de financement et d’investissement du groupe : GBIS. Pour cela j’ai été intégré à l’équipe ITEC/FCC/OSD/CBS qui assure le maintien et l’évolution de l’application CBS (Covered Bonds System).

Il m’a été confié deux missions pour ce stage. Je devais participer comme les autres développeurs de l’équipe à l’implémentation des demandes d’évolution sur l’application CBS (Covered Bond System) et pour cela monter en compétences dans le domaine technique. Je devais ensuite créer en collaboration avec mon chef technique M. Olivier TERRIEN, de nouveaux serveurs, en environnement de développement pour l’application CBS et en environnement UAT et HOMOLOGATION pour l’application TDA. Cela permet d’optimiser l’intégration continue sur chaque application existante, en n’ayant plus un seul serveur commun pour lancer les tests d’intégration, mais un serveur par application.

Ce rapport de fin d’étude reprend l’ensemble des étapes réalisées au cours du développement de ces projets. Il a vocation à présenter mon travail dans son ensemble, en commençant par présenter le contexte dans lequel je suis intervenue, …, la software factory puis un bilan sur l’expérience que j’ai vécue.

Abstract

As part of my last year in university school Polytech Paris UPMC for obtaining the engineering degree in Computer Electronics - Embedded Systems, I had the opportunity to do a 6 month internship in the IT department of Société Générale (ITEC). This manages the maintenance and development of the information system of the entity providing financing activities and group investment: ga. For that I was part of the team ITEC / FCC / OSD / CBS ensuring maintenance and the evolution of the CBS application (Covered Bonds System).

It was given to me two assignments for this course. I had to participate like other developers team in the implementation of change requests on CBS application (Covered Bond System) and for this up skills in the technical field. Then I had to create my art in collaboration with chef Olivier Terrien, new servers, development environment for CBS implementation and UAT environment APPROVAL for TDA application. This optimizes continuous integration on each existing application, by not having a single common server to run integration tests, but one server per application.

This end of the study report covers all the steps performed during the development of these projects. It aims to present my work as a whole, starting with the context in which I intervened, ..., the software factory and then a report on the experience I had.

// à corriger !!

Remerciements

Au terme de ce travail, je voudrais remercier en premier lieu Mme. ZALEGH Fella, manager de l’équipe en charge de l’application CBS, de m’avoir accueilli dans la SOCIETE GENERALE.

Je tiens à adresser mes remerciements les plus sincères à M. Damien DENIZART mon chef de projet, pour son suivi, son soutien tout au long de mon stage et pour tous le temps qu’il m’a consacré.

Je tiens à exprimer toute ma reconnaissance à mon encadreur M. Amen Allah BHAR, ingénieur développeur à la SOCIETE GENERALE, pour son aide, son soutien, ses conseils précieux, ses critiques constructives, ses explications et remarques pertinentes. Je le remercie pour les moyens mis en œuvre pour que mon expérience soit enrichissante.

Je remercie également M. Olivier TERRIEN pour ses encouragements et ses conseils qui m’ont été très bénéfiques. Je lui suis aussi reconnaissante d’avoir toujours trouvé le temps nécessaire pour aborder les points difficiles et orienter mes recherches.

Je tiens à montrer toute ma gratitude à tous les membres de l’équipe ITEC/FCC/OSD et plus particulièrement tous les membres de l’équipe projet CBS à Bangalore, M. POSHAM Sathyanarayana, Kiran et Amarnath. J’ai particulièrement apprécié leurs gentillesses.

Je remercie M. Andrea PINNA, ainsi que tout le corps enseignant et administratif pour leur encadrement et leur accompagnement.

Enfin, je remercie celles et ceux qui me sont chers. Leurs attentions et encouragements m’ont accompagnée tout au long de ces années. Je suis redevable à mes parents pour leur soutien moral et matériel et leur confiance indéfectible dans mes choix. Enfin, j’ai une pensée toute particulière à mon frère, Makhlouf YOUNSI, qui a toujours été là pour moi et qui me pousse à aller toujours plus loin.

Table des matières

[I. Présentation de l’organisme d’accueil 7](#_Toc459642569)

[a. La Société Générale 7](#_Toc459642570)

[b. GBIS 7](#_Toc459642571)

[c. ITEC et sa politique 8](#_Toc459642572)

[i. ITEC/FCC/OSD 9](#_Toc459642573)

[ii. Interaction avec les autres équipes 9](#_Toc459642574)

[II. Contexte et méthodologie adaptée 9](#_Toc459642575)

[a. L’agilité chez ITEC 9](#_Toc459642576)

[b. Organisation des projets dans l’équipe CBS 11](#_Toc459642577)

[c. Les impacts dans CBS 12](#_Toc459642578)

[III. Mission principale : le projet CBS 13](#_Toc459642579)

[a. Présentation du projet 13](#_Toc459642580)

[b. Organisation du stage 14](#_Toc459642581)

[c. Implémentation des demandes d’évolutions sur l’application CBS 15](#_Toc459642582)

[Dans cette partie, je vais vous présenter les phases du développement des besoins pour l’application CBS. 15](#_Toc459642583)

[Je vais commencer d’abord par présenter la phase de spécification et d’analyse des besoins. Puis, je ferai une description détaillée et la conception de l’application. Enfin, le résultat de l’implémentation que nous avons ainsi que les environnements de développement utilisés. 15](#_Toc459642584)

[i. Analyse et spécification des besoins 15](#_Toc459642585)

[IV. Software factory : Création de slaves Jenkins avec ITASS 16](#_Toc459642586)

[a. Mise au point sur l’existant 16](#_Toc459642587)

[b. Quelles évolutions pour quels impacts ? 16](#_Toc459642588)

