



Office Automation Analyst

Jonathan DEKHTIAR

Génie Informatique

Université de Technologie de Compiègne

Email : jonathan.dekhtiar@etu.utc.fr // contact@jonathandekhtiar.eu

Marc LAURENT

IS Shared Service Manager

Valeo DAV - Annemasse

Email : marc.laurent@valeo.fr

Ali CHARARA

Suiveur Universitaire

Université de Technologie de Compiègne

Email : ali.charara@hds.utc.fr

Valeo DAV
Rue Jules Verne
74100 Annemasse

Remerciements

Je tiens tout particulièrement à remercier M. Marc LAURENT, ISSS Manager, non seulement pour m'avoir accepté en tant que stagiaire Assistant Ingénieur pour le poste d'Office Automation Analyst mais également pour m'avoir accordé sa confiance sur de nombreux projets, me donnant ainsi l'opportunité de découvrir nombreux sujets passionnants et le fonctionnement d'un grand groupe comme Valeo.

Mes remerciements s'adressent tout autant à Mme Sylvie BERTOIN, Coordinatrice RH et Paie, qui m'a donné la chance de réaliser cette expérience.

J'aimerais remercier les personnes avec qui j'ai pu échanger sur de nombreux sujets : Adnene ELBEJI, Gabriela-Isabela LENOIR, Caroline DEPREUX, Catherine Houlbert, Hakim EL-ABDELLAOUI, Jean-Philippe BERNIER, Karl VERDIER, Nada BOUDALI, Patrice GEORGES, Stephane GUERRIER pour les différents points de vue de l'entreprise qu'ils ont sus me faire partager.

Finalement, je remercie plus particulièrement Alice METRAL, Christine BENEDETTI, Guy BOSSONNET, Leandre CASANOVA et Marc BIZET pour leur sympathie, pour l'aide qu'ils ont pu me fournir pendant ma mission, leur professionnalisme et les compétences que j'ai pu acquérir en travaillant avec eux.

Table des matières

Remerciements	2
Introduction	8
1 Présentation de l'entreprise	9
1.1 Valeo, un équipementier Automobile	9
1.1.1 Présentation de l'entreprise	9
1.1.2 La structure interne de Valeo.	10
1.1.3 La structure interne du Business Group : CDA	11
1.2 L'IS Shared Service Center - CDA France	12
2 Présentation du Sujet de Stage	14
2.1 Présentation du sujet de stage	14
2.2 Fiche de Poste	15
3 Le Cloud Computing	16
3.1 Qu'est ce que le Cloud Computing ?	16
3.2 IaaS vs PaaS vs SaaS vs STaaS	17
4 Cordys Process Factory	19
4.1 Une plateforme Applicative "Cloud"	19

4.1.1	Qu'est ce que Cordys Process Factory ?	19
4.1.2	Lotus Notes vs Cordys Process Factory	20
4.1.3	Le contexte dans lequel s'insère Cordys Process Factory	20
4.2	Présentation du fonctionnement de la Plateforme de Service	22
4.2.1	Architecture de la plateforme	22
4.2.2	Présentation de l'interface standard d'une application	24
4.2.3	Présentation de l'interface developpeur de Cordys	27
4.3	Titre de Fonction : "Automation Analyst"	35
4.3.1	Description de mon rôle	35
4.3.2	Retour sur expérience	36
5	Un Système ERP : SAP	37
5.1	Un système ERP : SAP	37
5.1.1	SAP, vue d'ensemble	37
5.1.2	Les intérêts Business de l'utilisation du système SAP	38
5.2	Le projet Comp@ssV4	39
5.2.1	Une migration logicielle et architecturale	39
5.3	Un client = Une fonction partenaire	42
5.3.1	Les différentes fonctions partenaires	42
5.4	Reprise des clients dans Comp@ssV4	43

5.4.1	Deux groupes de clients : “Cluster” et “Référentiel Groupe”	43
5.4.2	Cadre de reprise d’un client	43
5.5	Une nouvelle codification client	44
5.5.1	Présentation de la nouvelle codification	44
5.6	Processus de migration de Comp@ssV2 à Comp@ssV4	45
5.7	Explication du travail effectué	46
5.7.1	Définition de l’objectif	46
5.7.2	L’algorithme de Ratcliff et Obershelp	47
5.7.3	Objectifs de notre méthode	48
5.7.4	La problématique du seuil de détection	48
5.7.5	Principe de l’outil que nous avons développé	49
5.7.6	Les principales difficultés que j’ai rencontrées	49
5.7.7	Quel futur pour cet outil	50
6	Conclusion	51
	Rédaction du Rapport en langage LaTeX	53

Table des figures

1.1	Valeo aujourd'hui - Les chiffres clés - [1]	9
1.2	4 Business Groups et Valeo Service - [2]	10
1.3	Les 4 Product Groups rattachés à CDA - [3]	11
1.4	Les clusters informatiques - Europe de l'ouest	13
2.1	Fiche de poste	15
3.1	Représentation du Cloud - [4]	16
3.2	IaaS vs PaaS vs SaaS - [5]	18
4.1	<i>Schéma de l'architecture Cordys chez Valeo</i>	23
4.2	<i>Interface Standard d'une application Cordys - Menu "Change Application"</i>	24
4.3	<i>Interface Standard d'une application Cordys - Onglet "My Page"</i>	24
4.4	<i>Interface Standard d'une application Cordys - Onglet "Setup" - Vue Administrateur</i>	25
4.5	<i>Interface Standard d'une application Cordys - Onglet "Setup" - Vue Utilisateur</i>	25
4.6	<i>Interface Standard d'une application Cordys - Onglet "Report"</i>	26
4.7	<i>Interface de développement de Cordys - Page D'accueil</i>	27
4.8	<i>Interface de développement de Cordys - Interface de définition des Application Object</i>	28
4.9	<i>Interface de développement de Cordys - Interface de définition des Formulaires</i>	30
4.10	<i>Interface de développement de Cordys - Exemple d'un Business Process</i>	32

5.1	Structure du système Comp@ssV2 - Pas de référentiel client	40
5.2	Structure du système Comp@ssV4 - Présence d'un référentiel client Groupe	40
5.3	Comp@ssV2 vs Comp@ssV4 - Nouvelle Architecture Client	41
5.4	Reprise des clients Comp@ssV2 - Referentiel Groupe vs Cluster	43
5.5	Nouvelle Codification Client du système Comp@ssV4 - Codification Unifiée	44
5.6	Processus de migration de Comp@ssV2 à Comp@ssV4	45

Introduction

Dans le cadre de mes études en *Génie Informatique* à l'**Université de Technologie de Compiègne** (<http://www.utc.fr>), j'ai réalisé un stage de 6 mois Assistant Ingénieur chez l'équipementier automobile Français : Valeo. J'ai travaillé à la direction des systèmes d'information du site d'Annemasse, dans le service informatique partagés entre plusieurs sites (Annemasse, Créteil, Abbeville, Mondeville, Ben Arous, Nevers).

Marc LAURENT, mon tuteur, est le manager de cette section.

Depuis déjà plusieurs années, Valeo s'est engagé dans une politique de standardisation de ses outils et méthodes dans le but d'améliorer les usages des nombreux sites et ainsi gagner en coût, délai et qualité.

C'est donc dans ce contexte de standardisation et que j'ai effectué mon stage, un environnement international marqué par la décentralisation des familles de produit Valeo.

Mes activités ont été très variées au cours des 6 mois de stage.

J'y ai mené principalement deux missions : administrateur, formateur et développeur de la plateforme cloud de service *Cordys Process Factory* (<http://www.cordys.com>) pour le périmètre cité plus haut, et finalement j'ai participé à la préparation du changement de version de l'ERP en place : **SAP**. Une mise à jour biaxiale portant sur la version logicielle et sur l'architecture de l'ERP qui se dirige vers un système unifié, hiérarchisé et centralisé pour tout Valeo. Le *Kick-Off* de ce projet devrait avoir lieu avant l'été 2014.

S'ajoutent à cela des tâches ponctuelles de support informatique, la participation à la réalisation d'un DRP sur l'ensemble des infrastructures informatiques du site Valeo de Nevers, j'ai également aidé à la bonne conduite d'un audit de sécurité sur le site Valeo d'Annemasse et finalement j'ai collaboré avec les équipes en charge de la réalisation d'une migration des lignes WAN et ISP en vue d'une augmentation du débit tout en respectant la vision sécurité qu'apporte le DRP, une vision vitale pour l'entreprise.

Présentation de l'entreprise

1.1 Valeo, un équipementier Automobile

1.1.1 Présentation de l'entreprise

Valeo est un des leaders mondiaux dans l'activité d'équipementier automobile.

Depuis 1923, lors de création de la SAFF (*Société Anonyme Française de Ferodo*) par Eugène Buisson, Valeo a toujours œuvré pour viser l'excellence.

A partir de 1962 avec le rachat de la SOFICA (*Société de Fabrication Industrielle de Chauffage et d'Aération*), l'entreprise s'ouvre vers l'Europe. C'est le début de multiples rachats et créations d'usines en Europe (particulièrement en Espagne et en Italie).

C'est finalement en 1980 que le nom de Valeo (du latin : *"Je vais bien"*) est adopté par le groupe.

Cette même année, Valeo inaugure son premier site aux États Unis, suivis en 1982 du Mexique.

L'année de mon stage, 2013, fut celle des 90 ans de Valeo, et toujours le même objectif :

"Accompagner les constructeurs et les automobilistes pour rendre les véhicules plus propres, performants et plus sûrs"

Jacques Aschenbroich - Communiqué des 90 ans de Valeo



Figure 1.1 – Valeo aujourd'hui - Les chiffres clés - [1]

1.1.2 La structure interne de Valeo.

Valeo est divisée en quatre "Business Groups", à cela s'y ajoute un pôle de deuxième monte : Valeo Service.

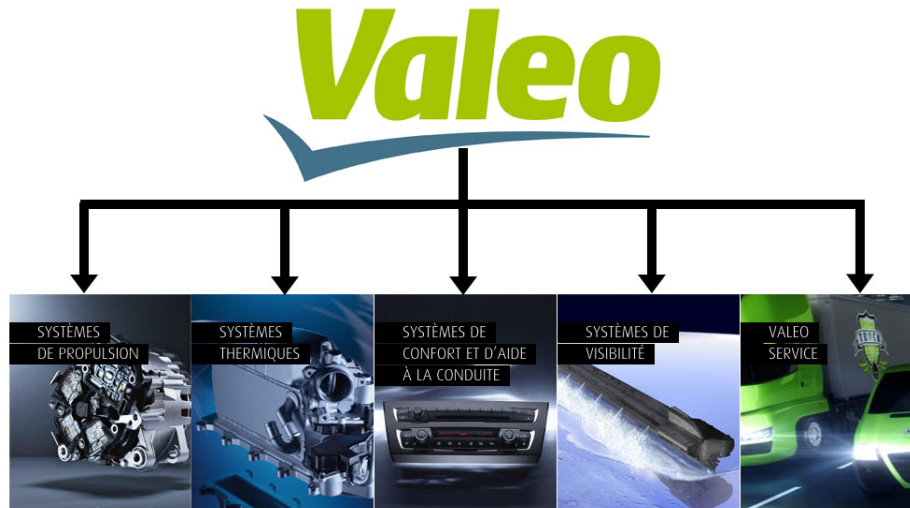


Figure 1.2 – 4 Business Groups et Valeo Service - [2]

Le Pôle Systèmes de Propulsion développe des solutions de propulsion innovantes visant à réduire la consommation de carburant et les émissions de CO₂.

Le Pôle Systèmes Thermiques développe et fabrique des systèmes, des modules et des composants assurant la gestion de l'énergie thermique du groupe motopropulseur ainsi que le confort de chaque passager dans l'habitacle.

Le Pôle Systèmes de Confort et d'Aide à la Conduite développe des systèmes d'interface entre le conducteur, le véhicule et son environnement, contribuant à l'amélioration du confort et de la sécurité.

Le Pôle Systèmes de Visibilité conçoit et produit des systèmes innovants qui assurent au conducteur une parfaite visibilité, contribuant ainsi à sa sécurité et à celle de ses passagers.

Valeo Service fournit des pièces de rechange aux constructeurs automobile et au marché de la Rechange indépendante. Il propose à tous les réseaux de la rechange dans le monde une large gamme de produits et services.

Source : <http://www.valeo.com/nos-activites/>

1.1.3 La structure interne du Business Group : CDA

J'ai effectué mon stage au sein du site d'Annemasse, site qui est rattaché au Business Group : Confort and Driving Assistance (CDA).

