



PROJET IT DE GESTION DES OBLIGATIONS SECURISEES CBS (COVERED BOND SYSTEM)

Rapport de stage de fin d'études



Ecole d'ingénieur Polytech Paris UPMC

Spécialité Electronique Informatique – Système Embarqués

Maitres de stage :

M. Amen Allah BHAR

M. Damien DENIZART

Tuteur académique :

M. Andrea PINNA

REALISE PAR : MLLE. THIN-HINANE YOUNSI

MARS – AOUT 2016

ENTREPRISE D'ACCUEIL : SGCIB - 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 nom des clients.

Fait à la Défense, le 26 août 2016

Signatures

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 Investement banking (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 intégrée à l'équipe ITEC/FCC/OSD/SCM/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 :

1. Je devais participer comme les autres développeurs de l'équipe à l'implémentation des demandes d'évolution sur l'application CBS et pour cela monter en compétences dans le domaine technique.
2. Je devais ensuite créer en collaboration avec mon leader technique (Tech Lead) 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 cloud 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, les évolutions du projet CBS, la software factory puis un bilan sur l'expérience que j'ai vécue.



Abstract

As part of my last year in Polytech Paris UPMC for validating my engineering degree in Computer Electronics - Embedded Systems, I had the opportunity to join the ITEC department of Société Générale for an internship of 6 months. ITEC manages the maintenance and the development of information systems for entities which are financing the group activities and other investments activities:

During my internship, my main missions were:

- Assisting the team to ensure the maintenance and the evolution of the business applications CBS (Covered Bonds System).
- Participating to the creation of new servers and environments (DEV and UAT).

The missions above allowed me to learn the best practices when we use the continuous integration on existing applications.

This probationary report covers all the tasks performed during my internship. It aims to present my work as a whole.



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. J'ai particulièrement apprécié sa gentillesse.

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, Makhoul 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	6
A.	La Société Générale	6
B.	GBIS	6
C.	ITEC et sa politique	7
a.	ITEC/FCC/OSD	7
b.	Interaction avec les autres équipes	9
II.	Contexte et méthodologie adaptée	9
A.	L'agilité chez ITEC	9
B.	Organisation des projets dans l'équipe CBS	11
a.	Méthode Scrum	11
C.	Les impacts dans CBS	14
III.	Mission principale : le projet CBS	15
A.	Présentation du projet	15
B.	Organisation du stage	16
C.	Implémentation des demandes d'évolutions sur l'application CBS	16
a.	Analyse et spécification des besoins	16
b.	Conception	21
c.	Mes activités dans l'application CBS	26
d.	Outils technologique	29
IV.	Software factory : Création de slaves Jenkins avec ITAAS	32
A.	Mise au point sur l'existant	32
B.	Quelles évolutions pour quels impacts ?	33
C.	Processus de création d'un Slave Jenkins	34
i.	Rest Client	34
ii.	Le corps de la requête	37
iii.	Création d'une clef SSH	38
iv.	Création de procédure	38
V.	Conclusion	40
	Bibliography	42



I. Présentation de l'organisme d'accueil

A. 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 des 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 l'existant en plus des gros projets innovants comme le « continuous delivery ».

B. 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 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.

C. 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

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 équipes.

J'ai effectué ces six mois de stage intégrée à l'équipe ITEC/FCC/OSD/SCM/CBS. 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.

a. 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).

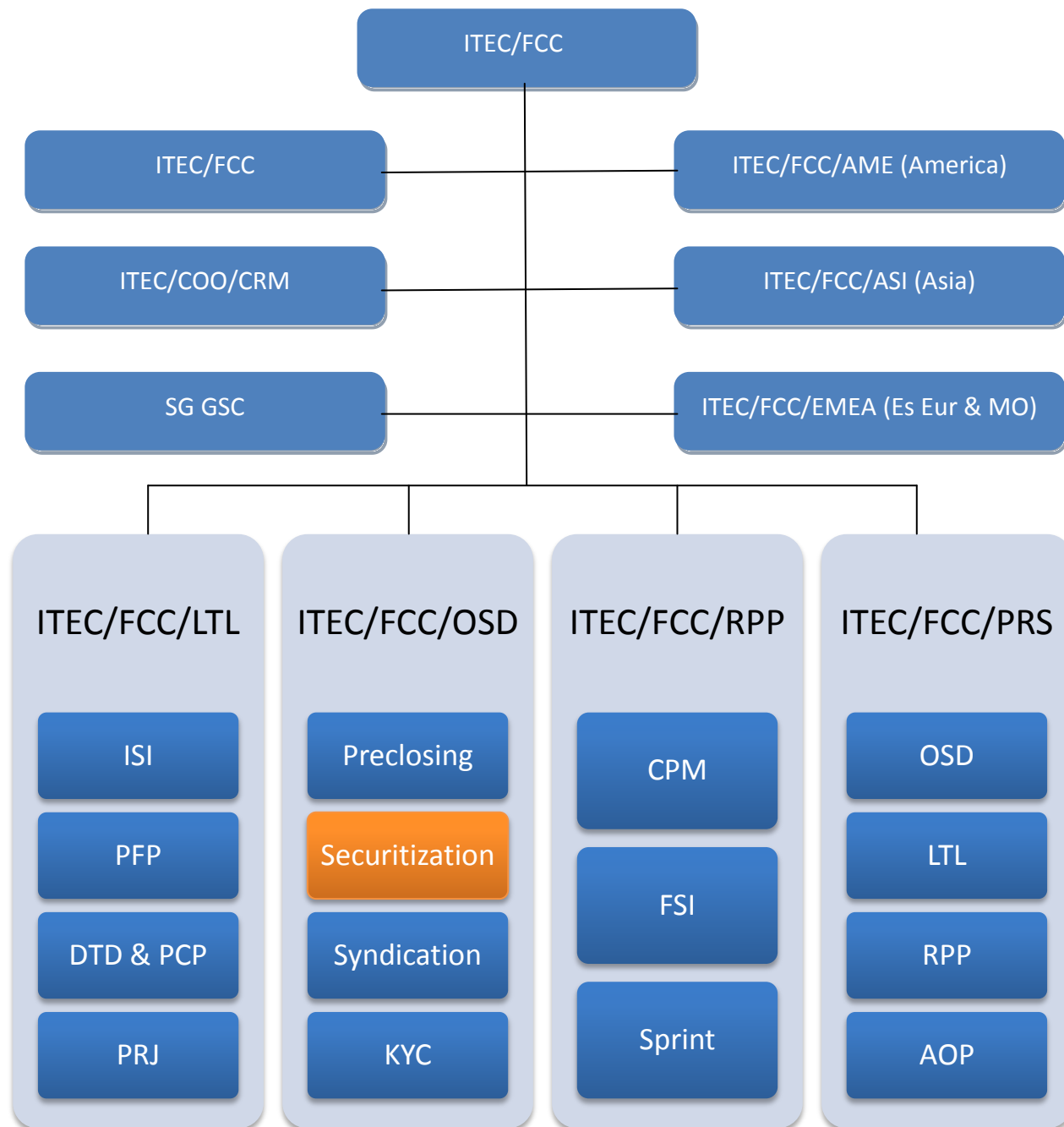


Figure 1 : Organisation de ITEC/FCC [1]

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)
- SCM (Syndicated Capital Market) tout ce qui concerne la titrisation et la syndication

b. 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).

II. Contexte et méthodologie adaptée

A. 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 en ne définissant plus ses concurrents comme les systèmes d'information des autres banques mais plutôt ceux des géants du web comme Google/Amazon/Facebook, une nouvelle approche est née. Le but est d'assurer rapidement un service adapté aux besoins de l'utilisateur et de bonne qualité. La méthode 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, 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.

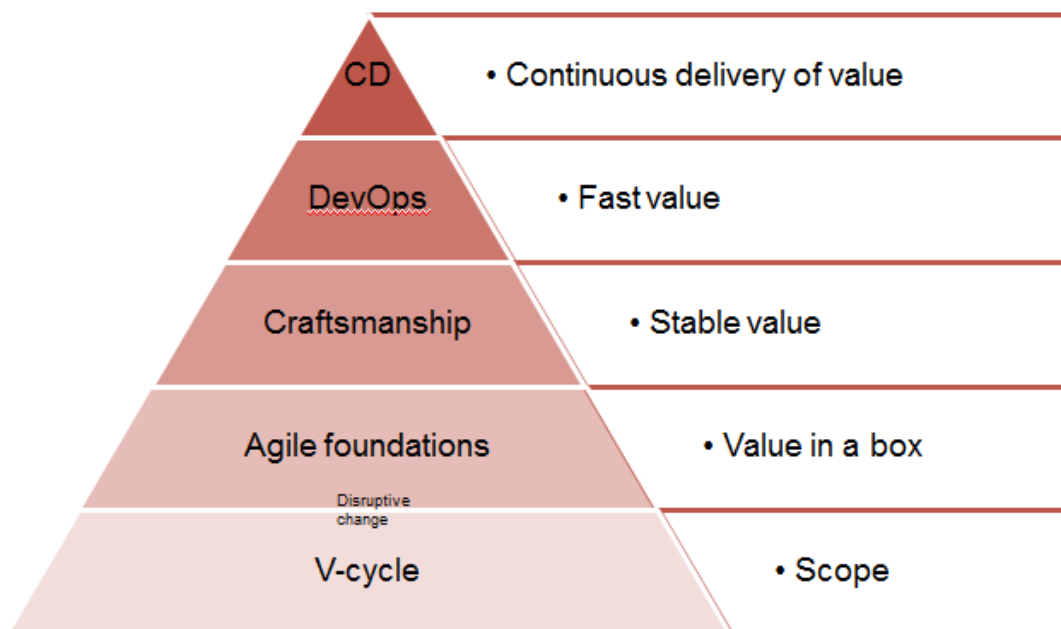


Figure 2 : les étapes de l'agilité chez ITEC [3]