[c. Processus de création d’un Slave Jenkins 17](#_Toc459642589)

[i. Rest Client 18](#_Toc459642590)

[ii. Le corps de la requête 20](#_Toc459642591)

[d. Création d’une clef SSH 22](#_Toc459642592)

[e. Création de procédure 22](#_Toc459642593)

[V. Conclusion 24](#_Toc459642594)

[VI. Bilan personnel 25](#_Toc459642595)

1. Présentation de l’organisme d’accueil
   1. La Société Générale

La Société Générale est l’une des principales banques françaises et l’une des plus anciennes. Elle fait partie des trois piliers de l'industrie bancaire française non mutualiste (aussi appelés « les Trois Vieilles ») avec LCL (Crédit lyonnais) et BNP Paribas. Le groupe Société Générale a été créé le 4 mai 1864, il compte 148300 collaborateurs, et est présent dans 76 pays. Son siège social est situé au 29, Boulevard Haussmann, 75009 Paris (France).

La banque est organisée en trois piliers :

* Banque de détail en France (BDD)
* Banque de détail et Services Financiers Internationaux (BHFM)
* Banque de Grande Clientèle et Solutions Investisseurs (GBIS, anciennement SGCIB)

Les valeurs défendues par la Société Générale sont l’esprit d’équipe, le professionnalisme et l’innovation. Au cours de mon stage j’ai pu observer que ces trois principes étaient effectivement mis en œuvre. En effet, le travail d’équipe a été déterminant et toujours très encouragé tout au long de mon expérience dans cette entreprise, avec également des points et partages de connaissance pour mieux connaître le travail des autres ce qui aide à créer un esprit d’entreprise. Le professionnalisme est toujours au cœur du travail des équipes projets, et on y est aidé par les procédures mises en place. Enfin des initiatives ne cessent de se mettre en place pour améliorer la production en plus des gros projets innovants comme le « continous delivery ».

* 1. GBIS

GBIS (Global Banking Investment Solutions) est né suite à la volonté de l’entreprise d’harmoniser ses solutions d’investissements. Il regroupe toutes les activités de banque d’investissement, banque privée, gestion d’actifs et de services aux investisseurs. Il occupe une position de premier plan à l’international, avec une présence dans plus d’une cinquantaine de pays pour plus de 20 000 collaborateurs.

GBIS est un des pôles les plus importants de la Société Générale, et lance plusieurs travaux de transformation digitale. Parmi ceux-ci, le principal est le déploiement continu (Continuous Delivery) qui est appliqué à l’heure actuelle à une trentaine d’applications; C’est pour cela que l’entité s’est récemment dotée d’un Cloud privé, afin de se rapprocher des infrastructures Cloud d’acteurs majeurs tels qu’Amazon ou Google. L’enjeu final serait d’avoir des applications web totalement « Responsive Design » (donc utilisables sur de nombreuses plateformes), basées sur un framework maison, « EasyWeb »

L’organigramme se trouve en annexe, mais je vais rapidement présenter les départements avec lesquels j’ai plus particulièrement travaillé. ITEC/FCC, la structure à laquelle j’ai été intégrée, gère tout l’applicatif pour les activités de financement et de couverture clients. Ses utilisateurs sont donc des collaborateurs chez GLFI (le front office), OPER (middle et back office) et FIND pour la comptabilité.

* 1. ITEC et sa politique

Au cours de ces dernières années, ITEC (Information Technology) est devenu l’un des départements phares de la Société Générale. Bien qu’il ne soit pas un service métier, il est d’une importance capitale au sein de la banque.

Il assure tous les services informatiques au sein de GBIS. Il a pour missions principales de garantir une bonne mise en production/œuvre des infrastructures et applications, et de faciliter d’un point de vue technologique tous les axes stratégiques et la croissance de l’entreprise.

ITEC a une grande particularité : il est vecteur permanent d’innovation. Beaucoup de projets sont lancés en parallèle, et visent à améliorer les performances côté métier. L’informatisation des processus et activités étant fortement ancrée dans le milieu bancaire, chaque décision, chaque innovation est faite en tenant compte des ressources disponibles chez ITEC.

ITEC est divisé en plusieurs départements :

• PRD\*, qui s’occupe du support à grande échelle pour tout ITEC

• RRF\* qui est en charge des risques et financements associés

• FCC, traitant essentiellement du financement et de la couverture client

• CSY\*, COO\* et une multitude d’autres départements.

Chaque département d’ITEC s’occupe d’une ou plusieurs branches métiers. Le but est de ne pas disperser la connaissance, tout en permettant une certaine souplesse. Ce découpage entre aspects métiers et aspects IT fait la force du département : ainsi les développements, maintenances, supports et analyses des besoins du business s’exécutent plus rapidement.

Par ailleurs, depuis quelques années, les méthodes Agiles sont une composante importante dans le développement d’ITEC, toujours en accord avec la stratégie globale fixée par GBIS. Ensuite, chaque équipe est libre d’adapter ces méthodes en fonction des spécificités des tâches attribuées.

J’ai effectué ces six mois de stage intégrée à l’équipe ITEC/FCC/OSD. Pour pouvoir comprendre mon expérience, il faut donc tout d’abord savoir la place qu’occupe ce service dans GBIS ainsi que ses missions principales.

* + 1. ITEC/FCC/OSD

Au sein de FCC (Financial and Client Coverage), il subsiste plusieurs services : LTL\*, RPP\*, OSD, PRS et d’autres correspondant à la direction et aux implantations à l’étranger.

Il y a trois services projets LTL (Loan Servicing, Trade Finance and Leasing), RPP (Risk, Provisions and Profitability) et OSD (Origination, Structuring and Distribution), qui s’occupent de créer et développer des applications pour une branche métier, et un service de support applicatif et technique (PRS).