LES CHIFFRES CLÉS DU PÔLE À FIN DÉCEMBRE 2012



Figure 1.3 – Les 4 Product Groups rattachés à CDA - [3]

La mission du pôle systèmes de confort et d'aide à la conduite est de développer des systèmes d'interface entre le conducteur, le véhicule et son environnement tout en contribuant à l'amélioration du confort et de la sécurité.

La conduite intuitive est le credo du pôle, grâce à quatre activités complémentaires :

- Interaction ergonomique et simple du conducteur avec le véhicule
- Agilité de conduite par une meilleure perception de l'environnement
- Connectivité
- Accès personnalisé et sécurisé au véhicule

Source : <http://www.valeo.com/nos-activites/>

1.2 L'IS Shared Service Center - CDA France

J'ai effectué mon stage au sein du **Centre Informatique de Service Partagé** (Information System Shared Service Center : IS SSC) du périmètre CDA France - Tunisie sous la tutelle de Marc LAURENT, le manager de cette section.

Valeo a structuré les ressources informatiques en structure que l'on appelle "**Cluster**".

Chaque Cluster informatique est en charge d'un certain nombre de sites, à la fois dans leur supervision, suivis de projets, que dans l'applicatif (ERP, Business Intelligence, Office Automation, DRP, Barcoding ...).

La partie infrastructure (réseau et serveur) ne fait pas partie des attributions de l'ISSS sauf en ce qui concerne l'ERP ; cette partie est supervisée par un autre service et externalisée au maximum dans une optique de réduction des coûts.

Ainsi notre cluster se compose des sites suivants :

- **Annemasse** (mon lieu de stage).
- **Nervers CIE**
- **Abbeville**
- **Mondeville**
- **Ben Arous** (*Tunisie*)
- **Créteil Europarc**

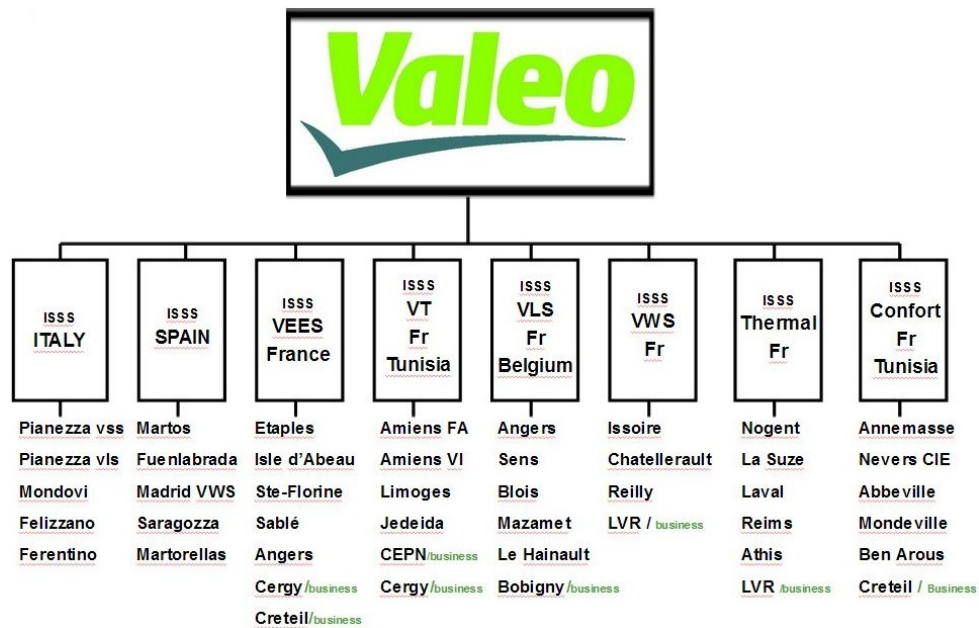


Figure 1.4 – Les clusters informatiques - Europe de l'ouest

Présentation du Sujet de Stage

2.1 Présentation du sujet de stage

Sujet :

Développement d'un système workflow en cloud computing. Accompagnement d'un audit de sécurité

Description :

Deux parties composent le stage :

Au sein du centre de service partagés d'Annemasse composé de 5 sites (Abbeville, Mondeville, Créteil, Ben Arous, Annemasse), le développement d'un ensemble d'applications sous deux environnements en cloud computing (Google Apps et Google Script, et Cordys).

Pour cette partie, les objectifs sont :

1. Déployer un plan d'action pour qualifier, planifier, développer et mettre en production un ensemble d'applications intégrant du workflow sur un environnement Cordys (processus de gestion de workflow)
2. Utiliser et développer les évolutions des modèles de Google Site et Google Scripts sur un parc applicatifs existant
3. Planifier et ordonnancer les lancements de mise en productions avec les Business Owners

Au sein du site d'Annemasse, aider à la préparation d'un audit de sécurité informatique du site organisé par le Groupe. Pour cette partie, les objectifs sont :

1. Le développement de scripts de contrôle au niveau du réseau
2. La préparation d'une documentation
3. L'accompagnement dans le projet dans les démarches sécuritaires (opérations techniques, et communications aux users)

L'ensemble des actions menées durant le stage devront être réalisées dans une démarche industrielle devant respecter les contraintes Coûts / Qualité / Délais.

Des qualités de communication sont requises (français maîtrisé/ anglais parlé) car les contacts avec la Direction sont fréquents. Des déplacements pourront être envisagés dans le cadre du stage.

2.2 Fiche de Poste



Job Profile

Job Title: Office Automation Analyst Job Code : MIS200	Entity : Country
	Department: IS Shared Service
Hierarchical Manager Title : IS Shared Service Manager	Functional Manager Title : Group Office IS Director

Mission

<p>The job aims at satisfying two main missions:</p> <ul style="list-style-type: none"> ▪ Improve business efficiency & productivity (on a personal, team & company levels) through a better usage of Office automation solutions from productivity suite to advanced VeGA services: Google Apps products, Cordys Process Factory, Enterprise Directory Services... ▪ Implement and support the migration processes to new solutions such as Kinesys and Lotus migration to Google Apps & Cordys platform.
--

Responsibilities

<p>Ensure successful deployment of administrative processes within plants</p> <ul style="list-style-type: none"> ▪ Deploy standard administrative processes & Office applications within plants. ▪ Support local IS teams and/or outsourcers (level 2 for usage and functional issues). 	
<p>Lead the continuous improvement of office automation applications</p> <ul style="list-style-type: none"> ▪ Collect & share success stories & best practices. ▪ Improve local administrative & collaborative processes. ▪ Promote innovation (Office automation perimeter), ▪ Train, communicate in a 'continuous improvement' mode. 	
<p>Provide support to Group projects and satisfy specific feature request from management</p> <ul style="list-style-type: none"> ▪ Participate in Group projects (trusted testing for new services and new features, working groups), ▪ Manage feature requests (triage, use cases and business cases formalisation). 	
People Management: YES <input type="checkbox"/> NO <input checked="" type="checkbox"/> If People Management, how many employees are managed?	Budget Management: YES <input type="checkbox"/> NO <input checked="" type="checkbox"/>

Figure 2.1 – Fiche de poste

Le Cloud Computing

3.1 Qu'est ce que le Cloud Computing ?

Le Cloud Computing (ou "Informatique dans les Nuages" en français) est un secteur de l'informatique en plein essor depuis quelques années. Un de ses concepts fondateurs est la mise à disposition de technologies et services à la demande : "*as a service*".

La vision est d'offrir aux utilisateurs un service à la demande où celui ci ne paie que sa consommation (une analogie avec le service d'eau, d'électricité, de gaz peut être établie en France contrairement à certaines villes comme Moscou où l'on paie un forfait à l'année pour le gaz en illimité).

Cette technologie est actuellement en plein essor. De nombreux acteurs misent sur cette nouvelle manière d'envisager l'informatique. Parmi lesquels nous retrouvons Google, Amazon, Microsoft, IBM, Salesforce, DropBox, Apple, ... Des technologies à la fois orientés grand public et entreprises sont développés. Je citerai par exemple le Google Drive et Google Apps qui sont en fonction chez Valeo. De plus en plus de domaines sont à présent concernés : *Emails, Traitement de Texte, Stockage de Données, Jeux ...*

Le "Cloud" se base sur une faculté native à s'adapter de manière élastique aux besoins des utilisateurs. À fournir des services en temps réel ou en ponctuel, à forte ou faible capacité (en terme de débit de connexion et de capacité de calcul). Ce sont tous ces facteurs qui font du "Cloud" une valeur montante de l'informatique.

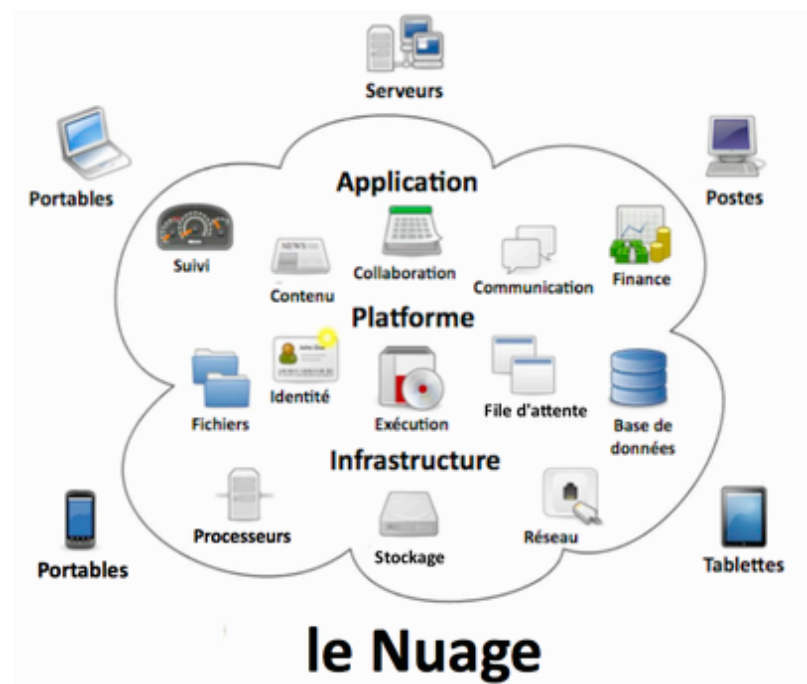


Figure 3.1 – Représentation du Cloud - [4]

3.2 IaaS vs PaaS vs SaaS vs STaaS

Quand on parle d'un "Service Cloud", il y a trois catégories possibles :

— Infrastructure as a Service (IaaS) :

Dans ce type d'architecture, **le fournisseur** du service s'engage à livrer et maintenir le serveur, son OS (virtuel ou non), le stockage et le réseau nécessaire.

Il reste à la charge **du client** la gestion de la *plateforme* et des diverses applications qui fonctionnent sur cette dernière, les bases de données et les différents logiciels installés.

Ce type de service est grandement utile aux entreprises désireuses de revendre des services Cloud type SaaS ou PaaS de louer des infrastructures hautement flexibles. Pour ne citer que quelques exemples voici quelques fournisseurs de IaaS : *Cloud Power, DotRiver, Amazon EC2, Windows Azure, Rackspace, ...*

— Platform as a Service (PaaS) :

Dans ce type d'architecture, **le fournisseur** du service s'engage à livrer et maintenir tout ce qui compose l'IaaS. A cela s'ajoute en plus la partie plateforme cloud, les bases de données, et les logiciels nécessaires au fonctionnement de la plateforme ou de l'administration du serveur.

Il ne reste à la charge **du client** que la gestion des *applications* (et leur développement au besoin) sur la plateforme cloud fournie.

Ce type de service est généralement destiné aux utilisateurs finaux, et sont utilisés par de plus en plus de personnes ayant des besoins spécifiques et ne voulant déployer de gros moyens à chaque nouveau projet (petit ou grand) : On retrouvera dans les plateformes : *Force.com pour Salesforce, Google AppEngine, Cordys Process Factory* (sur lequel j'ai travaillé), Microsoft Azure ...

— Software as a Service (SaaS) :

Le Software as a service (SaaS) est l'ultime modèle de Cloud, où **le fournisseur** Cloud maintient absolument tout et développe les applications.

Le **client** lui est simple "*utilisateur*" des applications.

Ainsi nous retrouvons les applications clouds suivantes en tant que SaaS : *Google Document, Google Spreadsheet, ...*

— Storage as a Service (STaaS) :

Très similaire au SaaS, seulement cette fois-ci le client ne veut pas une application mais un espace de stockage.