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 développeur, 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.
- 2) **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.
- 3) **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 ».
- 4) **Continuous Delivery** : être capable d'effectuer des releases en continu sur toute la chaîne d'applications qui participent au business process. Cela est fait avec deployit

B. Organisation des projets dans l'équipe CBS

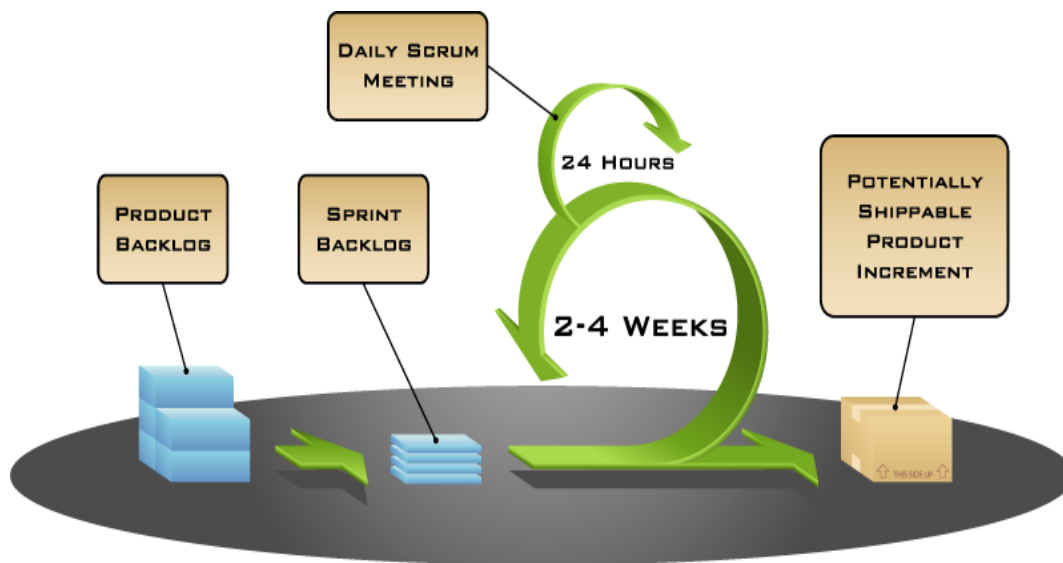
Il existe deux schémas d'organisation du développement, la méthode Scrum et Kanban. Dans cette partie, je vais vous présenter ces deux méthodes.

Dans le cadre du projet CBS, c'est la méthode Kanban qui est utilisée.

a. Méthode Scrum

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, avec laquelle il communique directement. Cette dernière estime le coût de chaque élément de la liste afin de lui fournir une prévision 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 est montré au client. Ce dernier peut alors se rendre compte par lui-même très tôt du travail réalisé, de l'alignement sur son besoin.



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

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 : lister 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.

- 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.

b. Méthode Kanban

Kanban est une méthode de gestion des connaissances relatives au travail, qui met l'accent sur une organisation de type Juste-à-temps en fournissant l'information ponctuellement aux membres de l'équipe afin de ne pas les surcharger. Dans cette approche, le processus complet de l'analyse des tâches jusqu'à leur livraison au client est consultable par tous les participants, chacun prenant ses tâches depuis une file d'attente.

Il existe cinq pratiques centrales de la méthode Kanban :

- Visualiser : La visualisation du workflow (flux de travaux) et sa matérialisation permettent de comprendre comment fonctionnent les processus. Un moyen courant de visualiser le workflow est d'utiliser un tableau avec des colonnes. Les colonnes représentent différents stades et étapes.

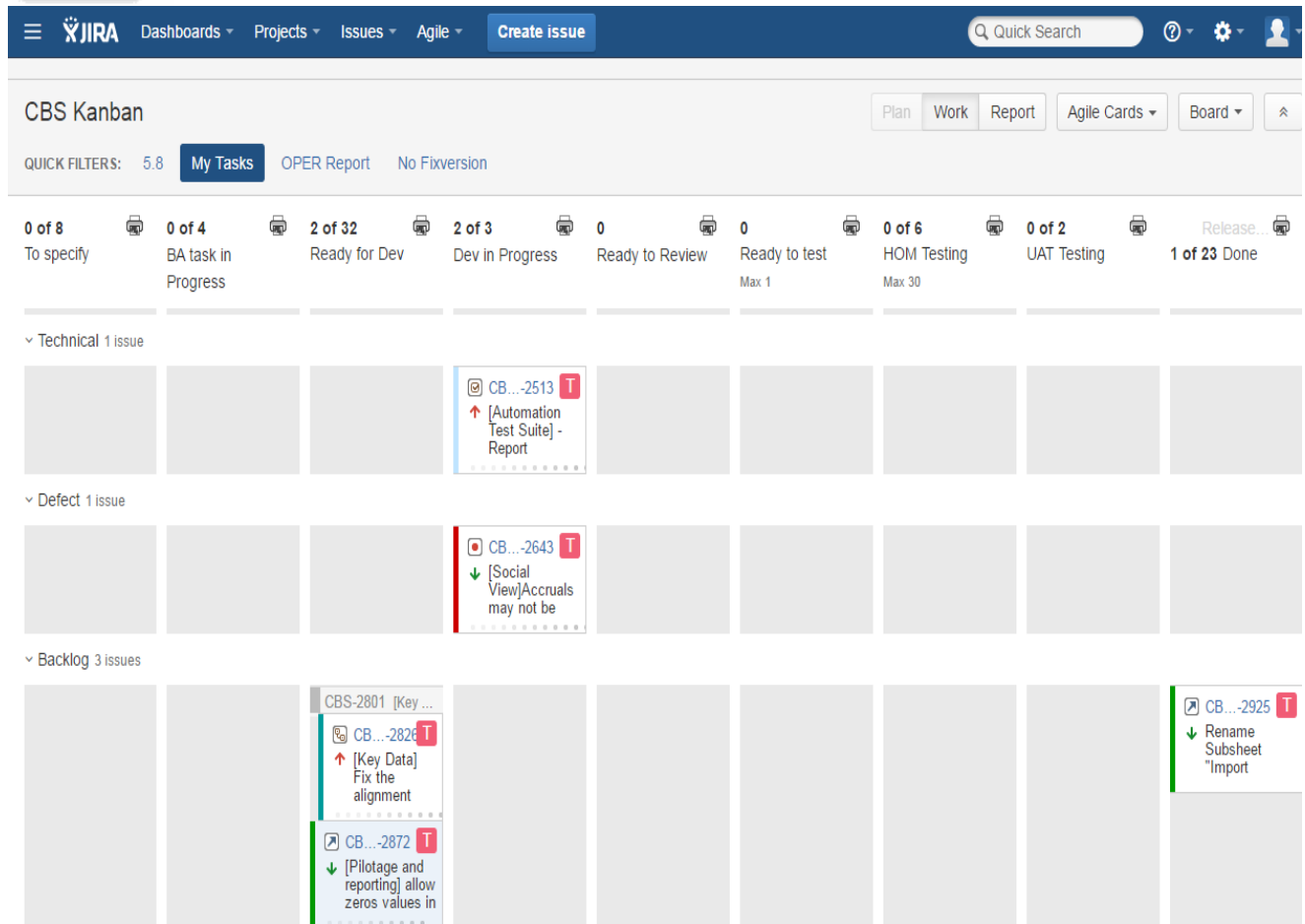


Figure 4 : visualisation du workflow

- Limiter le nombre de tâches en cours : La limitation des tâches suppose que le système de tirage est mis en application sur une partie ou sur l'ensemble du workflow. Le système de tirage servira de stimulus principal pour les changements continus, augmentés et évolutifs du système.
- Gestion du flux : Le déroulement du travail à travers chaque stade du workflow doit être suivi, mesuré et rapporté. En gérant activement le workflow, les changements continus, augmentés et évolutifs apportés au système peuvent être évalués.
- Rendre les normes de processus explicites : Établir les règles et recommandations par la compréhension des besoins et le suivi des règles. Les normes détermineront quand et pourquoi un ticket est déplacé d'une colonne à une autre.
- Utiliser des modèles pour reconnaître les opportunités d'amélioration : Lorsque les équipes partagent une compréhension des théories sur le workflow, le processus et le risque, elles pourront comprendre les problèmes et proposer des actions d'amélioration.



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.

C. Les impacts dans CBS

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



The screenshot displays the JIRA interface for issue CBS-2872. The header includes navigation tabs like Dashboards, Projects, Issues, and Agile, along with a 'Create issue' button and a search bar. The issue title is '[Pilotage and reporting] allow zeros values in parameters through a force generation flag'. Below the title are buttons for Edit, Comment, Assign, More, Open, In Progress, and Workflow. The 'Details' section shows the issue is an 'Improvement' of 'Minor' priority, currently 'IN REVIEW' with an 'Unresolved' resolution. The 'People' section lists the assignee as Thin-Hinane YOUNSI and the reporter as AHMED EL MORABITI. The 'Dates' section shows it was created on 08/Jul/16 at 11:48 AM and updated today at 9:06 AM. The 'Description' section contains the following text: '[Pilotage and reporting] allow zeros values in parameters through a force generation flag. Requirement is to add a flag next to Generate with following label "Force Generate" to allow to generate report even if there are values not populated. This flag is not visible for BO reports. This feature would allow the users to generate the respective report even though the relevant Specific parameters are not populated. DFS Update: Pilotage and Reporting DFS updated on page number 67.' The 'Attachments' section shows a document 'Force Generate BO Report issues.docx' (243 kB) uploaded on 08/Aug/16 at 1:33 PM.