Figure 1 : Organisation de ITEC/FCC

L’équipe OSD (Origination, Structuring and Distribution) gère donc le support fonctionnel pour les applications de FCC/OSD. Elle est divisée en deux périmètres :

* Preclosing (PCL) qui s’occupe des applications liées au Preclosing (Tout ce qui concerne l’avant signature des deals avec les clients)
* Origination (ORI) tout ce qui concerne la titrisation et la syndication
  + 1. Interaction avec les autres équipes

Les clients d’ITEC/FCC sont en fait les utilisateurs des applications et donc des collaborateurs de GBIS.

Les autres équipes avec lesquelles des points réguliers et du travail en commun est mis en œuvre sont l’équipe projet Bangalore (travaillant sur CBS), le support applicatif et les AOP (support technique) qu’ils soient à Paris ou à Bangalore(Inde).

1. Contexte et méthodologie adaptée
   1. L’agilité chez ITEC

ITEC a fait des méthodes Agiles l’un des points les plus importants de son développement à partir de 2011. En effet une nouvelle approche a été définie en ne définissant plus ses concurrents comme les systèmes d’information des autres banques mais les géants du web comme Google, Amazon ou Facebook. Le but est d’assurer un service extrêmement adapté aux besoins de l’utilisateur, de bonne qualité et rapide. Agile répond parfaitement à cette attente.

A mon arrivée j’ai pu assister à une formation sur l’Agilité assurée par mon chef de projet. Pour l’accompagnement de ce changement de méthode en effet, ITEC a constitué une équipe de spécialistes qui ne travaillent que sur le sujet, il y a ainsi un coach pour chaque projet Agile. Cette formation expliquait les bénéfices, contraintes, méthodes et outils conseillés. Cela m’a été très utile étant donné que j’ai pu avoir un premier aperçu de l’adaptation des principes de l’Agilité que j’avais vus en cours au contexte particulier de la Société Générale.

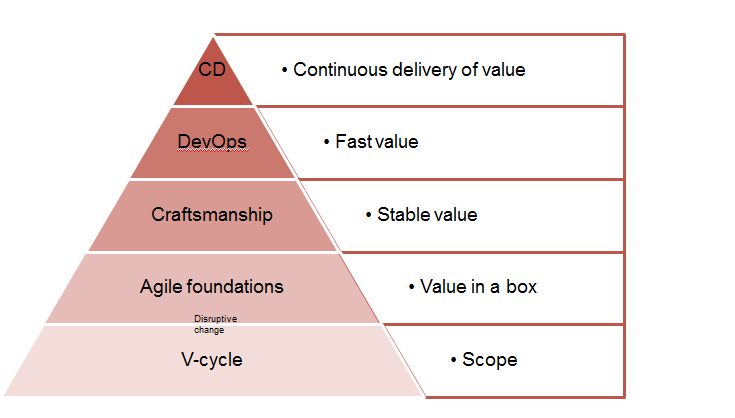


Figure 2 : les étapes de l’agilité chez ITEC

Les centre Agile chez ITEC a défini plusieurs étapes pour passer du cycle en V au Continuous Delivery qui consiste à pouvoir livrer en continu sans action de l’utilisateur et sans diminuer la qualité.

1. **Agile foundation** : première implantation des méthodes Agiles avec le travail par itérations, implication des responsables métiers le plus tôt possible dans le processus pour comprendre le besoin intelligemment, faire attention à apporter de la valeur métier

Pour cela on a inventé le BDD (Behaviour driven developpement) qui consiste à construire les spécifications ensemble avec un developpeur, un Business Analyst, un testeur et le client. De cette façon on s’assure de bien comprendre le besoin et on éduque l’utilisateur à être précis.

1. **Craftsmanship** : rendre le code plus robuste avec automatisation des tests.

La Société Générale utilise une méthode TDD (Test Driven Developpement), c'est-à-dire que les tests unitaires sont écrits avant le développement. De plus certains outils permettent d’automatiser tout ou partie des tests.

1. **DevOps** : gagner du temps, accélérer les livraisons, aller au rythme du client et ne plus être un frein. C’est là qu’interviennent les équipes de support applicatif et technique.

Cela peut être atteint avec des « Non event releases », donc des livraisons de petites fonctionnalités, plutôt que grosses releases de type « Big Bang ».

1. **Continuous Delivery**: être capable d’effectuer des releases en continu sur toute la chaîne d’applications qui participent au business process.
   1. Organisation des projets dans l’équipe CBS

Dans le cadre du projet de CBS développement logiciel, le client élabore sa vision du produit à réaliser et liste les fonctionnalités ou les exigences de ce dernier. Il soumet cette liste à l’équipe de développement, communique directement avec elle (plutôt que par papier) qui estime le coût de chaque élément de la liste. Le projet manager peut ainsi se faire une idée approximative du budget global.

L’équipe sélectionne ensuite une portion des exigences à réaliser dans une portion de temps courte appelée itération. Chaque itération inclut des travaux de conception, de spécification fonctionnelle et technique quand c’est nécessaire, de développement et de test. A la fin de chacune de ces itérations, le produit partiel mais utilisable et montré au client. Ce dernier peut alors se rendre compte par lui-même très tôt du travail réalisé, de l’alignement sur le besoin.



Figure 3 : La méthodologie Scrum

Cette figure décrit une méthode de gestion de projet appliquée au cours du développement et de la livraison des demandes des clients.

* Les éléments « Product Backlog » :

Objectif : liste de fonctionnalités attendues du produit.

Contenu : Les fonctionnalités listées sous forme de « User Stories ».