Services Connus : Google Drive, Dropbox, iCloud, Microsoft Skydrive, ...

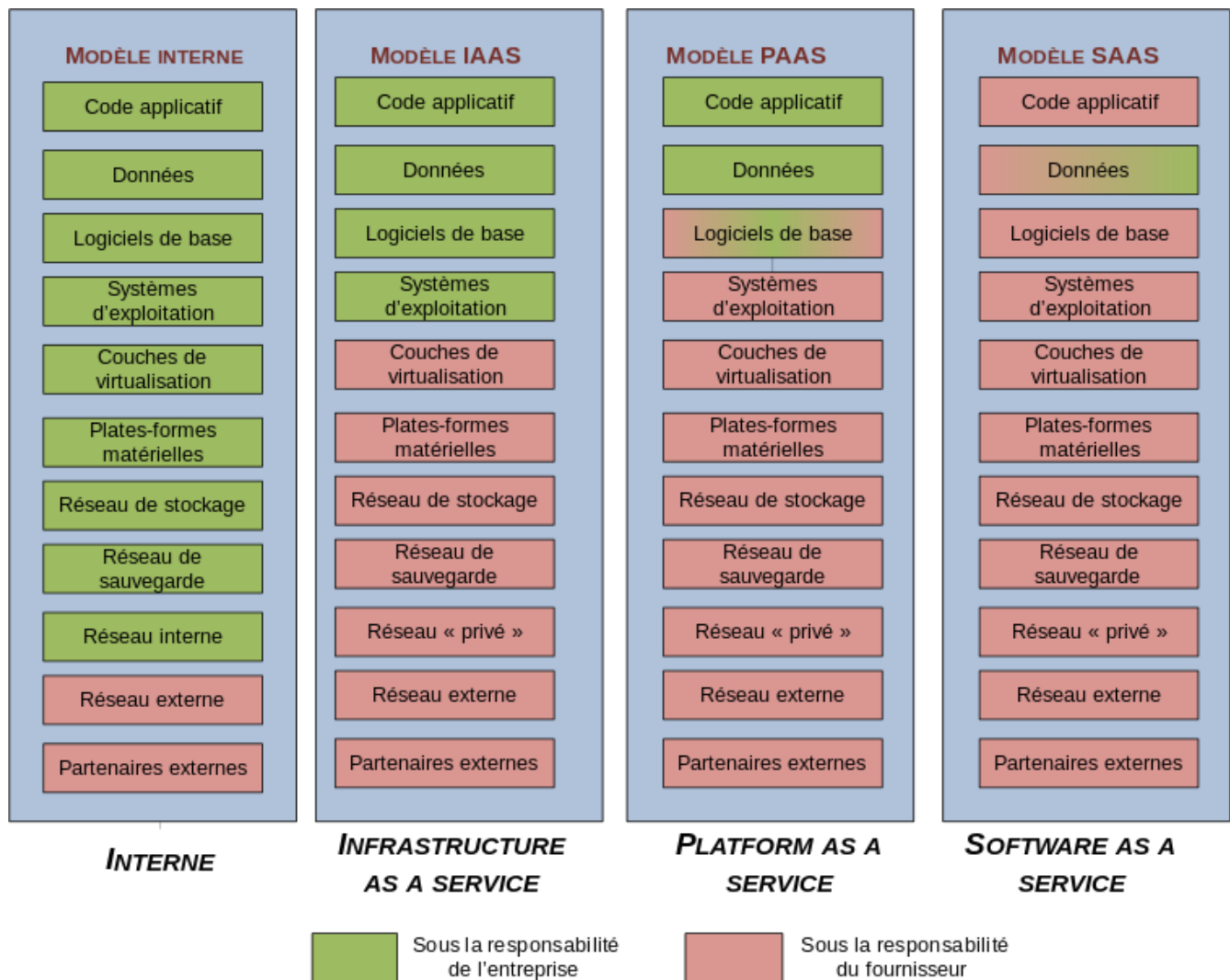


Figure 3.2 – IaaS vs PaaS vs SaaS - [5]

Cordys Process Factory

Avant Propos

Cordys Process Factory fut un des points centraux de mon stage. Comme il ne m'est pas possible de décrire avec précision chacun de ses points, j'ai choisi de détailler mon travail dans la partie qui pour moi présente le plus d'intérêt.

Cordys Process Factory étant une plateforme propriétaire et bénéficiant d'une interface de *développement au clic*, je m'attacherai à en décrire avec minutie le fonctionnement, tous les outils qui sont mis à disposition sur la plateforme et leurs interactions avec autant que possible des références au travail que j'ai pu mené à Valeo.

4.1 Une plateforme Applicative "Cloud"

4.1.1 Qu'est ce que Cordys Process Factory ?

Cordys Process Factory est un service orientée *Cloud Computing*. Cette solution Cloud est de type PaaS : Plateforme as a Service. Elle permet de développer des applications répondant à une demande de "*Business Workflow*". C'est un besoin critique dans un environnement comme celui de Valeo.

Les utilisateurs classiques, tout comme les développeurs, peuvent utiliser la plateforme pour **composer** des applications de manière simple et rapide en ayant reçu une brève formation sur l'outil(**sans connaissances en programmation informatique** ou presque).

La plateforme est également capable de s'interfacer avec les différents systèmes existants et les *Cloud Services* actuellement en fonction dans l'entreprise. Un point qui fut très important dans le choix de cette plateforme.

4.1.2 Lotus Notes vs Cordys Process Factory

Lotus Notes (solution logicielle développée par **IBM**) et Cordys Process Factory (solution Cloud développée **Open Text**) ont un objectif commun : proposer une réponse au besoin de “*Business Processing*” au sein d’une entreprise.

Cependant elles diffèrent sur l’architecture (serveurs et réseau) nécessaire à son fonctionnement :

- Lotus Notes se base sur une architecture type logiciel Client / logiciel Serveur, les serveurs sont inter-connectés entre eux et “*répliquent*” certaines de leur données à intervalle de temps régulier. Ce fonctionnement est coûteux , il demande une infrastructure imposante ainsi qu’une administration au jour le jour.
Plus gênant, comme il est difficile d’avoir une vue d’ensemble du système actuel, la quantité d’application dupliquée ne cesse d’augmenter, ce qui induit des coûts de fonctionnement de plus en plus élevés.
- Cordys Process Factory propose une réponse orientée “*Cloud Computing*” à ces problématiques, nous pouvons donc faire *abstraction* de l’architecture réseau et serveur nécessaire à son fonctionnement (gérée par le fournisseur du service). La plateforme est accessible via un navigateur web, nul besoin de paramétrer son ordinateur ou d’installer un quelconque logiciel.
Le choix de Cordys Process Factory est donc cohérent avec la politique actuelle du groupe Valeo qui cherche à *externaliser* la gestion de la partie infrastructure et l’administration des diverses solutions informatiques.

4.1.3 Le contexte dans lequel s’insère Cordys Process Factory

Valeo a fait le choix de se baser en grande partie sur la suite Google Apps pour un maximum de services et besoins bureautiques. Ainsi nous utilisons de manière non exclusive :

- Gmail
- Google Drive
- Google Apps (SpreadSheet, Docs, Presentation, Scripts, Sites)
- Google Agenda
- Google AppEngine

En complément nous utilisons un LDAP, commun à tous les sites Valeo du monde, interfacé avec la suite applicative de **Google**. Valeo appelle cet ensemble de systèmes inter-opérants : VeGA (*Valeo empowered by Google Apps*).

Et *Cordys Process Factory* dans tout ça ?

Fort de sa position de partenaire *Google Entreprise*, Cordys Process Factory permet une inter-connection aux services Google Apps et le LDAP de Valeo : *l'Entreprise Directory*.

Ainsi les *Businness Process* et *Businness Workflows* peuvent s'appuyer sur les services et applications déjà en place chez Valeo afin de réduire les coûts de mise en place et de développement.

4.2 Présentation du fonctionnement de la Plateforme de Service

4.2.1 Architecture de la plateforme

La plateforme Cordys fonctionne en trois mandants.

- Mandant de Production
- Mandant de Test (ou Acceptance)
- Mandant de Développement

Les utilisateurs "classiques" n'ont accès qu'au mandant de production, c'est ce dernier le plus important : il contient les données Business de l'entreprise.

Les développeurs d'applications sur la plateforme Cordys peuvent accéder à l'environnement de test qui est une réplication complète de l'environnement de production (en terme de services et logiciels en place). Il est essentiel de tester une application dans ce mandant avant de la déployer en production de sorte à vérifier la compatibilité avec le futur environnement. Cette étape est absolument nécessaire du fait de l'impossibilité d'altération ou suppression des données Business et stratégiques dans ce mandant.

Concernant le mandant de développement, c'est le seul mandant qui n'est pas rattaché aux services Valeo, Google et LDAP. Ce dernier est uniquement là pour développer et tester les applications de manière succincte avant de les soumettre à validation.

Nous suivons donc un Workflow de développement de type DTAP : Development, Testing, Acceptance and Production :

1. Une fois que le développeur pense que l'application est prête, elle est copiée dans l'environnement de test pour vérifier qu'elle fonctionne comme attendue.
Les tests ne sont pas normalisés comme ils devraient l'être dans un Workflow de type DTAP. Il appartient à chaque développeur de s'assurer du fonctionnement de son application.
2. Une fois les tests positifs, l'application est passée en revue avec le "Business Owner" de la future application (celui qui sera responsable de cette application et des données Business qui y transiteront). Ce dernier vérifiera que l'application correspond bien à son besoin.
3. Après l'acceptation de l'application par le "Business Owner", l'application est déployée en production et rendue accessible.

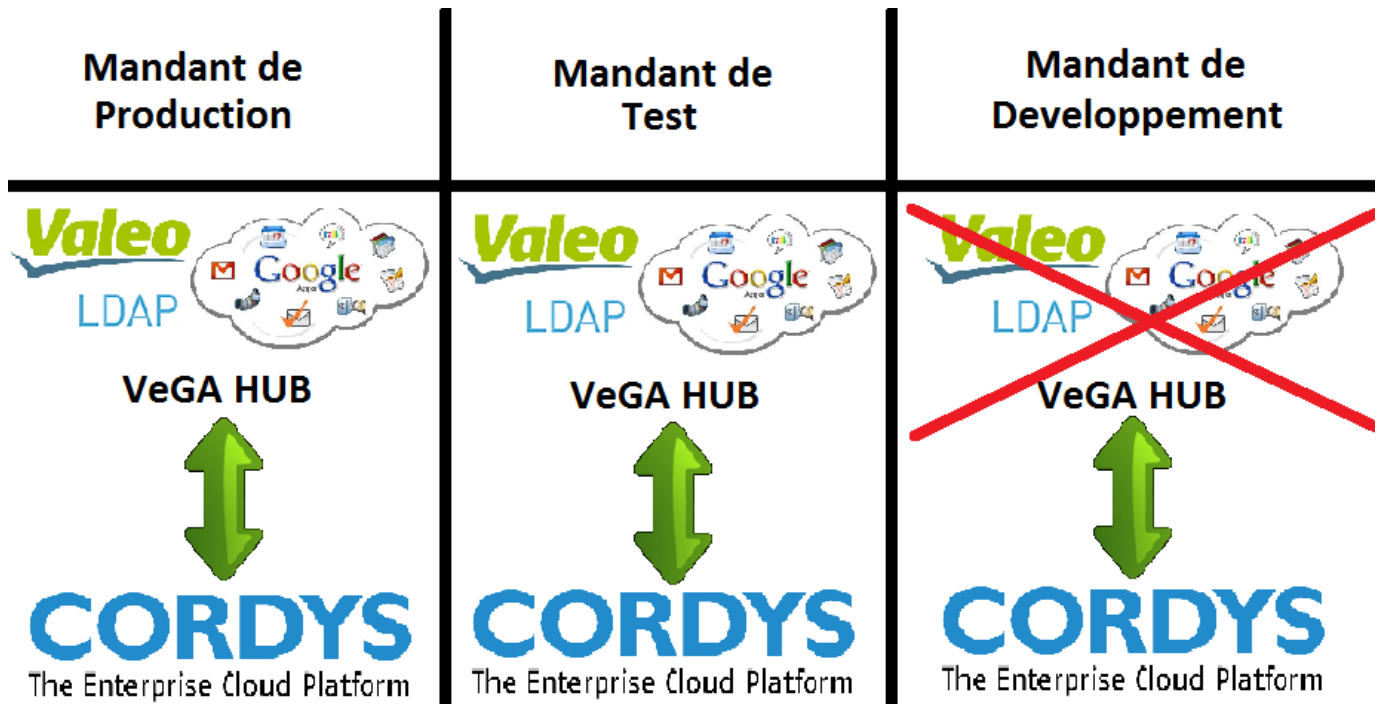


Figure 4.1 – Schéma de l'architecture Cordys chez Valeo

4.2.2 Présentation de l'interface standard d'une application

L'interface d'une application Cordys est relativement identique quelque soit l'application. Ainsi, quel que soit l'application, deux onglets sont systématiquement disponibles : *My Page* & *Setup*. Le nom de l'application se situe en haut à gauche, se qui constitue un bon repère visuel. En plus de cela, un menu déroulant "Change Application", permet d'afficher la liste entière des applications auxquelles vous avez le droit d'accéder.