Figure 5 : Aperçu du logiciel JIRA

III. Mission principale : le projet CBS

A. Présentation du projet

CBS (Covered Bonds System) s'occupe du suivi et de la génération de rapports réglementaires (monitoring et reporting). Le principe est d'émettre des obligations à partir d'autres entités SG indépendante juridiquement du groupe Société Générale. Celles-ci reçoivent des prêts de la SG en collatéral, 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 fournir le montant de l'obligation émise directement à la SG avec l'argent récupéré des investisseurs.

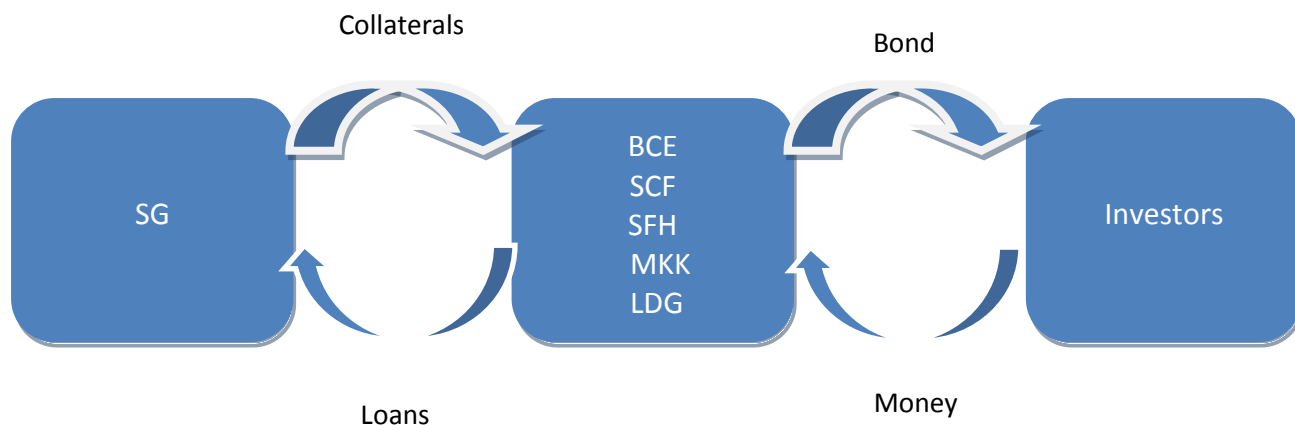


Figure 6 : Principe d'émission d'obligations couvertes

Problématique

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 créé l'application CBS qui permet de veiller à ce que :

- toutes les obligations émises soient couvertes par des actifs
- Reporting réglementaires soient fournis aux agences de notations

L'application CBS est partagée en deux équipes l'une à Bangalore et l'autre à Paris.



B. Organisation du stage

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

Le suivi 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 améliorations à des modules déjà existants
- Résoudre des problèmes de régression liés à des erreurs de compréhension ou d'une implémentation inappropriée du code
- Mettre en place/maintenir des tests unitaires (TDD : Test Driven Development)
- Développer des interfaces Front End dédiées aux utilisateurs de l'application

C. 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. Puis, un exemple d'une demande d'évolution. Enfin, les environnements de développement utilisés.

a. 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 technique de celle-ci, les différentes spécifications techniques, ainsi que l'architecture de l'application CBS.

i. Besoins fonctionnels

CBS possède cinq modules fonctionnels :

- Authentification
- Structure : permet la gestion/suivi de l'ensemble des instruments (prêts, obligation)
- Pilotage and Reporting : permet à l'utilisateur de générer tous les rapports réglementaires dans l'application
- Referentials : gère les référentiels tiers et privés
- Administration : gère les droits d'utilisateurs



Les acteurs de l'application CBS sont:

- Les utilisateurs
 - Membres de la Société Générale
 - Nombre par emplacement et par entité : 20 maximums à Paris
 - Pas d'utilisateurs international ni externe à la SG (le projet n'est pas ouvert sur internet)
 - Possèdent les droits de lecture et/ou d'écriture.
- 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 suivi des données de production

ii. Besoins non-fonctionnels :

Ces besoins permettent 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 : ouverture d'écrans et de délais de rafraîchissement
 - Performance en temps de traitements (fonctions, calculs, importations/exportation de données...

Ci-dessous, un diagramme d'un cas d'utilisation de l'application CBS :

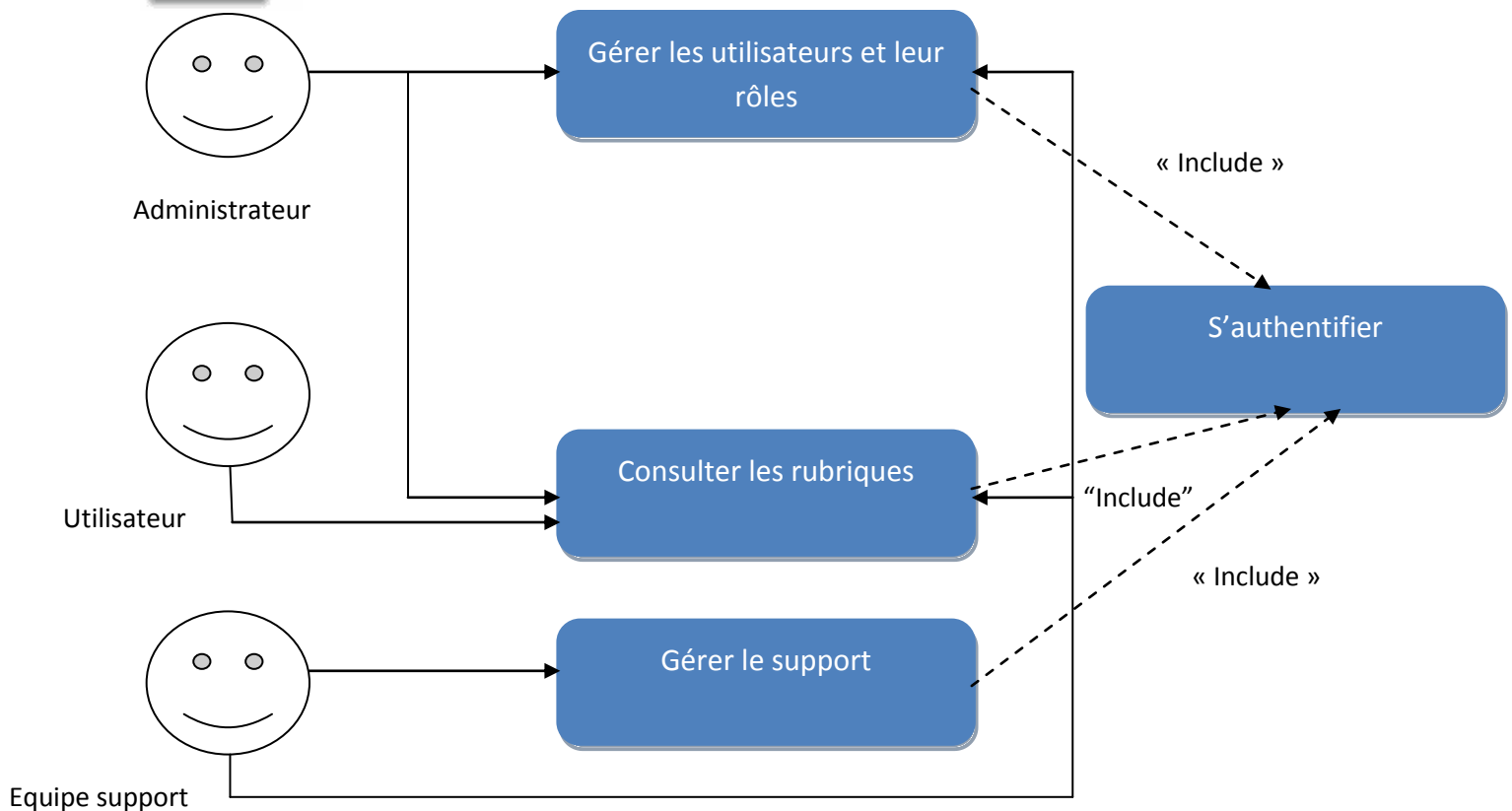


Figure 7 : Diagramme des cas d'utilisation général

iii. Spécification technique

L'architecture technique de CBS est une fusion de technologies bien établies et émergentes.

Il y'a ainsi quelques contraintes et choix technique qui ont été mis en place pour le développement de l'application :

- Respect des normes d'ITEC/FCC : architecture basée sur la technologie J2EE.
- Accès facile et sécurisé : utilisation de l'architecture client léger (HTTPS)
- Confidentialité et accès des données : Les autorisations seront gérées par Spring Security 3. Cela permet d'avoir une solution de maintenance de coûts flexibles et faibles pour la gestion de l'autorisation.
- Authentification : Sesame (outil interne à la Société Générale) est utilisé pour la couche d'authentification

iv. Architecture de l'application

Tout d'abord, nous avons l'architecture physique qui décrit l'ensemble des composants matériels. Ci-dessous un schéma représentant les composants logiciels (base de données) déployés sur les composants matériels (machine virtuelle).