Rôles : Sous la responsabilité du Product Owner et Scrum Master qui se chargent de classer les éléments par priorité et déterminent l’ordre de réalisation.

Utilisation : Elaboré avant le lancement des sprints et utiliser pour planifier une release.

A tout moment le Backlog est visible dans l’outil JIRA par l’ensemble de l’équipe. Pour suivre l’avancement des différents projets de l’équipe, le chef de projet anime chaque jour des réunions téléphonique avec nos collègues de Bangalore. Ces réunions ont pour but de décrire pour chacun d’entre nous l’avancement, les réalisations, les difficultés rencontrées et la mise à jour du statut des items dans Jira. Ces réunions correspondent au « Daily scrum meeting ». Elles ont lieu tous les jours à la même heure et durent environ 15 minutes.

* Les éléments « Sprint planning » :

Correspondent à une sélection des fonctionnalités du Backlog à développer pendant le sprint.

Le sprint planning a lieu à chaque début de sprint. Tout le monde est présent à cette réunion. Durant cette réunion le scrum master sélectionne des fonctionnalités à développer pour le prochain sprint en fonction de la priorisation des fonctionnalités et la capacité de production de l’équipe.

* Un sprint

Un sprint est une itération, c’est-à-dire une période au bout de laquelle est livré un incrément de produit. Et chaque sprint est composé d’une liste de fonctionnalités à réaliser.

* Une release

Une release est une somme de sprint. A la fin d’une release, on passe en production.

* 1. Les impacts dans CBS

La longueur des sprints est de 2 semaines, ce qui permet d’ajouter des demandes de haute priorité sans enlever quoi que ce soit à partir du backlog de sprint déjà traité.

L’application CBS s’appuie essentiellement sur l’approche Kanban qui consiste globalement à visualiser le Workflow (Le processus de traitement d’une tâche). La personnalisation du tableau Kanban Jira est accessible par toute l’équipe et par le Product Owner (PO).



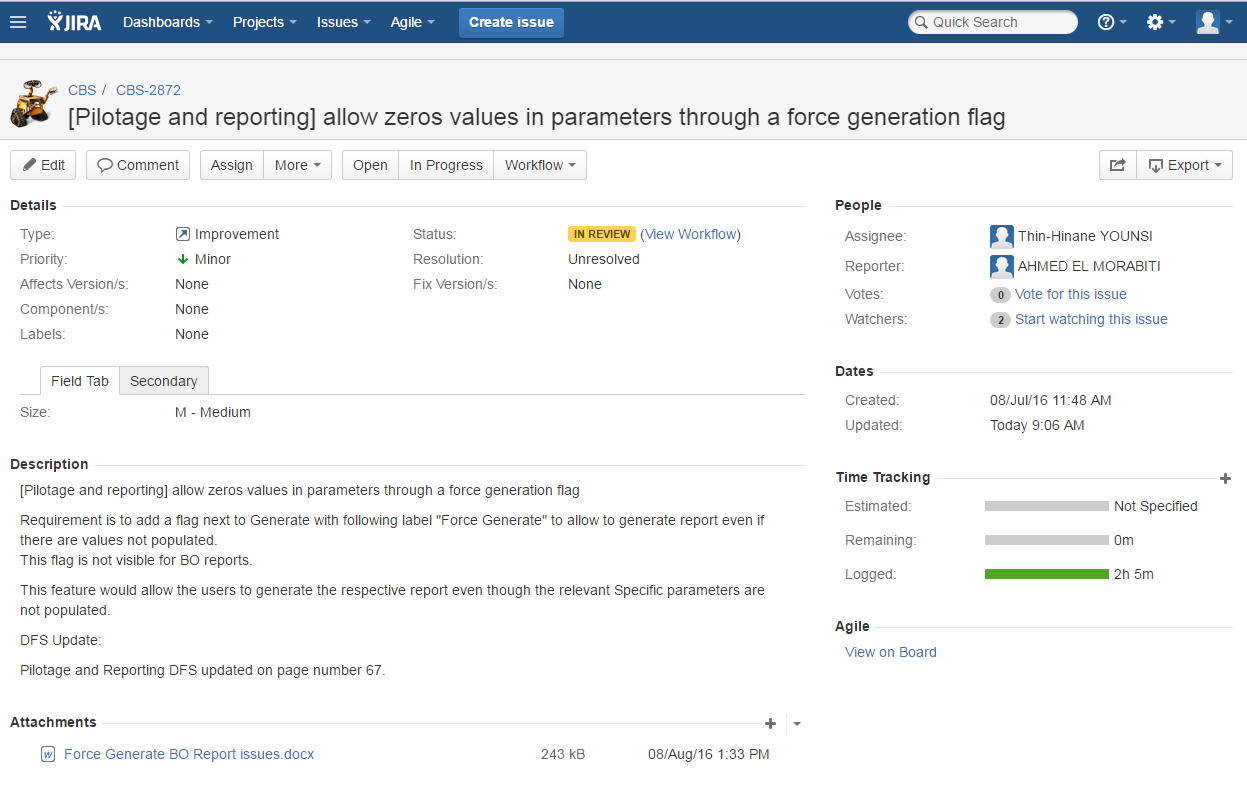


Figure 4 : Aperçu du logiciel JIRA

1. Mission principale : le projet CBS
   1. Présentation du projet

CBS (Covered Bonds System) s’occupe de l’émission d’obligations couvertes. Le principe est d’émettre des obligations à partir d’autres entités que la Société Générale. Celles-ci reçoivent des prêts en collatéral de la SG, ce qui leur permet de couvrir intégralement le risque et donc de pouvoir émettre des obligations notées AAA (ce qui est plus compétitif que si la SG émettait elle-même directement). Les entités peuvent ensuite rembourser la SG avec l’argent récupéré des investisseurs.