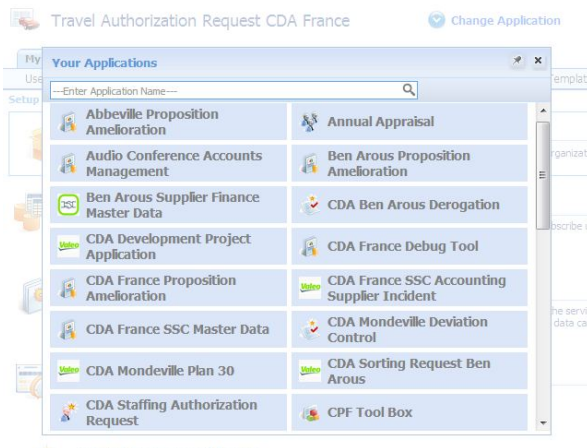


Figure 4.2 – Interface Standard d'une application Cordys - Menu "Change Application"

L'onglet My Page

La vue par défaut d'une application est l'onglet : "My Page". Il regroupe les dernières données relatives à l'utilisateur.

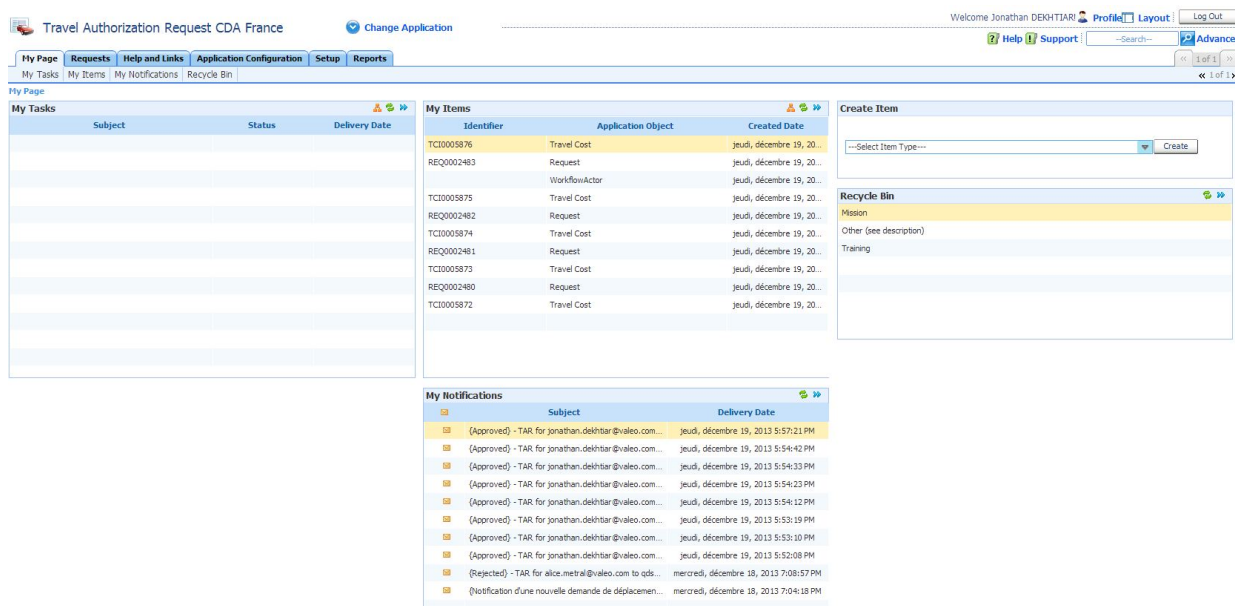


Figure 4.3 – Interface Standard d'une application Cordys - Onglet "My Page"

L'onglet Setup

L'onglet Setup varie en fonction des droits d'accès. Un utilisateur aura accès uniquement à la personnalisation de son interface (de manière assez restreinte) et l'administrateur pourra, lui, accéder à plusieurs rubriques permettant l'import de données dans l'application, l'analyse des processus en cours d'exécution, les tâches actuellement en attentes (et les réaffecter si nécessaire), la liste des utilisateurs de l'application et leurs droits respectifs (en lecture et en modification), ainsi que d'autres fonctions moins utiles comme les modèles d'emails ou l'état des nombres auto-incrémentés.

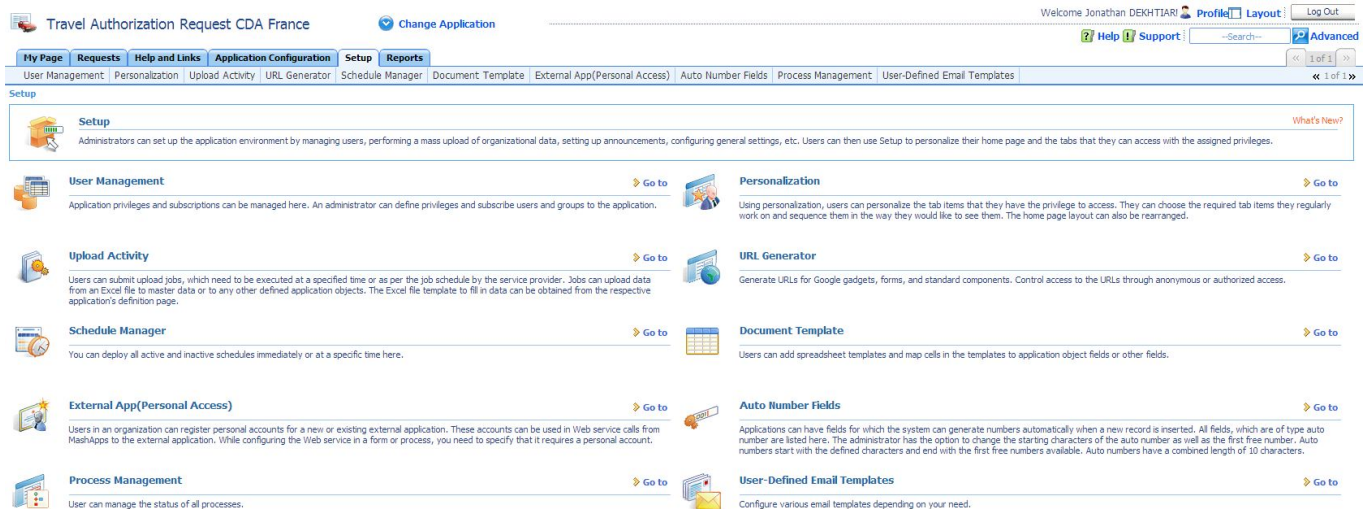


Figure 4.4 – Interface Standard d'une application Cordys - Onglet "Setup" - Vue Administrateur

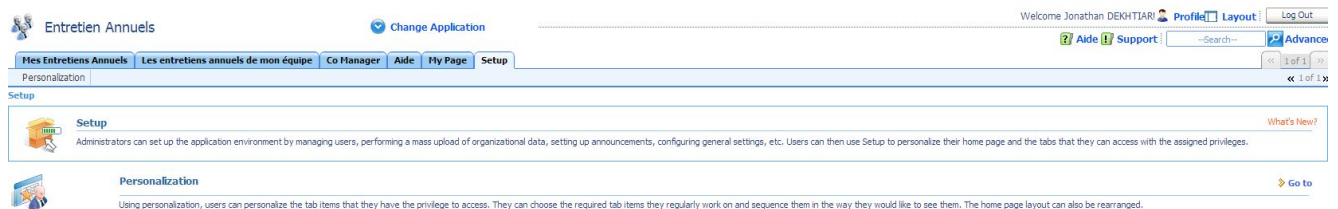


Figure 4.5 – Interface Standard d'une application Cordys - Onglet "Setup" - Vue Utilisateur

L'onglet Report

L'onglet Report est automatiquement disponible pour chaque application, cependant l'administrateur peut choisir ou non de le faire apparaître dans l'application et/ou de le rendre disponible uniquement pour une partie des utilisateurs ou pour tout le monde en modifiant les droits d'accès.

Les Reports sont à "paramétrer" en développement pour qu'ils apparaissent en production de manière systématique. Sinon "*l'instant Report*" permet d'extraire des données de manière rapide et flexible. Il est cependant plus long à effectuer qu'un Report automatique et nécessite la connaissance du nom des champs que l'on veut extraire.

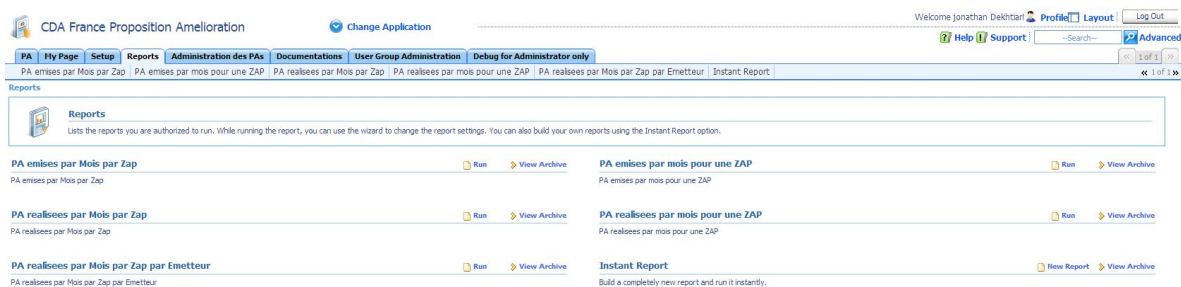


Figure 4.6 – Interface Standard d'une application Cordys - Onglet "Report"

4.2.3 Présentation de l'interface développeur de Cordys

La page d'accueil de la plateforme de développement

La page d'accueil de la plateforme de développement est assez sobre, on y retrouve les différentes sections sur la gauche et une vue "graphique" au centre expliquant à quoi sert cet onglet. Toujours dans l'optique de permettre à un utilisateur, sans bagage informatique, de pouvoir mettre en place simplement et rapidement en place une application.

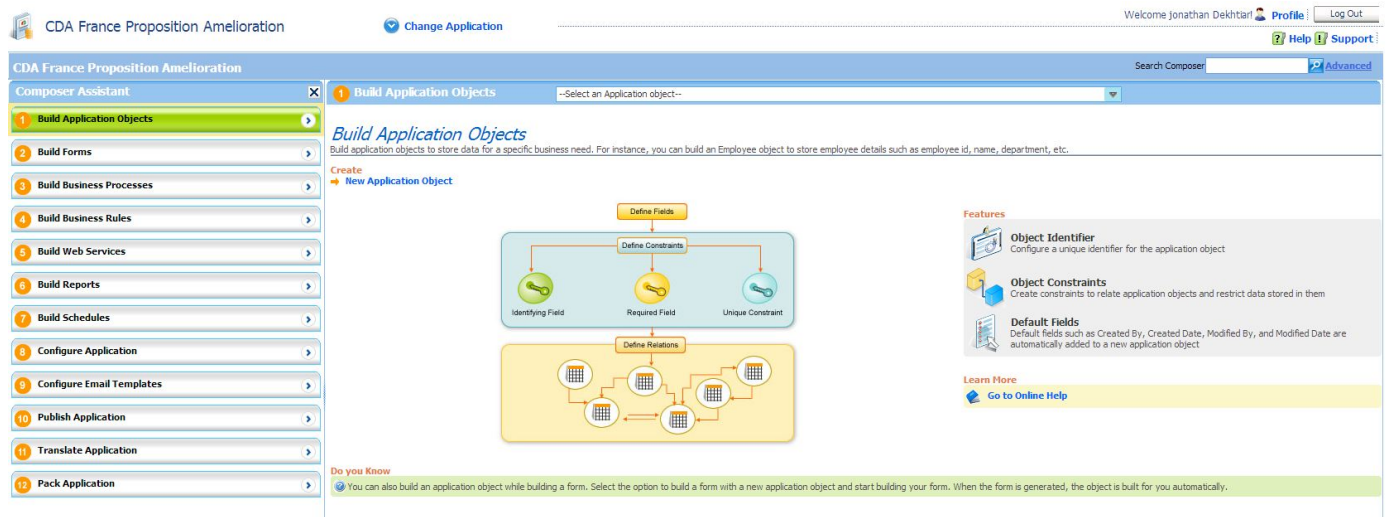


Figure 4.7 – Interface de développement de Cordys - Page D'accueil

L'onglet "Application Object"

Un *Application Object* est l'équivalent de ce qu'on appelle en programmation orientée objet : **un objet**.