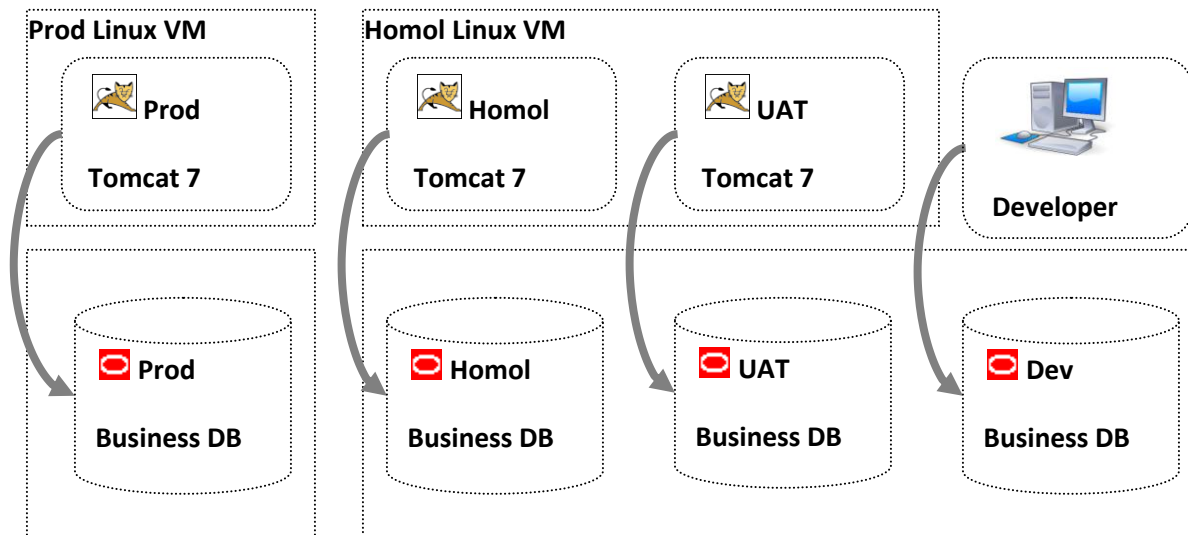


Figure 8 : Schéma technique de l'architecture physique de l'application CBS [2]

Les Business BD utilisées sont sous Oracle 11.

Puis, dans un second temps nous avons l'architecture logique qui permet de décrire les différentes couches qui communiquent entre elles à travers des requêtes (HTTPS, SQL,...). Ci-dessous un schéma représentant cette structure.

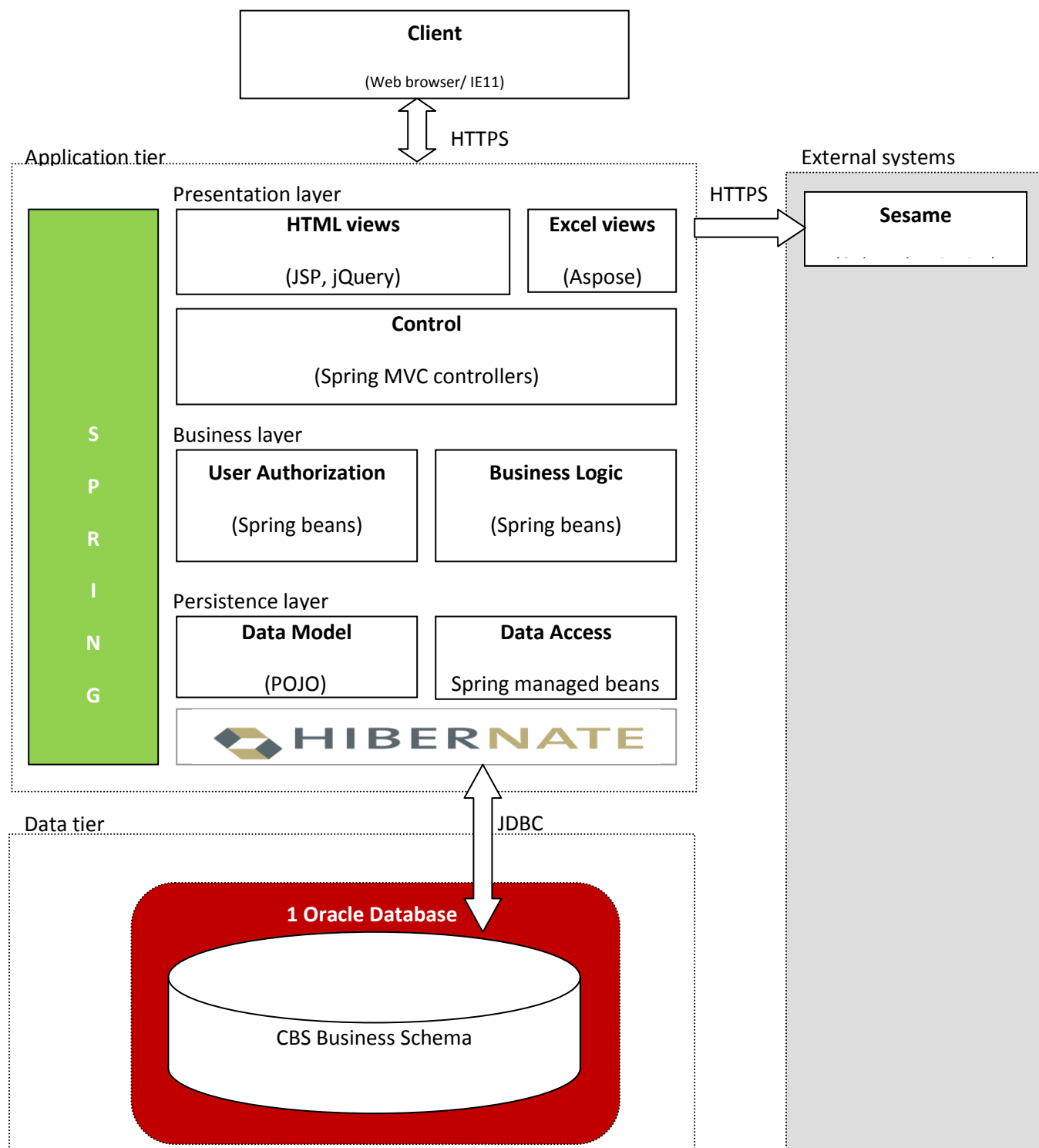


Figure 9 : Schéma de l'architecture logique de l'application CBS [2]



Les niveaux de l'application CBS suivent une architecture en couches superposées:

- Couche de présentation
- Couche métier encapsulant la logique métier et le contrôle
- Couche de persistance encapsulant la logique d'accès aux données

L'assemblage des couches est basé sur POJO et utilise le Framework Spring.

Les Frameworks qui sont utilisés dans chaque couche sont :

- JPA et Hibernate 3 pour la couche de persistance.
- Spring, Spring MVC, JSP, Aspose pour les deux couches : présentation et métier.

L'injection de dépendance de Spring permet de réduire le couplage entre les couches.

b. Conception

Dans cette partie, je vais présenter la phase conceptuelle qui résulte du croisement entre la branche fonctionnelle et la branche technique.

i. Conception de base

On peut représenter la dépendance fonctionnelle des données du système à l'aide d'un diagramme de packages.

En effet, notre application est divisée en quatre paquetages :

- Le paquet présentation : qui communique directement avec le paquet service afin de d'importer les éléments nécessaires pour créer les pages JSP.
- Paquet service : Des classes de services traitent les méthodes nécessaires d'accès à la base de données
- Paquet persistance : ce sont les classes qui sont en interaction avec la base de données. Elles sont gérées par JPA (Java Persistence API).
- Paquet repository : facilite l'écriture des couches d'accès aux données.

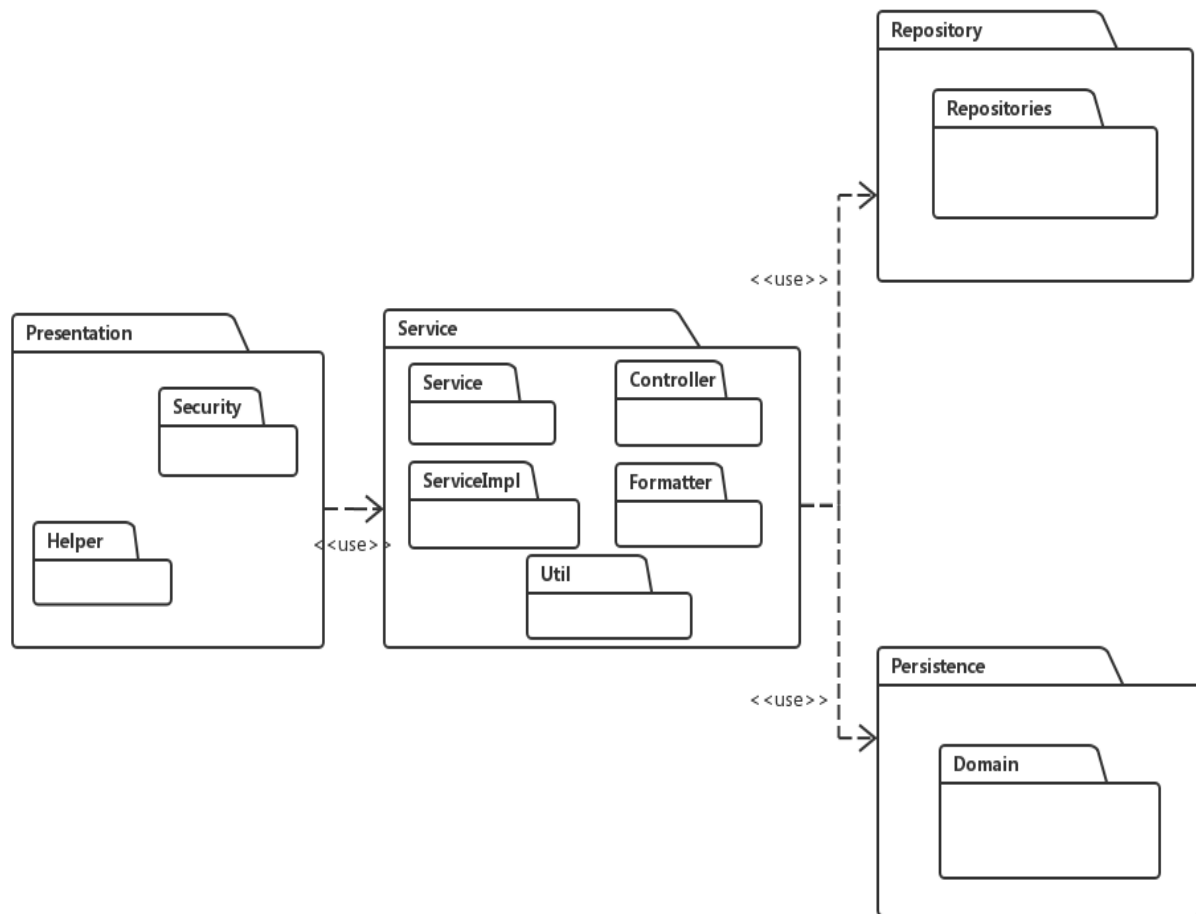


Figure 10 : Digramme de package

Les éléments qui composent ce système ainsi que leur relation peut être, quand à lui représenté à l'aide d'un diagramme de classe.