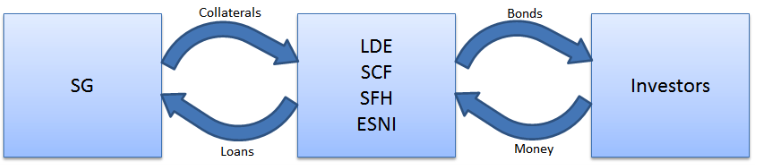


Figure 5 : Principe d’émission d’obligations couvertes

***Pr*o*blématique***

Les Covered Bonds sont des obligations garanties par un portefeuille dédié d’actifs sous-jacents, destinés à garantir aux porteurs obligataires une sécurité supérieure à celle d’un titre obligataire classique. Depuis la crise un intérêt particulier est mis sur les émissions de cette nature, qui jouent désormais un rôle très significatif dans le paysage obligataire européen.

***Solution***

Pour répondre à cette problématique, le département ITEC/FCC/OSD a crée l’application CBS qui permet de veiller à ce que toutes les obligations émises soient couvertes par des actifs. Pour cela deux équipes ont été mis en place l’une à Bangalore et l’autre à Paris.

* 1. Organisation du stage

Dès mon arrivée j’ai été totalement intégré à l’équipe CBS pour pouvoir répondre à la problématique. J’ai donc participé à toutes les activités d’un développeur informatique.

L’une des trois valeurs affichées par la Société Générale est l’esprit d’équipe, et j’ai trouvé tout au long de mon stage que c’était effectivement le cas. De fait, tout est mis en œuvre pour favoriser cet esprit, tant par les différents points permettant de suivre l’avancée des autres que par le travail en groupe ou les événements extérieurs à tous les niveaux.

Le suivie de mes missions se fait principalement dans un mode agile de type Kanban.

Je participe aux corrections des bugs et à la mise en place de nouveau besoins (Defects et backlog). Par exemple :

* En apportant des enrichissements et des améliorations à des modules déjà existants
* Résoudre des problèmes de régression liés à des défauts, défaillances et des erreurs issues d’une mauvaise spécification ou d’une inappropriée implémentation du code
* Mettre en place/maintenir des tests unitaires (TDD : Test Driven Development)
* Développer des interfaces Front End dédiées aux utilisateurs de l’application
  1. Implémentation des demandes d’évolutions sur l’application CBS

Dans cette partie, je vais vous présenter les phases du développement des besoins pour l’application CBS.

Je vais commencer d’abord par présenter la phase de spécification et d’analyse des besoins. Puis, je ferai une description détaillée de la conception de l’application. Enfin, le résultat de l’implémentation que nous avons ainsi que les environnements de développement utilisés.

1. Analyse et spécification des besoins

Dans le but d’une meilleure compréhension des objectifs de l’application CBS, je vais énoncer les besoins fonctionnels et non fonctionnels de celle-ci.

*Besoins fonctionnels :*

CBS possède cinq modules fonctionnels :

* Authentification
* Structure : permet la gestion de l’ensemble des instruments (prêts, échanges, obligation)
* Pilotage and Reporting : permet à l’utilisateur de générer tous les rapports dans l’application
* Referentials : gère les structures, les taux indexés et les référentiels tiers
* Administration : gère les droits d’utilisateurs

Les acteurs de l’application CBS sont :

* Les utilisateurs
* Membres de la SG
* Nombre par emplacement et par entité : 20 maximums à Paris
* Pas d’utilisateurs internationaux ni externe à la SG (le projet n’est pas ouvert sur internet)
* Possèdent soit les droits de lecture, soit les droits de lecture et d’écriture. Ils peuvent Consulter les prêts, importer des garanties que suite à ça il peut consulter l’indicateur de qualité relatif, gérer tout type d’instruments (prêt, échange ou bien des obligations) , gérer toute structure, gérer les taux indexés ,contribuer aussi à la gestion d’un référentiel tiers et générer des rapports.
* Administrateurs : avec un double accès à l’application, ils gèrent les utilisateurs et possèdent les droits d’écriture et de lecture des données de l’application
* L’équipe Support : qui assure le service et maintient de l’application

*Besoins non-fonctionnels :*

Ces besoins apportent un bon fonctionnement de l’application. Nous avons :

* La sécurité
* Etablissement de la connexion selon les règles d’autorisation
* Demande de mot de passe et d’identifiant avant chaque connexion
* Performance
* Respect du temps de réponse : le chargement de l’application, ouverture d’écran et de délais de rafraîchissement
* Performance en temps de traitements (fonctions, calculs, importations/exportation de données…

Ci-dessus, un diagramme d’un cas d’utilisation de l’application CBS :