Ces objets sont la base de l'application, ils sont à définir selon les besoins et donc selon le cahier des charges qui a été établi au préalable.

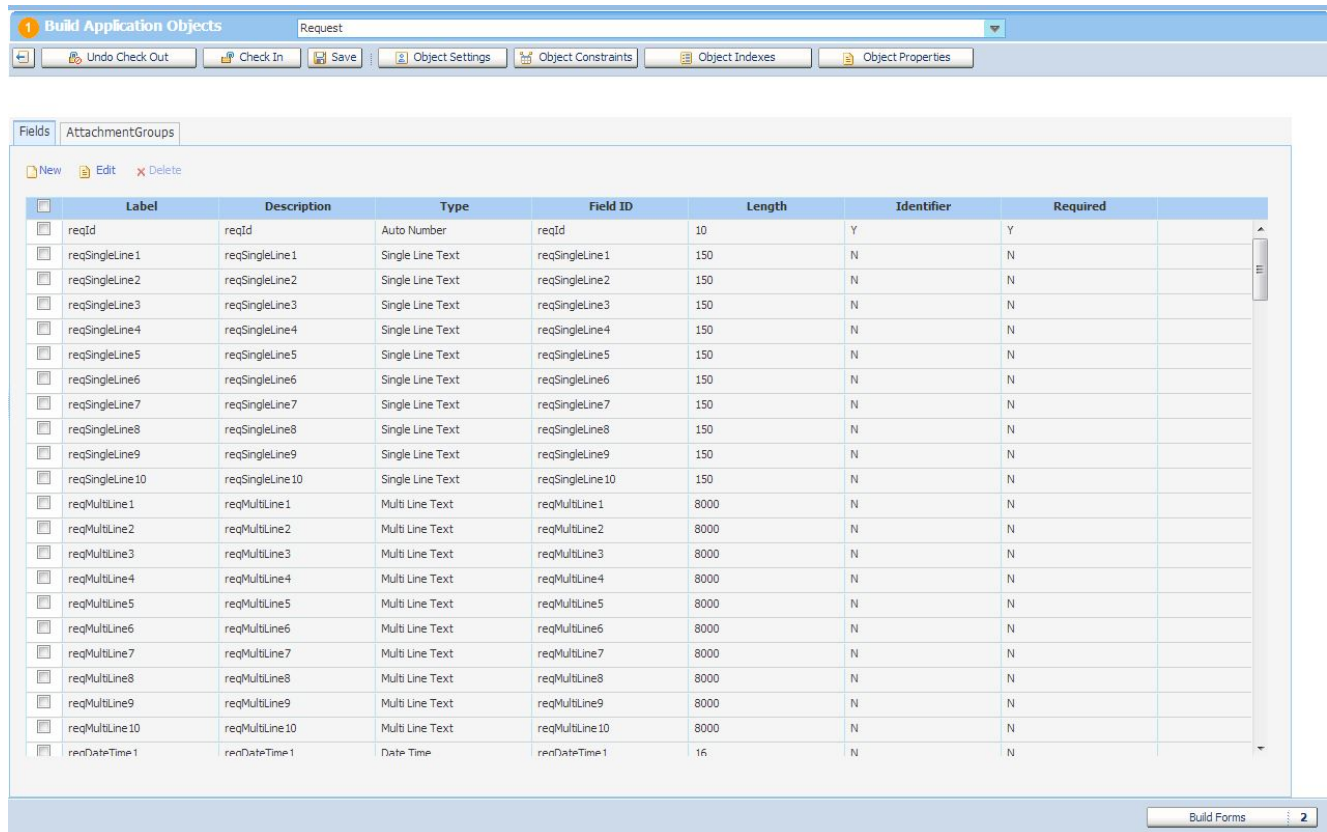
Des diagrammes UML sont donc à prévoir sur les applications lors de la définition du projet.

Les *Application Objects* permettent de stocker les données relatives à l'entreprise. Par exemple, les dépenses de l'entreprise dans une application dédiée à la validation des factures ou au remboursement des frais des employés.

Afin de stocker des données dans un *Application Object*, il est nécessaire d'ajouter donc des champs relatifs à l'utilisation (Exemple : FactureID, UserName, ApproverName, ...).

Lors de la création de l'*Application Object*, il est possible de définir la clé primaire, des index afin de faciliter et accélérer la recherche et des contraintes (sur un ou plusieurs champs).

Tous les *Application Object* sont stockés dans l' "*Application Object Repository*". Il est ainsi possible de les modifier, et de les supprimer par l'intermédiaire de ce répertoire.



	Label	Description	Type	Field ID	Length	Identifier	Required
<input type="checkbox"/>	reqId	reqId	Auto Number	reqId	10	Y	Y
<input type="checkbox"/>	reqSingleLine1	reqSingleLine1	Single Line Text	reqSingleLine1	150	N	N
<input type="checkbox"/>	reqSingleLine2	reqSingleLine2	Single Line Text	reqSingleLine2	150	N	N
<input type="checkbox"/>	reqSingleLine3	reqSingleLine3	Single Line Text	reqSingleLine3	150	N	N
<input type="checkbox"/>	reqSingleLine4	reqSingleLine4	Single Line Text	reqSingleLine4	150	N	N
<input type="checkbox"/>	reqSingleLine5	reqSingleLine5	Single Line Text	reqSingleLine5	150	N	N
<input type="checkbox"/>	reqSingleLine6	reqSingleLine6	Single Line Text	reqSingleLine6	150	N	N
<input type="checkbox"/>	reqSingleLine7	reqSingleLine7	Single Line Text	reqSingleLine7	150	N	N
<input type="checkbox"/>	reqSingleLine8	reqSingleLine8	Single Line Text	reqSingleLine8	150	N	N
<input type="checkbox"/>	reqSingleLine9	reqSingleLine9	Single Line Text	reqSingleLine9	150	N	N
<input type="checkbox"/>	reqSingleLine10	reqSingleLine10	Single Line Text	reqSingleLine10	150	N	N
<input type="checkbox"/>	reqMultiLine1	reqMultiLine1	Multi Line Text	reqMultiLine1	8000	N	N
<input type="checkbox"/>	reqMultiLine2	reqMultiLine2	Multi Line Text	reqMultiLine2	8000	N	N
<input type="checkbox"/>	reqMultiLine3	reqMultiLine3	Multi Line Text	reqMultiLine3	8000	N	N
<input type="checkbox"/>	reqMultiLine4	reqMultiLine4	Multi Line Text	reqMultiLine4	8000	N	N
<input type="checkbox"/>	reqMultiLine5	reqMultiLine5	Multi Line Text	reqMultiLine5	8000	N	N
<input type="checkbox"/>	reqMultiLine6	reqMultiLine6	Multi Line Text	reqMultiLine6	8000	N	N
<input type="checkbox"/>	reqMultiLine7	reqMultiLine7	Multi Line Text	reqMultiLine7	8000	N	N
<input type="checkbox"/>	reqMultiLine8	reqMultiLine8	Multi Line Text	reqMultiLine8	8000	N	N
<input type="checkbox"/>	reqMultiLine9	reqMultiLine9	Multi Line Text	reqMultiLine9	8000	N	N
<input type="checkbox"/>	reqMultiLine10	reqMultiLine10	Multi Line Text	reqMultiLine10	8000	N	N
<input type="checkbox"/>	reqDateTime1	reqDateTime1	Date Time	reqDateTime1	16	N	N

Figure 4.8 – Interface de développement de Cordys - Interface de définition des Application Object

L'onglet "Forms"

Une fois un *Application Object* défini, il est possible de créer un ou plusieurs formulaires de manière à permettre l'instanciation de ces objets. Tous les champs relatifs à cet *Application Object* sont automatiquement ajoutés au formulaire, charge au développeur de choisir s'il désire tous les garder et de les agencer à sa convenance.

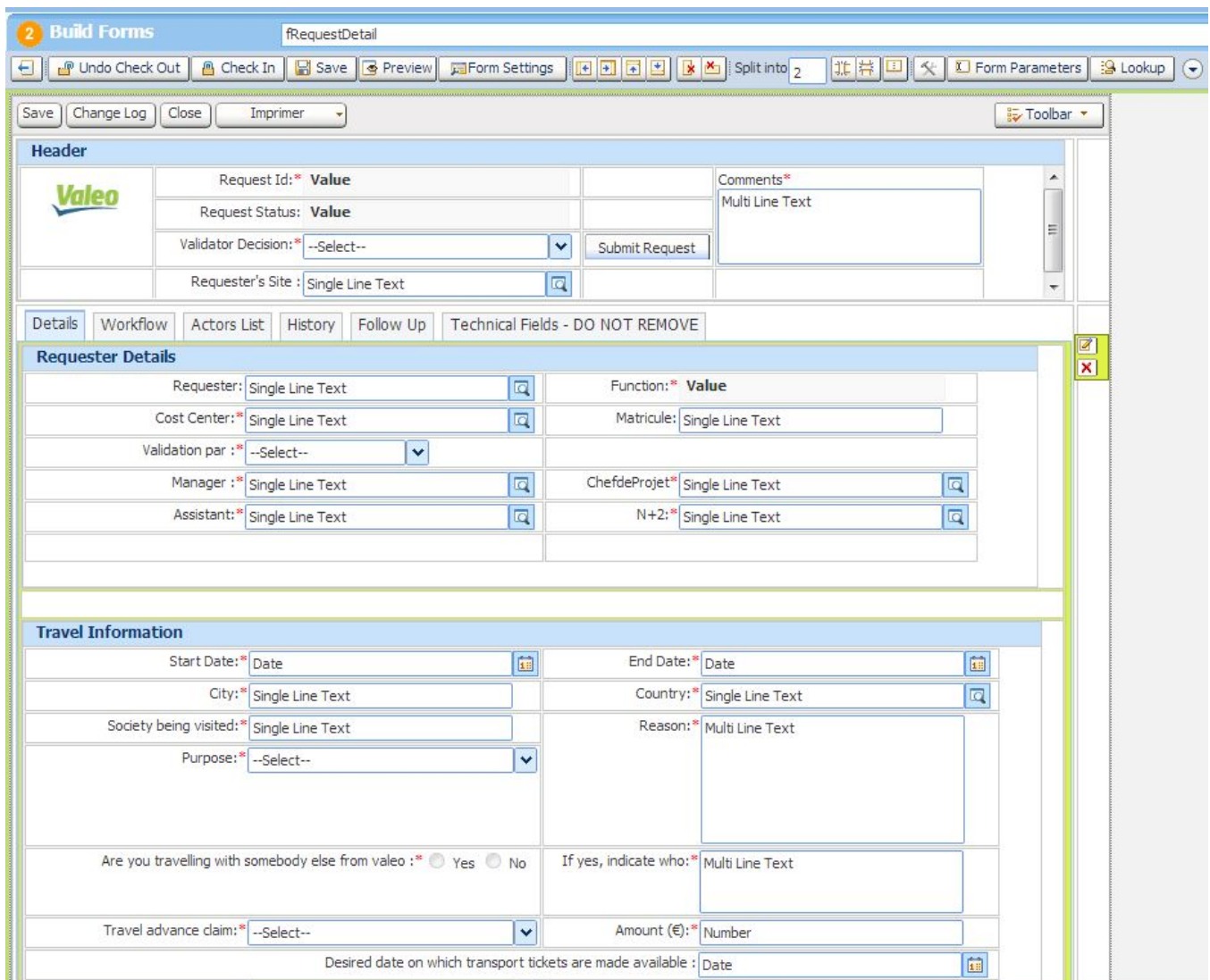
Il est également possible de créer un formulaire sans le rattacher à un *Application Object*. Dans ce cas, il est impossible de sauvegarder les informations entrées dans ce formulaire et celles affichées par les différents Web-Services appelés.