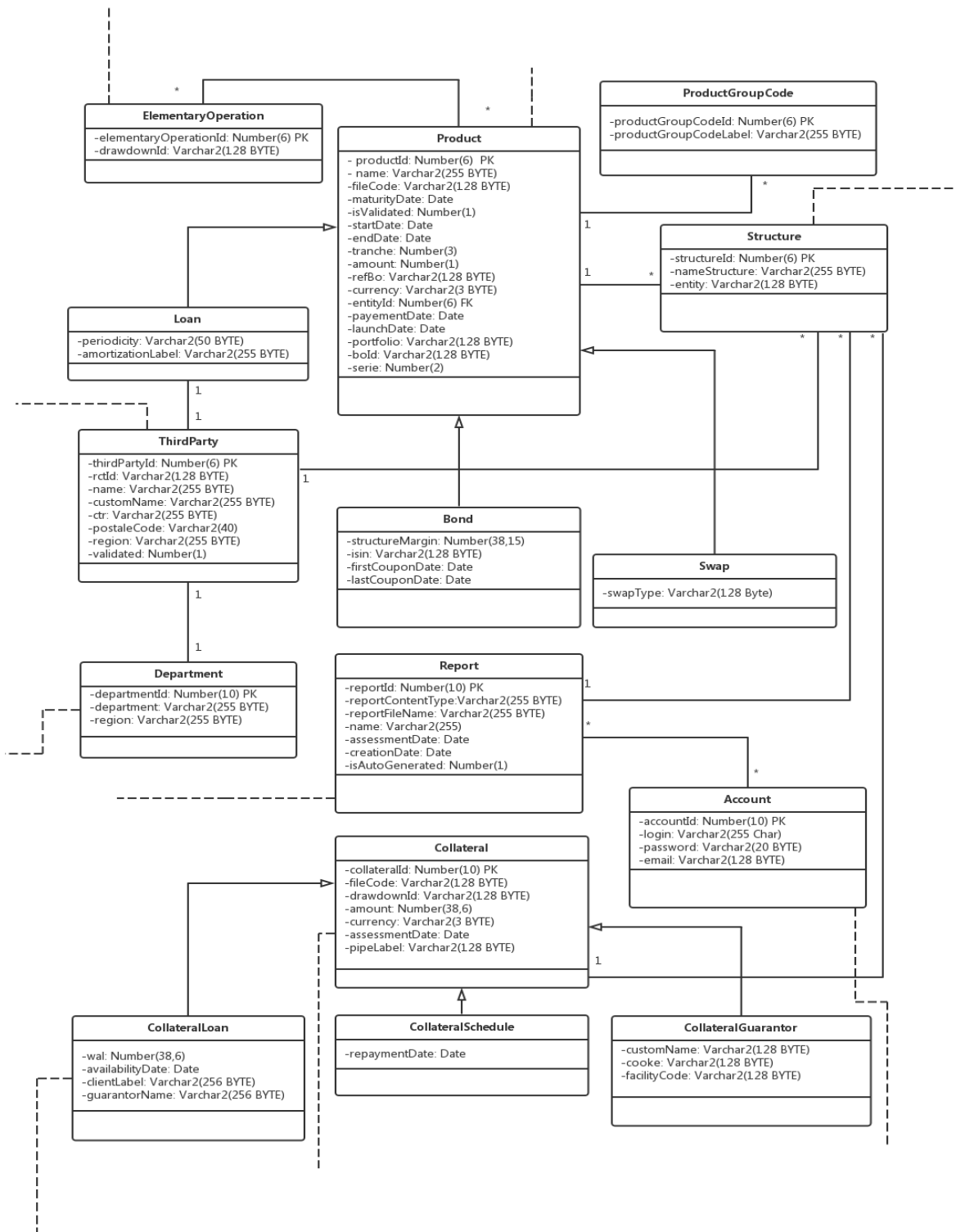


Figure 11 : Diagramme de classe global



ii. Conception : les solutions technique

Afin de satisfaire le cahier des charges et répondre aux attentes du client, des solutions technique sont mise en place. C'est ici que l'ingénieur se base sur son expérience, ainsi que sur les patrons de conception. Le but recherché est d'obtenir des logiciels de grande ampleur qui soient fiables, de qualité, et correspondent aux attentes de l'utilisateur.

1. Design pattern

Un design pattern est une solution à un problème récurrent dans la conception d'applications orientées objet. Un patron de conception décrit alors la solution éprouvée pour résoudre ce problème d'architecture de logiciel. Il permet en plus de gagner en rapidité et en qualité de conception. De plus, il facilite grandement la maintenance car les futurs développeurs savent le reconnaître.

Un patron de conception est représenté par :

- Nom : qui permet de l'identifier clairement.
- Problématique : description du problème auquel il répond.
- Solution : description de la solution souvent accompagnée d'un schéma UML.
- Conséquences : les avantages de cette solution.

Dans notre application, le patron d'architecture utilisé est le Framework Spring MVC (Model View Controller). Le modèle vue contrôleur est souvent décrit comme simple design pattern (motif de conception) mais c'est plus un architectural pattern (motif d'architecture) qui donne le ton à la forme générale d'une solution logicielle plutôt qu'à une partie restreinte.

Il possède trois parties qui sont :

- Le Modèle : Le modèle définit les données de l'application et les méthodes d'accès. Tous les traitements sont effectués dans cette couche.
- La Vue : La vue prend les informations en provenance du modèle qui lui sont passées par le contrôleur et les présente à l'utilisateur.
- Le Contrôleur : Le contrôleur répond aux événements de l'utilisateur et commande les actions sur le modèle. Cela peut entraîner une mise à jour de la vue.

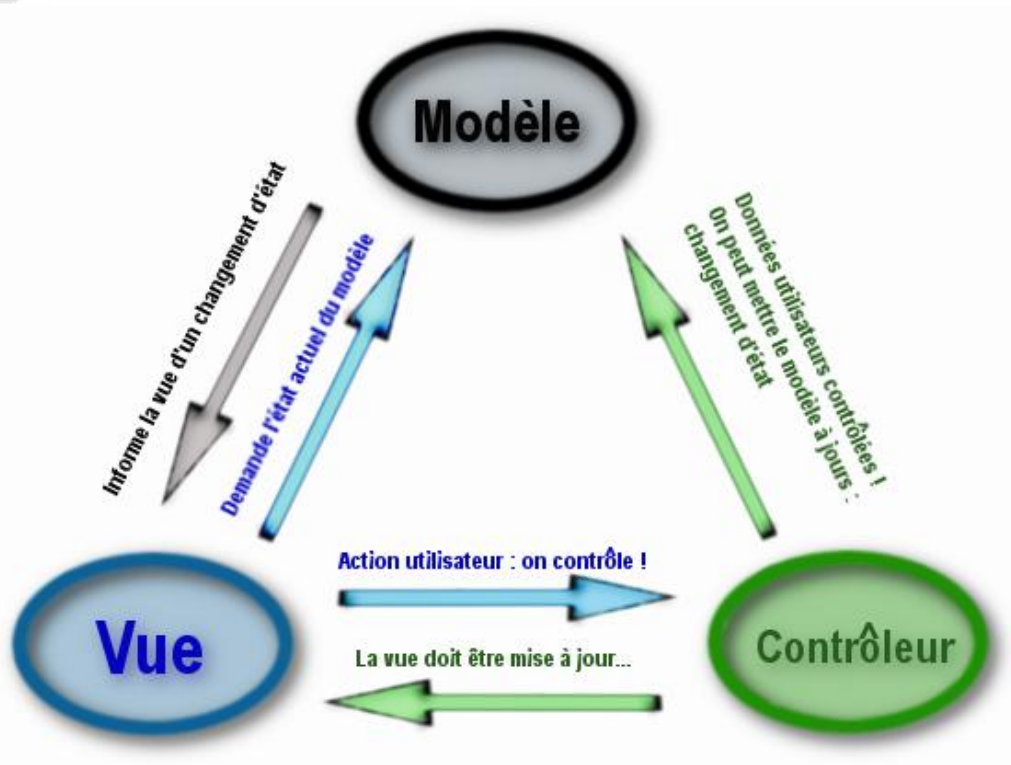


Figure 12 : Le pattern Modèle Vue Contrôleur [2]

Un autre design pattern utilisé est l'IOC (Inversion de contrôle). C'est un motif de conception logicielle commun à tous les Frameworks, devenu populaire avec l'adoption des conteneurs dits légers.