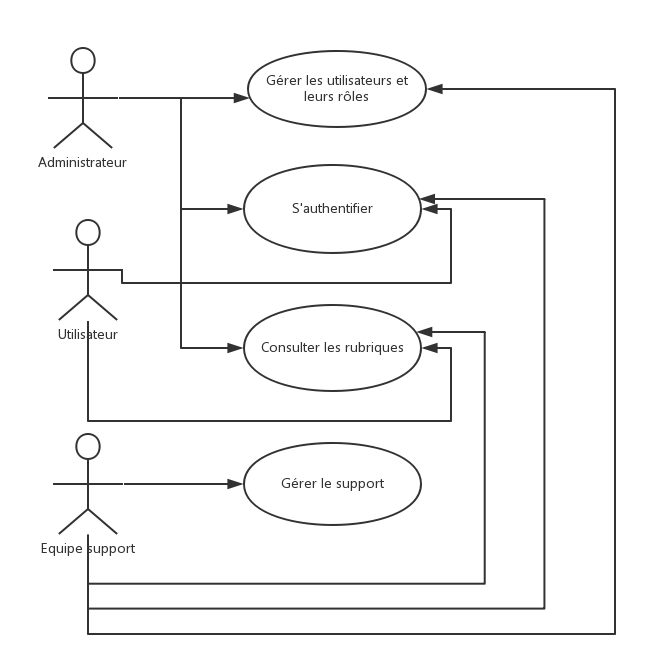


Figure 6 : Digramme des cas d’utilisation général

1. Software factory : Création de slaves Jenkins avec ITASS
   1. Mise au point sur l’existant

Au sein d’ITEC, l’entité FCC/OSD/SCM gère les projets d’évolution ainsi que la maintenance de toutes les applications (TDA, CBS, B3S, SLA Web, ANTALIS) couvrant ces secteurs.

L’intégration continue est un ensemble de bonnes pratiques utilisé dans l’entité FCC/OSD/SCM afin d’améliorer la qualité du code et le produit final. En effet, à chaque modification du code source, on vérifie que le résultat des modifications ne produit pas de régression dans les applications. Le but principal est de détecter les problèmes d’intégration lors du développement. De plus, cela permet d’automatiser les exécutions des suites de tests, de voir l’évolution du développement du logiciel et d’améliorer la qualité du code source.

Cette intégration repose sur la mise en place d’une brique logicielle, qui permet l’automatisation de certaines taches telle que la compilation, le lancement des tests unitaires et fonctionnels, clean code et DB refresh. L’exécution de cet ensemble de tâches se fait à chaque changement dans le code. Le code source est un code partagé issue d’un répertoire Git (dans la configuration des jobs Jenkins, nous passons l’url du répertoire ainsi que la branche à compiler). Ainsi, à chaque fois qu’on commit nos modifications, les taches sont lancées automatiquement. En ce qui concerne les tests d’intégrations, on utilise JUnit pour valider les applications. L’outil d’intégration utilisé est Jenkins (fork de hudson).

Cependant, les applications de l’entité FCC/OSD/SCM se partageaient le même slave Jenkins (fcc-srvparosdp01-nt) pour les lancements des jobs Jenkins, en environnement de développement. Ce slave correspond à un serveur Windows composé de quartes cœurs.

Ainsi, lancer plusieurs jobs en même temps sur ce slave prenait du temps et l’ensemble des jobs ne se lançaient pas en même temps.

* 1. Quelles évolutions pour quels impacts ?

Le choix a tout d’abord été de créer un slave Jenkins pour chaque application existante. Ainsi, chaque application possédera son propre serveur pour pouvoir lancer ses propres jobs.

Cela permettra de gagner du temps dans la résolution des demandes d’exécution des différentes tâches sur toutes les applications. Le panel d’applications gérées étant particulièrement important, il est nécessaire d’avoir un slave Jenkins pour chaque application.

Pour cela on a utilisé ITAAS (IT As A Service), qui offre un panel de services comme nous pouvons le voir sur ce schéma :

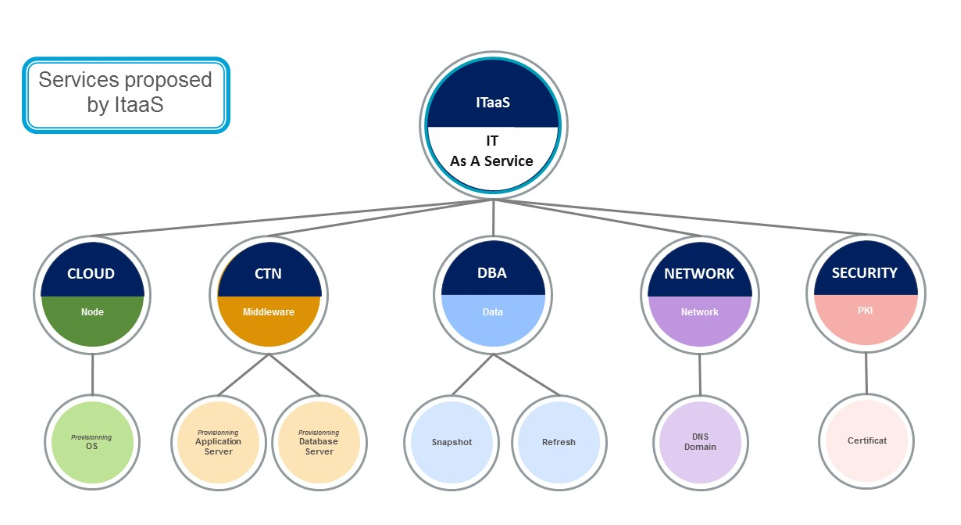


Figure X :

Le service qu’on utilise est le Cloud, qui nous permet de créer nos serveurs linux, qui sont ensuite utilisés comme slave Jenkins.

// Mon travail TDA et CBS

Trois serveurs linux ont été crées. Un serveur pour l’environnement de développement pour CBS, un serveur pour l’environnement UAT, ainsi que sur l’environnement HOM pour TDA.

**Prod Linux VM**

**ʷä Prod**

**Tomcat 7**

**Homol Linux VM**

**ʷä Homol**

**Tomcat 7**

**ʷä UAT**

**Tomcat 7**

**#è Prod**

**Business DB**

**#è Homol**

**Business DB**

**#è UAT**

**Business DB**

**#è Dev**

**Business DB**



**Developer**

//Page 39 pour la figure et explication des environnements

Pour la création des Slave nous avons du travailler en collaboration avec l’équipe GTS et COC.