Différents outils sont à disposition afin de mettre en place le formulaire répondant exactement aux attentes :

- **Les Types de Sections** : Il est possible de créer différents types de section pour organiser les champs du formulaire. Ces sections portent le nom de : "Group Box", "Grid", et "Tab sections". Il est également possible de créer une section pour attacher un ou plusieurs fichiers et des éventuels commentaires.
- **Les types de Champs** : De nombreux types de champs sont à disposition des développeurs, le choix dépend alors du type d'information qui y sera stocké.
- **Le "Repertoire" de Web Services** : Tous les Web services de l'application sont stockés dans un répertoire. Il est ainsi possible d'ajouter des Web Services, basés sur des Application Objects et Business Process, à un formulaire directement depuis ce répertoire. Il est également possible d'ajouter des Web Services "externes" SOAP ou REST.
- **Les Paramètres de Formulaire** : Une des sources de flexibilité dans la création des formulaires vient de ce qu'on appelle les "Form Parameters" ou en Français : Les Paramètres de Formulaire. Ils permettent de définir quels champs seront les entrées ou les sorties d'un ou plusieurs Web Services.
- **Design adaptatif** : Les formulaires peuvent être créés de manière "flexible" et s'adapter à la taille de la fenêtre ou du design établi (*Responsive Design*). Les Sections, champs et Web Services peuvent être placés de manière quelconque dans le formulaire.
- **Les champs de type "Lookup"** : Il est possible de paramétrer des champs pour les associer à un "lookup". Cela leur permettra de **sélectionner** des données relatives à des Application Objects ou aux Utilisateurs (Groupes, Fonction, Email ...).
- **Les règles de Formulaire** : Il est possible de définir des conditions qui vont modifier le comportement d'un formulaire en utilisant ce qu'on appelle une "Règle de Formulaire". Il est ainsi possible d'assigner des valeurs, afficher des messages, cacher ou forcer l'affichage, bloquer ou débloquer un champ, une section.
- **Les vues** : Il est possible de créer des vues de manière à trier les données pour les formulaires de type tableau ou "Grid" en anglais.
- **L'éditeur de Script** : Ce dernier module permet aux développeurs plus chevronnés d'avoir un contrôle total de leur formulaire avec l'aide du Javascript et une petite sur-couche *made in Cordys*.

En dehors des fonctions citées plus haut, un système de versionnement est en place et permet d’afficher les dix dernières révisions de l’objet ou du formulaire via un système de “Check In” / “Check Out” dans le but d’améliorer l’efficacité et de réduire la maintenance. Il est également possible de modifier le formulaire de sauvegarder et de ne finalement rendre les modifications effectives que plusieurs jours après en effectuant le “Check In” au bon moment.

Il est possible de prévisualiser le formulaire pendant son édition. Cette fonction permet de vérifier le bon comportement pendant l’exécution (en particulier les scripts et règles de formulaires).



2 Build Forms fRequestDetail

Undo Check Out Check In Save Preview Form Settings Split into 2 Form Parameters Lookup

Save Change Log Close Imprimer

Header

Valeo Request Id: * Value Request Status: * Value Validator Decision: * --Select-- Submit Request Comments: * Multi Line Text Requester's Site: Single Line Text

Details Workflow Actors List History Follow Up Technical Fields - DO NOT REMOVE

Requester Details

Requester: Single Line Text Function: * Value Cost Center: * Single Line Text Matricule: Single Line Text Validation par: * --Select-- Manager: * Single Line Text ChefdeProjet: * Single Line Text Assistant: * Single Line Text N+2: * Single Line Text

Travel Information

Start Date: * Date End Date: * Date City: * Single Line Text Country: * Single Line Text Society being visited: * Single Line Text Purpose: * --Select-- Reason: * Multi Line Text Are you travelling with somebody else from valeo: * Yes No If yes, indicate who: * Multi Line Text Travel advance claim: * --Select-- Amount (€): * Number Desired date on which transport tickets are made available: * Date

Figure 4.9 – Interface de développement de Cordys - Interface de définition des Formulaires

L'onglet "Business Process"

C'est la notion clé du système Cordys, et d'ailleurs mon sujet de stage mentionne cet élément sous le nom de **Workflow**. Ce n'est donc pas un outil à proprement parlé mais plutôt une finalité : sa mise en place.

Un Business Process est donc un ensemble d'activités liées qui produisent un service de manière à atteindre les objectifs du business. Les "Business Processes" sont **la manière** d'effectuer une tâche spécifique au sein d'une organisation spécifique. Un Business Process définit, dans notre cas, un parcours de validation ou signature et ainsi aide les employés à interagir avec les différentes entités de l'entreprise.

Pour exemple, prenons l'approbation du remboursement d'une note de frais :

Un employé peut avoir des dépenses durant un voyage d'affaire et doit être remboursé. Ainsi il serait intéressant d'établir un **Business Process** décrivant le parcours de signature à effectuer de manière à donner l'autorisation de manière automatique à la finance pour le paiement de la note de frais :
Demande de remboursement => Validation du supérieur hiérarchique => Validation du chef de projet => Demande de remboursement validée.

Il peut être judicieux de notifier tout le monde dès l'acceptation ou le refus de la demande. Le remboursement peut ainsi être débloqué par le service financier.

Automatiser ce genre de processus permet d'une part de limiter les erreurs humaines et d'autre part d'en maximiser l'efficacité et la traçabilité.

Un Business Process se définit par :

- **Un But** : Un Business Process possède un but bien défini, qui est le besoin Business d'une action systématique.
- **Des données en "entrée"** : Un Business Process a besoin de données en entrée, que l'on appelle "ressources", pour fonctionner. Pour exemple, reprenons notre exemple de la validation des notes de frais. Voici quelques ressources : Date du début du Voyage, Date de fin du voyage, montant de la note de frais, Utilisateur demandeur ...
- **Des données en "sortie"** : "Un Business Process produit une ou plusieurs sorties qui fournissent des informations importantes pour l'entreprise.

Il est possible de publier un Business Process et de l'intégrer à un formulaire à la manière d'un Web Service. Il est ensuite possible de définir les "déclencheurs" qui vont appeler le Business Process à intervalle régulier ou sous certaines conditions.

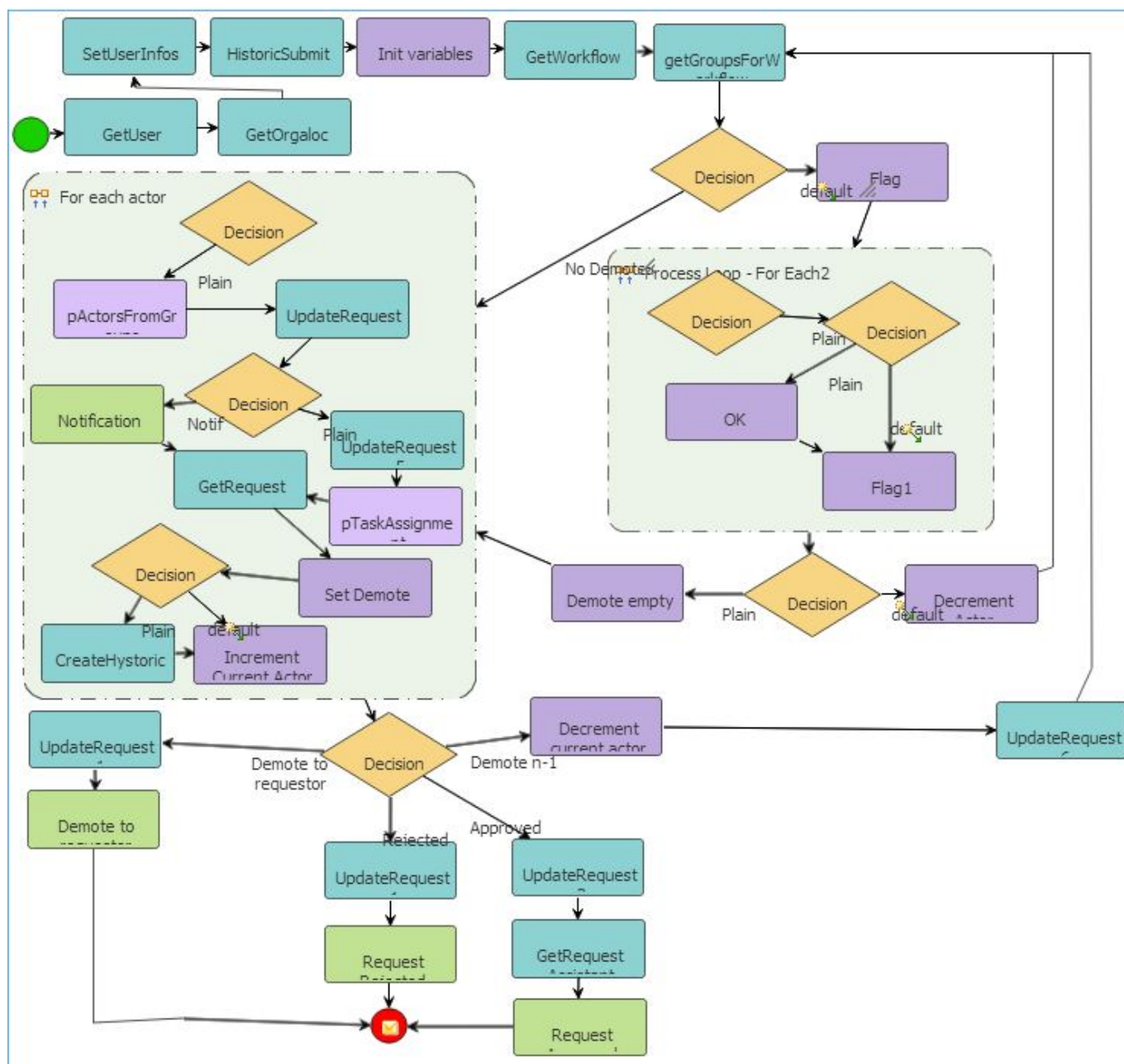


Figure 4.10 – Interface de développement de Cordys - Exemple d'un Business Process

L'onglet "Business Rules"

Les "Règles Business" sont des collections de procédures qui commandent une organisation et les services qui lui sont rattachés. Ils définissent les contraintes qui s'appliquent et aident à suggérer les activités clés du Business Process au moment opportun. Ils préparent et présentent les données de manière à faciliter les prises de décision critiques et stratégiques.

Une "règle Business" peut appeler un Business Process, envoyer des notifications, ou afficher des avertissements. Par exemple, une règle Business peut orienter le choix d'un Business Process ou d'un autre en fonction du montant de la note de frais pour reprendre notre exemple. (Si le montant de la note de frais est supérieur à 1000€ => Demander la validation du Directeur d'usine).

Pour résumer, une règle Business est un ensemble de conditions et d'actions. Structure algorithmique classique en *IF-THEN* et multiples *ELSE-IF*.

Voici le type d'actions que peut effectuer une "règle Business" lorsqu'une de ses conditions d'exécution est vérifiée :

- Assigner une valeur à un champs d'un formulaire ou d'un Application Object.
- Arrêter une transaction en cours et en notifier l'utilisateur par l'affichage d'un message.
- Envoyer une notification à un utilisateur.
- Déclencher un Business Process.

L'onglet "Web Services"

Un Web Service est une fonction logicielle construite pour réaliser des interactions machine à machine par l'intermédiaire d'un réseau. Ils utilisent un modèle standardisé de données comme : XML / SOAP / WSDL. XML est utilisé pour le formatage des données, SOAP pour effectuer le transfert des données, et WSDL pour décrire le service.

Les Web Services permettent une communication entre plusieurs applications originaires de différentes sources, le tout avec un temps de réponse au plus court et une compatibilité multi-plateforme grâce au formatage XML des données.

Ainsi une application en Java peut communiquer avec une application en Perl fonctionnant sur Unix ou Windows.

Cordys permet une génération automatique d'une série de Web Services Standards pour chaque Application Object : Lecture, Ecriture, Modification, Suppression.

Il est également possible de construire des Web Services plus avancés grâce à l'assistant dédié.

L'onglet "Configure Application"

Après la création et le paramétrage de tous les éléments de l'application, il devient nécessaire de configurer l'application et ses différents onglets.

Ces onglets peuvent regrouper un ou plusieurs formulaires et chaque formulaire et/ou onglet peut être accessible ou non en fonction du privilège utilisateur.