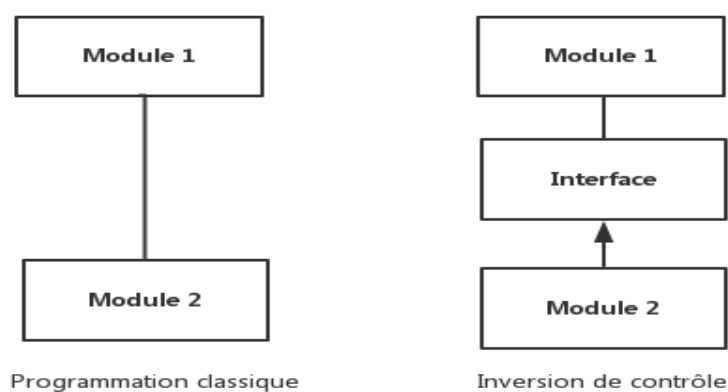


Figure 12 : principe de l'inversion de contrôle



L'idée est que les modules de haut et bas niveaux doivent dépendre d'interface, mais plus de dépendance entre module. Les modules dérivent des interfaces. L'inversion de contrôle les rend donc indépendant entre eux.

c. Mes activités dans l'application CBS

Dans cette partie, je vais vous présenter une des demandes d'évolution que j'ai effectuée durant mon stage. Il s'agissait d'une amélioration d'une fonctionnalité déjà existante.

i. Existant

La demande était relative au module Pilotage and Reporting, qui permet à l'utilisateur de générer tous les rapports dans l'application.

Il existe trois types de rapports dans l'application que nous pouvons générer : rapport Moody's, rapport S&P et rapport BO.

Une fois que l'utilisateur a accès à la vue de «Pilotage et Rapports», il doit suivre les étapes ci-dessous pour générer un rapport :

- 1) Naviguer dans la partie Generate
- 2) Saisir les données (Structure, date d'inventaire, type de rapport et format du rapport)
- 3) Pour pouvoir générer un rapport, il suffit de cliquer sur le bouton «Generate» de manière à y accéder à la page «Générer» permettant la génération du rapport approprié.

TEAM SPIRIT SOCIÉTÉ GÉNÉRALE Covered Bonds System v 5.7 - DEV thin Logout

Structure Pilotage and Reporting Referentials Administration

List Search Generate Monitoring 0

Assessment Date 2016-08-23 Projection Date 2016-08-23 Rate Date 2016-08-23

Structure SCF Entity AQB4507 - SCF ScenarioRef SCF-SCF-2016-08-23-01

Currency ALL Report Type [ALM] Pilotage ALM - Nouvelle vi

Report Format* Excel Use Forward Rates NO Store Scenario Open Scenario

Liabilities Collaterals Specific Parameters Graph

Generate

Discard Scenario Release Connection

Figure 13 : Fenêtre Pilotage and Reporting

Ci-dessous un diagramme représentant la génération d'un rapport.

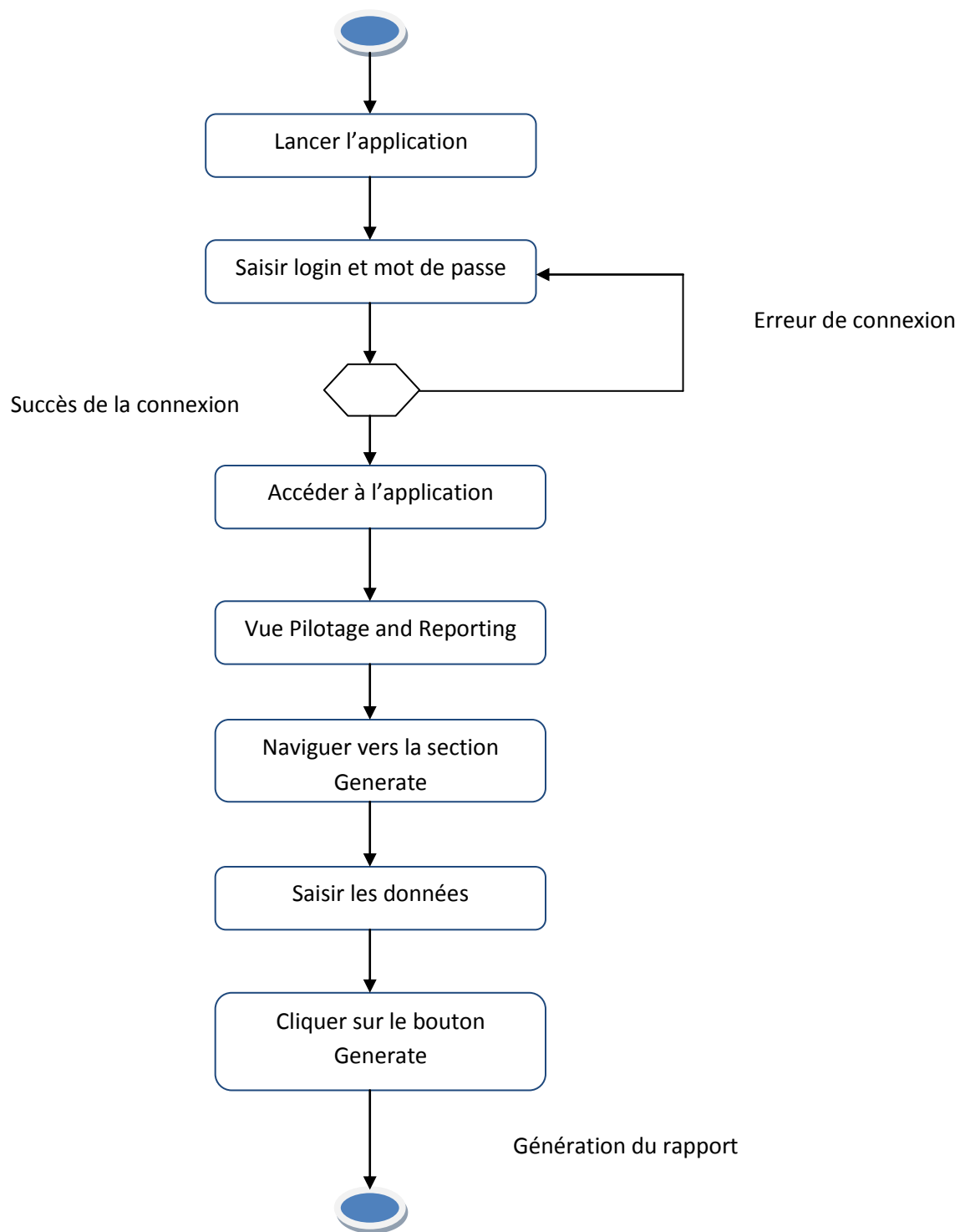


Figure 14 : Diagramme de génération d'un rapport

Cependant, pour certains types de rapport tel que les ALM projection (utilisant la technologie aspose), on ne pouvait générer un rapport que si certains paramètres spécifiques étaient remplis. Dans le cas contraire, le bouton « générer » était désactivé.

Liabilities
Collaterals
Specific Parameters

Taux de RA

%

Taux de couverture

%

WAL Gap Limit

Taux de couverture réglementaire

%

Haircut BCE

%

Ressources privilégiées (ratio de couverture)

Actifs Eligibles (ratio de couverture)

Enhanced WAL computation

N

Liquidity focus

2018-12-31

Rate choc applied for sensitivity

%

Periodicity

Daily

Graph

Generate

Figure 15 : Cas où le bouton Generate est désactivé

ii. Evolution

La demande de l'utilisateur était d'avoir voir un autre bouton nommé « Force Generate » qui permettait de générer un rapport même lorsque les paramètres spécifiques n'étaient pas saisis.

Ce bouton ne concerne pas tous les rapports généré via les rapports business object.

Structure		Pilotage and Reporting		Referentials		Administration	
List	Search	Generate	Monitoring 0				
Assessment Date	2016-08-23	Projection Date	2016-08-23	Rate Date	2016-08-23		
Structure	BCE	Entity	AWP2615 - ESNi	ScenarioRef	BCE-ESNI-2016-08-23-02		
Currency	ALL	Report Type	[ALM] Projections				
Report Format*	Excel	Use Forward Rates	NO	Store Scenario	Open Scenario		

Liabilities	
Collaterals	
Specific Parameters	
Taux de RA	<input type="text"/> %
Taux de couverture	<input type="text"/> %
Taux de couverture réglementaire	105.00 %
Periodicity	Daily

Graph	
Generate	Force Generate

Figure 16 : Bouton Force Generate

Ainsi, nous pouvons bien voir l'apparition du bouton « Force Generate » et la désactivation du bouton « Generate » au même moment.

d. Outils technologique

Dans cette partie, j'exposerai tout d'abord les outils de développement, puis je présenterai les différents outils de gestion avant de passer aux bases de données.

1. Outils de développement

L'environnement de développement avec laquelle je travaille correspond à :



IntelliJ IDEA qui est un IDE JAVA commercial développé par JetBrains. C'est un environnement de développement permettant de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Il dispose également de possibilité de navigation avancée dans le code et de refactorisation de code.



HIBERNATE

Hibernate est une solution open source de type ORM (Object Relational Mapping) qui permet de faciliter le développement de la couche persistance d'une application. Hibernate permet donc de représenter une base de données en objets java et vice versa.

Hibernate facilite la persistance et la recherche de données dans une base de données en réalisant lui-même la création des objets et les traitements de remplissage de ceux-ci en accédant à la base de données et en y accolant un système de cache.



Apache Tomcat est un serveur d'application de servlets et JSP Java EE. Issu du projet Jakarta, c'est un des nombreux projets de l'Apache Software Foundation. Il implémente les spécifications des servlets et des JSP du Java Community Process2, est paramétrable par des fichiers XML et des propriétés, et inclut des outils pour la configuration et la gestion. Il comporte également un serveur HTTP.



Maven est un outil pour la gestion et l'automatisation de production des projets logiciels Java en général et Java EE en particulier son objectif est de produire un logiciel à partir de ses sources, en optimisant les tâches réalisées à cette fin et en garantissant le bon ordre de fabrication. Il gère le cycle de vie classique d'un projet Java.



JUnit est un framework de test unitaire pour le langage de programmation Java.