//description des équipes

* 1. Processus de création d’un Slave Jenkins

Avec ITaas on crée un nœud linux en choisissant :

* Taille : Medium -2vCPU avec 2GB de RAM
* Environnement : Dev

De plus, on fournit une liste d’email. Des notifications sont, ainsi, envoyées par la suite des évènements suivant :

* Lorsque le nœud a été provisionné
* Lorsque le nœud se rapproche de l’expiration. Plusieurs e-mails sont envoyés à l’approche de la date d’expiration :
* 7 jours avant
* 5 jours avant
* 2 jours avant
* 1 jour avant
* Quand il y a moins de 24 heure restantes

Ensuite, on écrit un script Json dans lequel on spécifie tous les paramètres, ce qui correspond à faire une «*request* ».

* + 1. Rest Client

Tout d’abord, je configure Rest Client. [www.restclient.net](http://www.restclient.net) est un plug-in Firefox, testé avec succès avec l’API ITaas.

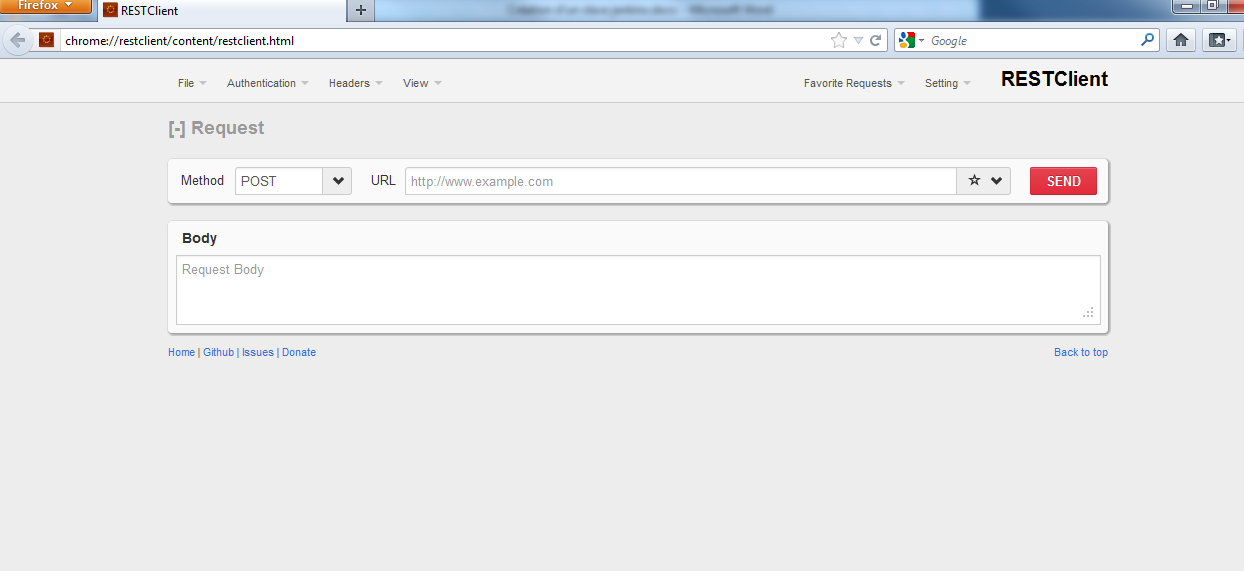


Figure X :

La première étape correspond à l’authentification :

* Dans le menu « Authentication », on sélectionne « Basic Authentication »
* Le login et mot de passe Windows sont utilisés
* La case « Remember me » peut être cochée aussi
* Lorsqu’on clique sur « OK », l’authentification apparait dans la section Headers

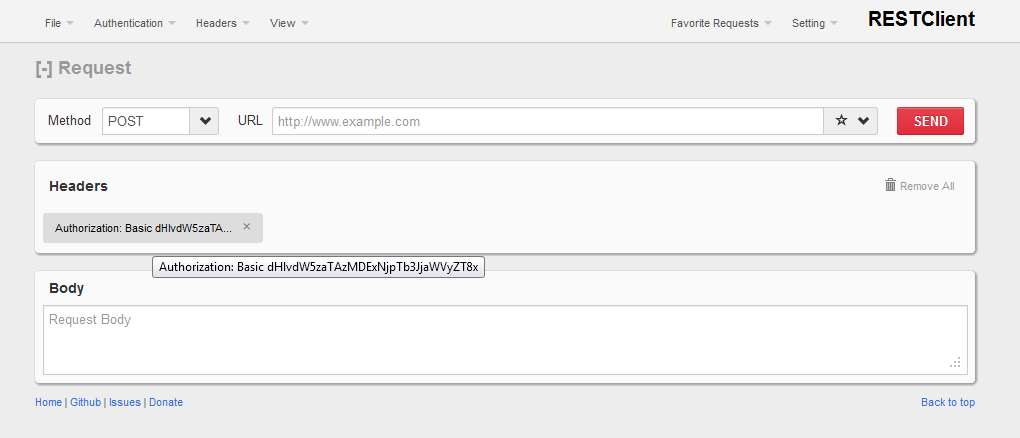


Figure X :

La deuxième étape correspond à la configuration des en-têtes :

* Dans le menu « Headers », on choisit « custum Header »
* Le premier en-tête à ajouter est :
* Nom -> Accept
* Valeur -> application/json
* Le second en-tête à ajouter est :
* Nom -> Content-Type
* Valeur -> application/json
* On coche « Save to favorite »
* En cliquant sur « OK », les en-têtes apparaissent dans la section « Headers »