Les privilèges utilisateurs sont à créer et à modifier dans cette section. Seul le droit administrateur est automatiquement en place, il permet un accès total à l'application.

Il est également possible de référencer une application comme source, cette dernière sera accessible à travers l'application actuelle (en particulier lors des Look-Ups)

Creation d'un "*Application Package*"

Une fois l'application prête pour être testée dans le mandant de test, il est nécessaire de générer un "Application Package" soit dans le but de créer une sauvegarde de l'application à l'instant T soit pour la déployer dans un autre mandant.

4.3 Titre de Fonction : “Automation Analyst”

4.3.1 Description de mon rôle

Mon rôle a été principalement orienté sur deux objectifs. En effet il m’a été demandé d’une part, de développer des améliorations sur un parc applicatif existant (en particulier par rapport aux attentes des “Business Owners”) ainsi que de développer et de déployer de nouvelles applications, d’autres part, il m’a été demandé d’effectuer le support applicatif sur ces mêmes applications via un système open source de tickets : GLPI.

Finalement, il m’a été demandé d’effectuer des formations *orientées utilisateur* sur plusieurs sites Français à propos de l’utilisation des applications déployées en production.

Ce genre de poste demande donc une double compétence :

- En effet, il nécessaire de connaître parfaitement son outil et la plateforme afin de développer la fonction qui manque ou d’améliorer l’interface d’une application. Une faculté d’analyse et d’adaptation rapide sont donc les qualités humaines prépondérantes dans cette activité en plus des connaissances techniques.
- Le deuxième rôle est d’effectuer un support relatif à une quinzaine d’applications. Il est donc extrêmement courant de d’interagir avec tout type d’utilisateur et ce plusieurs fois par jour. Le problème réside dans le fait que l’on ne connaît que rarement son interlocuteur, tout comme sa fonction dans l’entreprise. Il faut donc être capable d’adapter son discours en fonction des connaissances de l’utilisateur, de son “empressement” et de sa fonction dans l’entreprise. Il est donc critique d’avoir de bonnes facultés de communication (en anglais et français), une certaine capacité de discernement et de répartie.

4.3.2 Retour sur expérience

La partie technique

Concernant la partie technique, Cordys Process Factory est très peu documenté. Impossible de trouver un manuel ou même une documentation officielle. Valeo possède ses propres modules de formation mais cela reste très succinct et ce fût très complexe de me former à son utilisation, l'administration de la plateforme et particulièrement au développement sur cette plateforme. Malgré une formation intensive de deux jours à Paris, autonomie et des dizaines d'heures à tâtonner sur la plateforme furent les maîtres mots de mon expérience sur Cordys Process Factory.

Un Forum Valeo pour les développeurs sur Cordys est en place, cet outil me fut d'un grand secours, surtout durant les premiers mois de mon stage. Malgré cela je ne pense toujours pas maîtriser la plateforme. Je connais les fonctions et les éléments qui me permettent de faire ce dont j'ai besoin, mais sur une demande trop spécifique ou trop pointue il m'est nécessaire de demander de l'aide. Cordys est une plateforme propriétaire de **développement au clic**, très peu de code est à mettre en place. Ce qui signifie que si vous ne savez pas où cliquer pour effectuer telle ou telle action, vous pouvez chercher pendant des heures avant que la solution vous apparaisse. A l'inverse d'un langage de programmation qui, pour la plupart, malgré ses spécificités, reste similaire à de nombreux autres et avec de solides connaissances en algorithmique et une documentation il est relativement aisé d'arriver à ses fins.

La partie communicationnelle

C'est sans aucun doute la partie qui m'a demandé le plus d'efforts. Tant dans la forme que dans le fond. En effet, je m'étais majoritairement adressé, auparavant, à des personnes connaissant l'informatique dans le cadre de mes études. Et ce ne fut pas forcément aisé d'adapter mon discours aux connaissances de mon interlocuteur et surtout au fait que je ne sache jamais à qui je m'adresse. Un environnement comme celui de Valeo est très complexe et demande de grandes précautions lorsque l'on s'adresse à quelqu'un. Il est donc primordial d'**apprendre** à ne s'engager sur quelque chose même si on le croit acquis et 100% sûr. En effet, ce qui est possible dans un environnement quelconque n'est pas forcément possible dans l'environnement Valeo. A la fois d'un point de vue technique et également parce que l'on ne maîtrise pas tous leviers d'une action et qu'il faut obtenir l'accord d'un certain nombre de personnes avant de pouvoir déclencher certaines actions.

J'ai personnellement eu beaucoup de mal à gérer la pression durant les deux premiers mois. En effet, un environnement comme Valeo demande que tout soit fait en flux tendu et en parallèle voir limite "pour hier" pour ne paraphraser personne. C'est une situation à laquelle je n'avais jamais fait face, il a fallu **apprendre** à ne pas paniquer et se recentrer sur l'objectif, définir des priorités, gérer les timings et le tout de manière relativement automatique et intuitive car il n'est pas envisageable de passer plusieurs heures par semaine à établir un diagramme de Gantt pour la semaine. La *liste des tâches* à effectuer de Google fut sans aucun doute mon meilleur allié pendant ce stage.

Un Système ERP : SAP

5.1 Un système ERP : SAP

5.1.1 SAP, vue d'ensemble

SAP est un ERP (*Enterprise Resource Planning* ou en français : **Planification des Ressources de l'Entreprise**) développé par la société éponyme. Ce sigle signifie : **Systems, Applications, and Products for data processing**.

SAP est un puissant outil qui intègre de multiples Business Process et Fonctions en un seul système unifié. Fonctionnant sur une architecture Client / Serveur classique, chaque utilisateur (disposant des droits nécessaire) peut facilement gérer et tracer les ventes, la production, les finances de l'entreprise et les ressources humaines en temps réel.

SAP est un système modulaire qui se compose de plusieurs parties. Chacune de ces parties peut elle même se diviser en plusieurs sous modules de fonction spécifique. Tous les modules SAP sont cependant inter-connectés via une base de données centralisée qui connecte et recoupe les différents aspects et les différentes données de l'entreprise.

Pour n'en citer que quelques exemples, voici les quelques modules utilisés chez Valeo :

- **Module MM** : "Material Management" est le module logistique de SAP. Il gère les stocks, et les mouvements de stocks : entrées, sorties, et transferts.
- **Module PP** : "Production Planning" concerne la planification de la Production.
- **Module SD** : "Sales and Distribution" concerne l'administration des ventes.
- **Module FI** : "Financial" contient toutes les écritures des ventes et achats, lesquelles se déversent dans la comptabilité générale via la comptabilité client ou fournisseur.
- **Module CO** : "Controlling" concerne le contrôle de gestion.
- **Module BW** : "Business Information Warehouse" est un outil de fouille de données et par extension de Reporting.

5.1.2 Les intérêts Business de l'utilisation du système SAP

De nombreuses entreprises utilisent de nos jours des systèmes d'information qui ont été mis en place pour exécuter un certain nombre de tâches spécifiques tout en apportant un outil d'analyse et d'extraction des données transitant au sein du système.

Certains sont même capables de fournir des données temps-réels de l'état du système. En effet, ces systèmes possèdent toutes les informations nécessaires en leur sein pour permettre une utilisation rapide et multi-support afin de gérer tout type d'événements.

Typiquement, une compagnie possédant de nombreux systèmes sous son contrôle, et différents processus critiques à gérer en parallèle comme la production, la vente, la comptabilité, ... aura un système d'information d'une grande complexité. Chacun de ces systèmes fonctionnant avec une base de données bien spécifique.

Des interfaces de transfert de données entre ces systèmes sont utilisées aussi souvent que nécessaire. Le système devient très rapidement complexe et difficile à gérer. Le plantage d'un des éléments de tout le système d'information impliquera un service dégradé sur la globalité du système. Les coûts de maintenance, de sauvegarde ainsi que de maintien d'un DRP efficace deviennent prohibitifs.

SAP propose une vision différente, une vision unifiée : Un seul système d'information dans le système : SAP.

Toutes les applications peuvent accéder aux données communes du système. Chaque événement qui a lieu dans le système est enregistré dans le système et immédiatement accessible par l'ensemble des modules. L'utilisation d'un système ERP permet une amélioration de la cohérence des données et de l'efficacité du système. La simplification des actions, transactions et des processus réduit de plus la charge sur le système global.

5.2 Le projet Comp@ssV4

Préambule

J'ai travaillé uniquement sur la partie qui concerne la migration et la nettoyage des données des clients dans le système SAP. Le système SAP étant trop vaste, il m'est impossible de m'étendre sur tous les sujets. Je n'expliquerai donc dans ce rapport que les interactions à ce sujet.

Ce projet porte le nom de Comp@ssV4 : *Corporate Operational & Management Performance @vanced SyStem*. Il succèdera à Comp@ssV2 et non à Comp@ssV3 qui fut un projet avorté et non poursuivi suite à un changement de politique interne.

5.2.1 Une migration logicielle et architecturale

Cette migration porte sur deux aspects :

- Une mise à jour du logiciel.
- Un changement de l'architecture serveur (au regard des clients) pour une version plus unifiée.

Mise à jour du logiciel

Comme tout logiciel, il est maintenu pendant un certains temps. La version actuelle n'étant plus maintenue, il est nécessaire dans un environnement comme celui de Valeo de mettre à jour son système afin de réduire ses coûts de maintenance.

Changement de l'architecture serveur

En plus de la mise à niveau du logiciel SAP, Valeo a décidé de revoir entièrement son architecture applicative SAP et plus spécialement celle des clients. Une vision plus unifiée et plus centralisée a été pensée et mise en place afin de correspondre aux exigences de la nouvelle politique du Groupe Valeo.

Ancienne Structure : Comp@ssV2

L'architecture de Comp@ssV2 peut être décrite comme ceci :

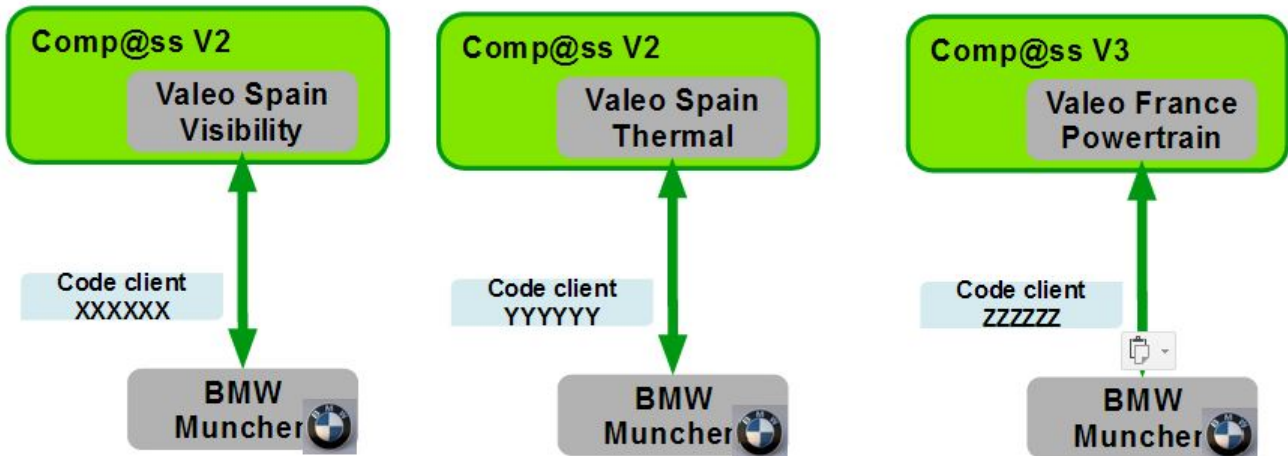


Figure 5.1 – Structure du système Comp@ssV2 - Pas de référentiel client

Nouvelle Structure : Comp@ssV4

L'architecture de Comp@ssV4 peut être décrite comme ceci :