Le JavaServer Pages ou JSP est une technique basée sur Java qui permet aux développeurs de créer dynamiquement du code HTML, XML ou tout autre type de page web. Cette technique permet au code Java et à certaines actions prédéfinies d'être ajoutés dans un contenu statique.

2. Outils de gestion



L'outil JIRA est une application web permettant à tous les membres de l'équipe de suivre pas à pas, l'avancement, les réalisations, et les erreurs à corriger, tout au long du projet. Il a été développé par Atlassian.



Deploy It est un outil développé par Xebia qui permet d'automatiser les déploiements des packages. C'était un prérequis essentiel aux releases étant donné qu'il effectue les mises en production plus rapidement (en quelques minutes) et surtout qu'il permet aux équipes d'être autonome. En effet s'il fallait soumettre les releases plus de cinq jours à l'avance, c'est parce qu'on avait besoin d'autres équipes pour le déploiement. Deploy It permet de se passer de cette contrainte et donc de soumettre les releases (et ainsi de les remplir) seulement le jour même si on le veut.



SonarQube est un logiciel libre permettant de mesurer la qualité du code source.



3. Base de données



Oracle SQL Developer est un environnement de développement intégré multiplateforme, fourni gratuitement par Oracle Corporation et utilisant la technologie Java. C'est un outil permettant d'interroger des bases de données Oracle à l'aide du langage SQL.

IV. Software factory : Création de slaves Jenkins avec ITAAS

A. 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 tâches telle que la compilation, le lancement des tests unitaires et fonctionnels, qualité du code et rafraichissement des bases de données (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 Repository Github (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 tâches 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.

Cependant, les applications de l'entité FCC/OSD/SCM se partageaient le même slave Jenkins pour les lancements des jobs Jenkins, en environnement de développement. Ce slave correspond à un serveur Windows composé de quatre 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.

B. Quelles évolutions pour quels impacts ?

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

Cela permettait 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 était 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 :

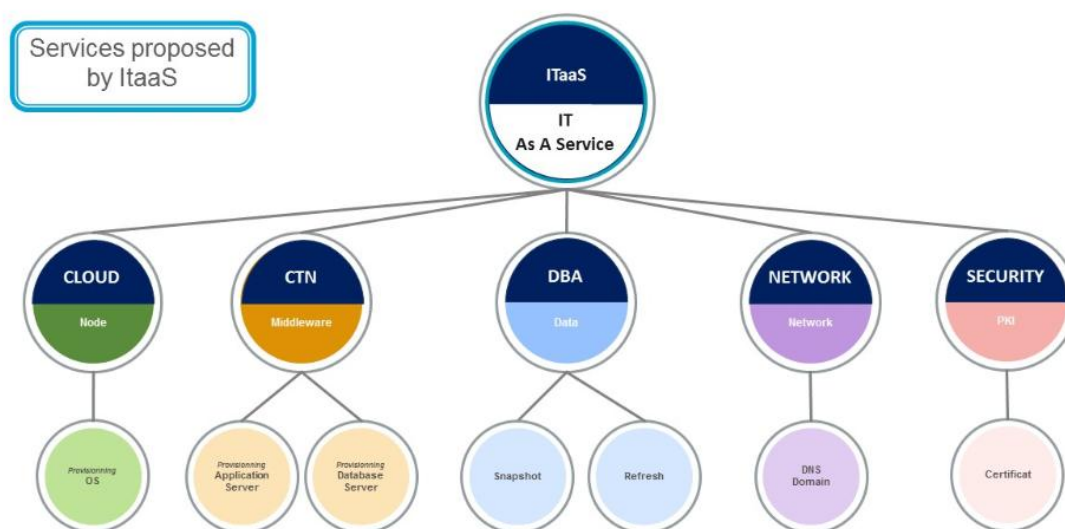


Figure 17 : Service proposés par ITAAS

Le service qu'on utilise est le Cloud, qui nous permet de créer de nouveaux servers (linux ou Windows) et de les héberger sur le cloud de l'entreprise. Le cloud étant un moyen d'abstraire l'emplacement physique de chaque machine. Ceux-ci sont ensuite utilisés comme slave Jenkins.

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

Moi je me suis occupée de la création des serveurs pour l'application TDA, ainsi que la configuration des jobs Jenkins pour CBS.

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



C. 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
- La taille de stockage
- Aucune expiration

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 fichier Json dans lequel on spécifie tous les paramètres, ce qui correspond à faire une « request ».

i. Rest Client

Tout d'abord, je configure **Rest Client**. www.restclient.net est un plug-in Firefox, testé avec succès avec l'API ITaas.

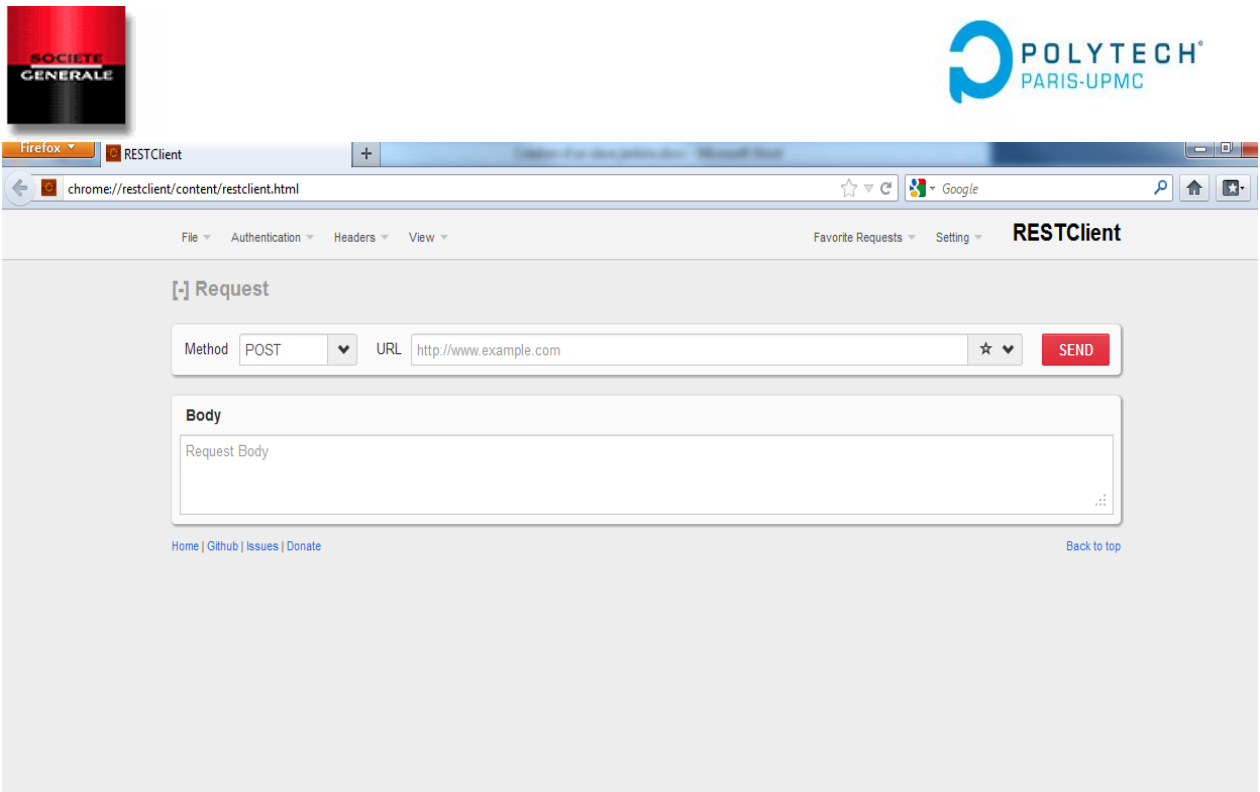


Figure 18 : Vue de l'interface Rest Client

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 apparaît dans la section Headers (Encodage en Base64)

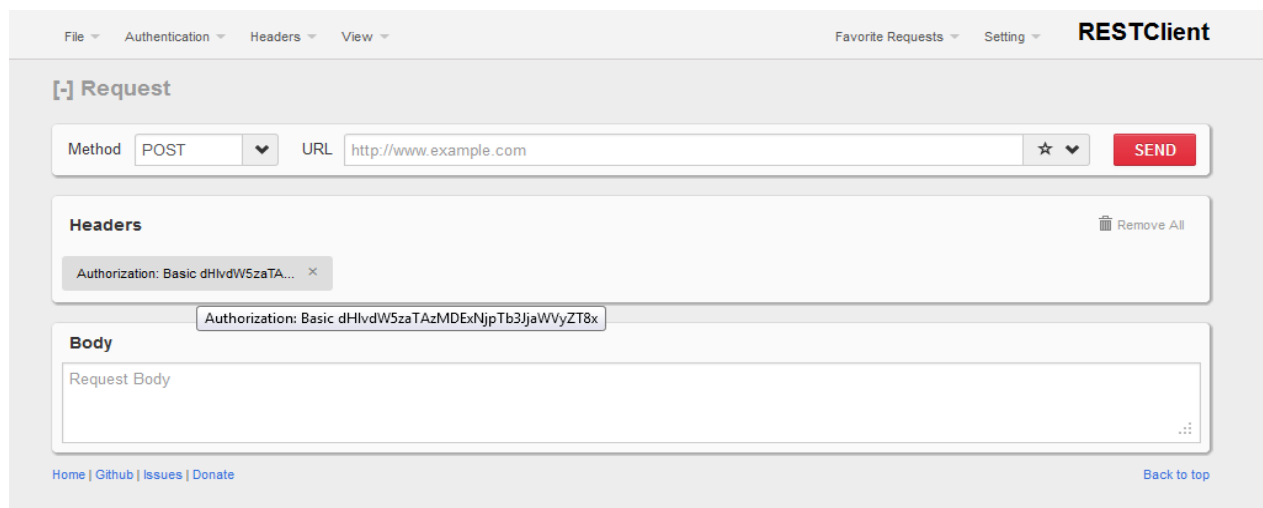


Figure 19 : Sélection de Basic Authentication

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

- Dans le menu « Headers », on choisit « custom 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 »