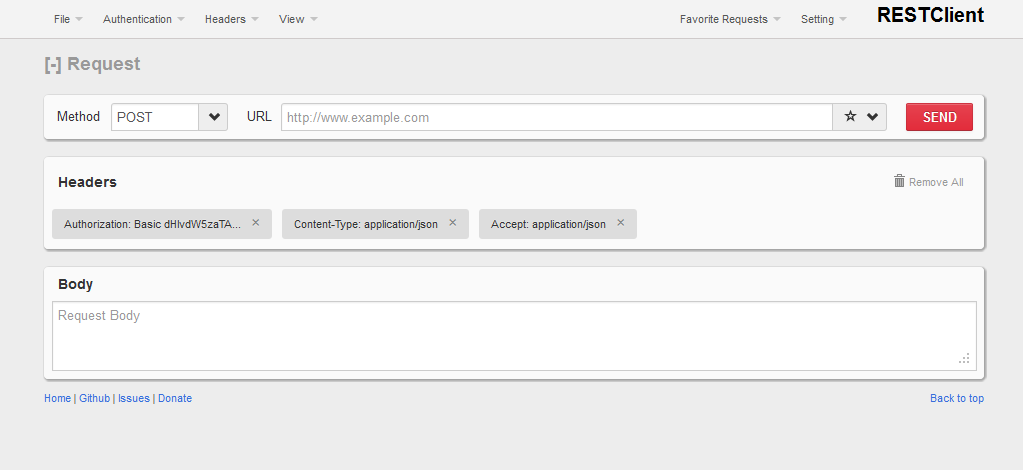


Figure X :

A la troisième étape, on choisit la méthode d’envoi du formulaire, la méthode POST sera utilisée dans notre cas.

Pour finir, dans la section « Body » on écrit notre requête, un script Json où sont définis tous les paramètres de l’API.

* + 1. Le corps de la requête

Dans cette partie, je vais expliquer toutes les étapes pour configurer tous les paramètres d’un serveur linux.

La documentation de l’API fournit un exemple de script :



Pour configurer le serveur linux dans le Cloud pour CBS, nous avions dû fixer certains paramètres :

* Size : ”M”
* Disk size : “60”, pour les dossiers cotenants maven/git/java …
* Lease days : “-1” afin d’avoir un contrat infini
* environment type : “dev”
* Centrify zone : “zd\_fr\_cbs\_01” crée par GTS group
* Description : Jenkins salve for CBS
* User group : “cbsdev” crée GTS group
* service account : “cbsdev02”
  + On fait la demande de création sur la zone CBS, en spécifiant l’utilisateur "cbsdev02" et le groupe associé (cbsdev). Ensuite, une fois que l’utilisateur est crée, s\_fr\_\_cbsdev02 est ajouté dans le groupe AD de CBS.

En effet, il y’a un groupe AD Windows pour avoir le droit de créer des VM dans le Cloud (SGLOUD) qui est le même pour les actions sur l’API REST de l’ITaas.

* app trigram : « CBS »
* region : eu-fr-paris
* availability zone : eu-fr-paris-1
* subnet : 192.160.64.0/21
* notifymail : [olivier.terrien@sgcib.com, thin-hinane.younsi@sgcib.com](mailto:olivier.terrien@sgcib.com,%20thin-hinane.younsi@sgcib.com)
* master\_os : RHEL 6.7 x64 MKT
* label : slave\_jenkins\_linux\_cbs

Enfin, on lance la procedure dans le Client Rest, avec l’URL:

<https://itaas-prod.fr.world.socgen:10443/v2/nodes>

Nous obtenons la réponse ci-dessus, qui nous renvoie un « id » et «creation\_job\_id ». Ceux-ci nous permettent de lancer une requête GET pour avoir le nom d’hôte du serveur.  Le nom d’hôte sera utilisé dans DeployIt.



* 1. Création d’une clef SSH

Une fois que le serveur est provisionné dans le Cloud, on génère une clé publique et une clé privée.

La clé privée est déployée sur le master.

La clé publique est déployée sur le slave dans « /home/<user>/.ssh/authorized\_keys

Et les dossiers possèdent ces permissions :

* ~/.shh est en chmod = 700
* authorized\_keys en chmod = 600
  1. Création de procédure

Le suivi du job pour créer les serveurs se fait en exécutant la requête GET avec l’Id\_creation\_job.

Lorsque le statut du job passe à «succeeded», la VM est prête.

Puis, pour la configuration de la machine virtuelle, c'est-à-dire le dépôt de la clé publique dans le bon répertoire, c’est l'équipe du COC qui s’en occupe.

Exemple :

Pour vérifier l’état du job : GET [https://itaas-prod.fr.world.socgen:10443/v2/jobs/{id}](https://itaas-prod.fr.world.socgen:10443/v2/jobs/%7bid%7d)

TDA UAT :

{"id":"222807","status":"running","date\_started":"2016-05-27T08:24:11Z","date\_ended":**null**,"job\_name":"Create VM Linux","dry\_run":"no","node\_id":8893,"middleware\_id":4136}

TDA Hom:

{"id":"222809","status":"running","date\_started":"2016-05-27T08:36:21Z","date\_ended":**null**,"job\_name":"Create VM Linux","dry\_run":"no","node\_id":8894,"middleware\_id":4137}

Quand les jobs sont créés :

* TDA UAT :

{"id":"222807","status":"succeeded","date\_started":"2016-05-27T08:24:11Z","date\_ended":"2016-05-27T08:57:40Z","job\_name":"Create VM Linux","dry\_run":"no","node\_id":8893,"middleware\_id":4136}

* TDA Hom:

{"id":"222809","status":"succeeded","date\_started":"2016-05-27T08:36:21Z","date\_ended":"2016-05-27T09:09:56Z","job\_name":"Create VM Linux","dry\_run":"no","node\_id":8894,"middleware\_id":4137}

1. Conclusion
2. Bilan personnel

Bibliographie