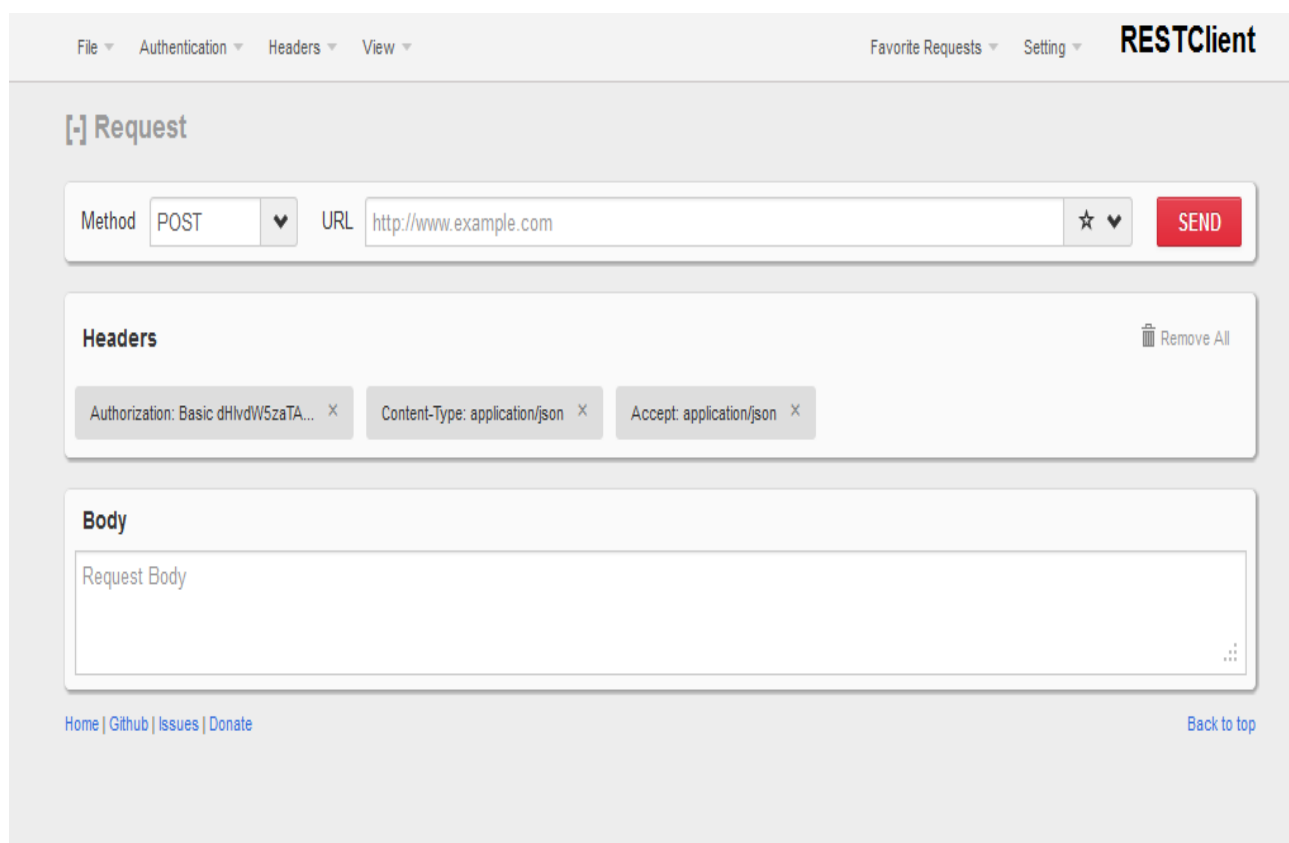


Figure 20: Ajout des headers Accept et Content-Type

A la troisième étape, on choisit la méthode d'envoi du formulaire, la méthode **POST** sera utilisée dans notre cas, Car c'est la méthode qu'attend ITAAS.

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

ii. 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 :

```
{
  "kind": "linux",
  "size": "S",
  "disksize": 20,
  "leasedays": 1,
  "environment_type": "dev",
  "centrify_zone": "zd_fr_cld_01",
  "description": "my description",
  "usergroup": "clddevsudo01",
  "service_account": "pocst01",
  "app_trigram": "TRG",
  "region": "eu-fr-paris",
  "availability_zone": " eu-fr-paris-1",
  "subnet": "192.160.64.0/21",
  "notifymail": "user1@example.com,user2@example.com>",
  "master_os": "RHEL 6.7 x64 MKT",
  "label": ""
}
```

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

- Size : "M"
- Disk size : "60", pour les dossiers contenant maven/git/java ...
- Lease days : "-1" afin que la durée de mise à disposition du serveur n'expire pas
- environment type : "dev"
- Centrify zone : "zd_fr_cbs_01" créée par GTS group
- Description : Jenkins slave for CBS
- User group : "cbsdev" créée GTS group
- service account : "cbsdev02"
- 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
- master_os : RHEL 6.7 x64 MKT
- label : slave_jenkins_linux_cbs

Enfin, on lance la procédure 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.

```
{
  "id": 8873,
  "kind": "linux",
  "hostname": null,
  "user": "oterrien092210",
  "created_at": "2016-05-26T13:35:54.355Z",
  "size": "M",
  "disksize": 60,
  "leasedays": -1,
  "usergroup": "cbsdev",
  "app_trigram": "CBS",
  "finished": false,
  "centrify_zone": "zd_fr_cbs_01",
  "service_account": "cbsdev02",
  "environment_type": "dev",
  "description": "Jenkins salve for CBS",
  "label": null,
  "notifyemail": "<olivier.terrien@sgcib.com, thin-hinane.younsi@sgcib.com>",
  "master_os": "RHEL 6.5 x64 MKT",
  "region": "eu-fr-paris",
  "availability_zone": " eu-fr-paris-1",
  "subnet": "192.160.64.0/21",
  "creation_job_id": 222441,
  "remaining_days": -1
}
```

iii. Création d'une clef SSH

Une fois que le serveur est provisionné dans le Cloud, on génère une paire de clés publique et 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 :

- ~/.ssh est en chmod = 700
- authorized_keys en chmod = 600

iv. 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}>

TDA UAT :

```
{ "id": "222807", "status": "running", "date_started": "2016-05-27T08:24:11Z", "date_ended": null, "job_name": "Create Linux", "dry_run": "no", "node_id": 8893, "middleware_id": 4136 }
```

 VM

TDA Hom:

```
{ "id": "222809", "status": "running", "date_started": "2016-05-27T08:36:21Z", "date_ended": null, "job_name": "Create Linux", "dry_run": "no", "node_id": 8894, "middleware_id": 4137 }
```

 VM

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 Linux", "dry_run": "no", "node_id": 8893, "middleware_id": 4136 }
```

 VM

- TDA Hom:

```
{ "id": "222809", "status": "succeeded", "date_started": "2016-05-27T08:36:21Z", "date_ended": "2016-05-27T09:09:56Z", "job_name": "Create Linux", "dry_run": "no", "node_id": 8894, "middleware_id": 4137 }
```

 VM



V. Conclusion

J'ai beaucoup apprécié cette expérience très enrichissante qui m'a énormément appris dans de nombreux domaines. J'ai pu également apporter durant ce stage des compétences et connaissances que j'ai acquises de par ma formation et cela a été l'occasion de les mettre en pratique.

Les enseignements reçus en entreprise et à l'école sont différents, bien que liés par certains aspects du domaine. L'aspect pratique est prédominant en entreprise, et beaucoup de concepts s'apprennent par la force des choses. Cependant les notions théoriques vues durant les études n'en restent pas moins formatrices.

Pouvoir développer sur l'application CBS m'a beaucoup plu. En effet, cela m'a permis d'améliorer ma compétence technique et de connaître bien le périmètre et les processus métier associés.

Ce stage s'est bien passé pour moi, j'ai été bien accueillie et bien intégrée à l'équipe. Les compétences techniques sont maîtrisées à l'issue de ce stage. De plus, le cadre de travail m'a permis d'enrichir mon expérience, car j'ai évolué dans une équipe internationale avec des collègues à Bangalore (Inde). J'ai pu rencontrer et interagir avec des équipes et des métiers différents : Business Analyst (maîtrise d'ouvrage), support, les équipes techniques et les utilisateurs. Cela m'a donné une vision des différents métiers de l'IT dans le domaine bancaire. De cette expérience, j'ai pu développer la capacité d'adaptation de ma communication en fonction des interlocuteurs.



Glossaire

CBS	Covered Bond System : Application permettant de gérer l'émission d'obligations garanties
FCC	Financing and Client Coverage Technology
GBIS	Global Banking and Investor Solutions
ITEC	Information Technology Department
OSD	Origination, Structuring and Distribution
Release	Livraison d'une nouvelle version d'une application
Sesame	service d'authentification
SG	Société Générale
SCF	Sociétés de Crédit Foncier
SGCIB	Société Générale Corporate and Investment Banking
TDA	Application permettant de gérer des montages financiers particuliers en Europe
UAT	User Acceptance Tests
HOM	La phase d'homologation d'un projet : étape primordial consiste à effectuer un certain nombre de tests pour vérifier le niveau de qualité de l'application livrée, sa conformité aux spécifications fonctionnelles détaillées ainsi qu'aux exigences techniques.



Bibliography

[1] Intranet "My Société Générale"

[2] Presentation de l'application CBS, document interne, 7 janvier 2014

[3] Agile Maturity Model, document interne

Annexe 1 : Organigramme de GBIS