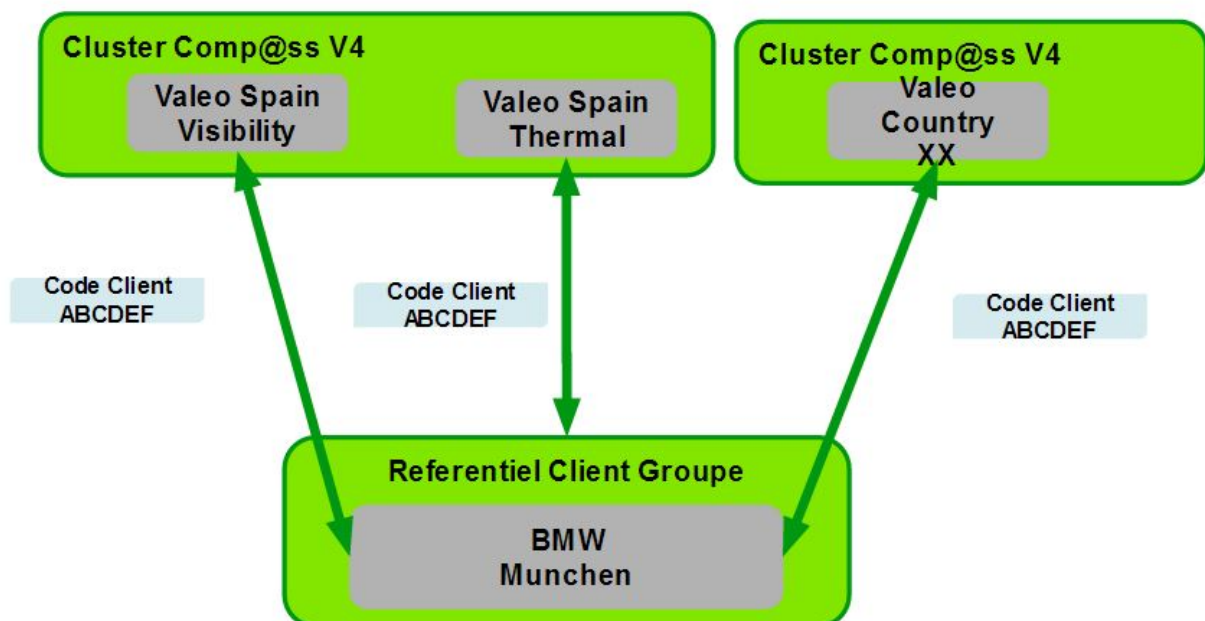


Figure 5.2 – Structure du système Comp@ssV4 - Présence d'un référentiel client Groupe

Résumé de la migration

Ainsi, c'est une migration biaxiale qui est en cours de réalisation.

Cependant, la partie logicielle n'étant pas de notre périmètre, je ne m'étendrai pas sur ce sujet.

Ainsi, pour résumer, il est établi que toutes les machines SAP de Valeo pointeront vers un annuaire SAP de clients.

Ainsi chaque site aura une codification unifiée et cohérente avec les autres. Le client numéro 1 sera le même quelque soit le système SAP de Valeo.

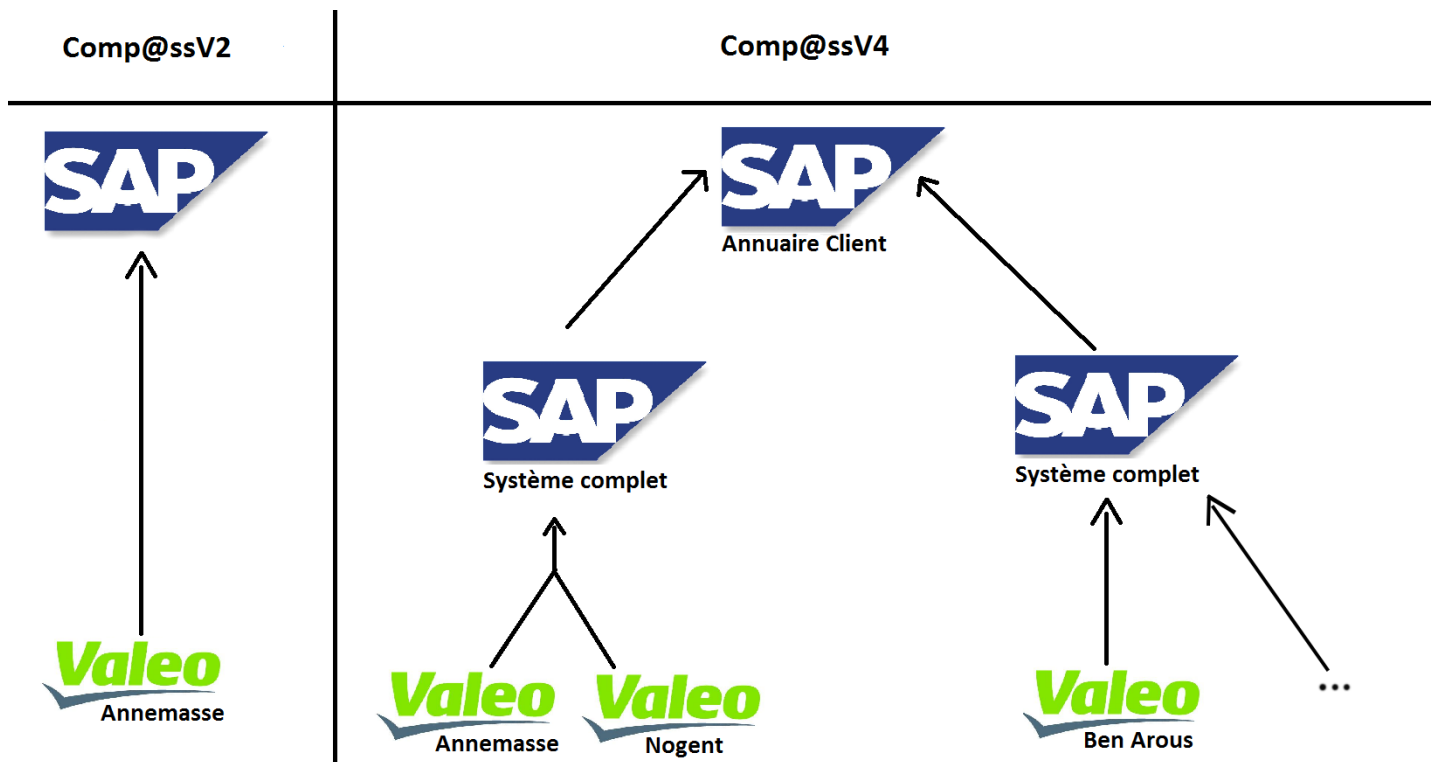


Figure 5.3 – Comp@ssV2 vs Comp@ssV4 - Nouvelle Architecture Client

Les avantages d'une telle structure sont :

- Avoir une codification commune à tout le groupe Valeo.
- Faciliter la comparaison entre les prévisions et les ventes réelles.
- Faciliter la communication entre les sites Valeo.
- Avoir une autorité commune sur un annuaire client unifié.
- Être capable de consolider le changement et les modifications des clients au niveau mondial.
- Avoir une vue globale du Business réalisé avec un client.

5.3 Un client = Une fonction partenaire

5.3.1 Les différentes fonctions partenaires

Un client peut, classiquement, être référencé sous quatre formes dans un environnement professionnel.

Ainsi nous y retrouvons :

- Le client **Donneur d'ordre**, fonction partenaire AG dans SAP (de l'Allemand : *Auftraggeber*)
- Le client **Livré**, fonction partenaire WE dans SAP (de l'Allemand : *Warenempfänger*)
- Le client **Facturé**, fonction partenaire RE dans SAP (de l'Allemand : *Rechnungsempfänger*)
- Le client **Payeur**, fonction partenaire RG dans SAP (de l'Allemand : *Regulierer*)

Un exemple simple sur les fonctions partenaires

Prenons une situation simple et courante pour expliquer que sont ces fonctions partenaires.

Un enfant veut acheter un cadeau à sa maman pour la fête des mères, pour cela il se connecte sur son site en ligne préféré et choisi ce qu'il veut commander.

Il passe la commande, seulement étant enfant il n'a pas de carte bancaire et ne peut donc payer ...

Papa aide donc son fils à passer commande avec sa carte bancaire.

Une semaine plus tard, **Maman** reçoit le colis.

Dans cette histoire, l'**enfant** fait office de donneur d'ordre : C'est lui qui effectue la commande. **Papa** quand à lui grâce à sa carte bancaire est devenu le client payeur et le client facturé. Quand à **Maman**, elle est donc le client livré.

Pour résumer :

- Enfant = Client Donneur D'ordre = **AG**
- Papa = Client payeur et Client Facturé = **RE & RG**
- Maman = Client Livré = **WE**

Cette explication est primordiale pour la suite car dans un contexte industriel il est très courant d'avoir quatre clients différents pour chacune des fonctions partenaires pour une seule et même commande.

Il est donc **nécessaire** d'établir une codification uniforme pour chacune de ces fonctions partenaires.

5.4 Reprise des clients dans Comp@ssV4

5.4.1 Deux groupes de clients : "Cluster" et "Référentiel Groupe"

Dans le contexte du projet Comp@ssV4, tous les clients ne seront pas repris au niveau du référentiel commun groupe.

Les autres seront définis uniquement au niveau de l'instance "Cluster", c'est à dire le système SAP sur lequel fonctionne le site Valeo car ces clients lui sont propres et ne sont pas partagés avec les autres sites.

#	Type de Clients	Groupe de Clients	Référentiel Groupe de Clients	Clientréféréncé Cluster
1	Client donneur d'ordre	Z001	X	X
2	Client livré	Z002	X	X
3	Client payeur	Z003	X	X
4	Client facturé	Z004	X	X
5	Employé Valeo, Service Vente	Z005		X
6	Sous Traitant	Z006		X
7	Client au coup par coup	Z007		X
8	Agent TVA	Z008		X
9	Client Intra Valeo	Z009	X	X

Figure 5.4 – Reprise des clients Comp@ssV2 - Referentiel Groupe vs Cluster

5.4.2 Cadre de reprise d'un client

Il n'est pas cohérent de reprendre tous les clients des systèmes SAP de tout Valeo. En effet, certains ne sont plus actifs, n'existent plus, ont déménagés, ou n'ont tout simplement plus aucune activité avec Valeo.

Ainsi chaque site Valeo, devra choisir parmi ses clients comp@ssV2, ceux qui répondent à l'ensemble des critères suivant :

- Tous les clients non marqués "à supprimer" (On ne supprime jamais rien, on applique des marqueurs de suppression dans un système ERP : soucis de traçabilité).
- Tous les clients ayant effectué une commande durant les 2 dernières années.
- Tous les clients avec une commande non clôturée (en cours ou en litige ou en attente de paiement)

Tous ces filtres sont ajoutés à l'outil d'extraction que nous utilisons : **Informatica**.

5.5 Une nouvelle codification client

5.5.1 Présentation de la nouvelle codification

#	Client type	Account group *	Codification
1	Donneur d'ordre	Z001	<u>XXXXCCSNNN</u> XXXX: Code Client CC: Pays de l'adresse du client 'S': Sold-To-Party (Donneur d'ordre en anglais) NNN: Compteur Numérique
2	Livré	Z002	<u>XXXXYYYYAA</u> XXXX: Code Client YYYY: Code Usine AA: Compteur alphabétique
3	Payeur	Z003	<u>XXXXCCPNNN</u> XXXX: Code Client CC: Pays de l'adresse du client 'P': Payeur NNN: Compteur Numérique
4	Facturé	Z004	<u>XXXXCCBNNN</u> XXXX: Code Client CC: Pays de l'identifiant TVA 'B': Bill-To Party (Facturé en anglais) NNN: Compteur Numérique
5	Client Intra Valeo	Z009	<u>VVVVBBBB</u> VVVVV: Code Orgaloc (nomenclature Valeo) BBB: Compteur hexadécimal

Figure 5.5 – Nouvelle Codification Client du système Comp@ssV4 - Codification Unifiée

La codification Comp@ssV4 assure que chaque client ne correspondra qu'à **une seule fonction partenaire** et non plusieurs comme cela était parfois le cas dans Comp@ssV2. Ainsi un client du groupe Z001 n'aura qu'une fonction partenaire : AG au lieu de AG/WE/RE et RG, comme parfois, dans Comp@ssV2.

La codification Comp@ssV4 définit le **code Client étant unique**. Ainsi deux clients différents auront un code client différent (exemple : *Renault et BMW*), par contre deux clients du même Groupe (au sens entreprise) auront un code client commun (exemple : Renault Douai et Renault Compiègne).

La codification Comp@ssV4 définit le code usine de manière unique quelque soit l'entreprise et l'usine. Ainsi deux usines ne peuvent avoir en aucun cas le même numéro (Exemple : Renault Douai, Renault Compiègne et BMW Munich auront tous les trois un code Usine obligatoirement différent).

5.6 Processus de migration de Comp@ssV2 à Comp@ssV4

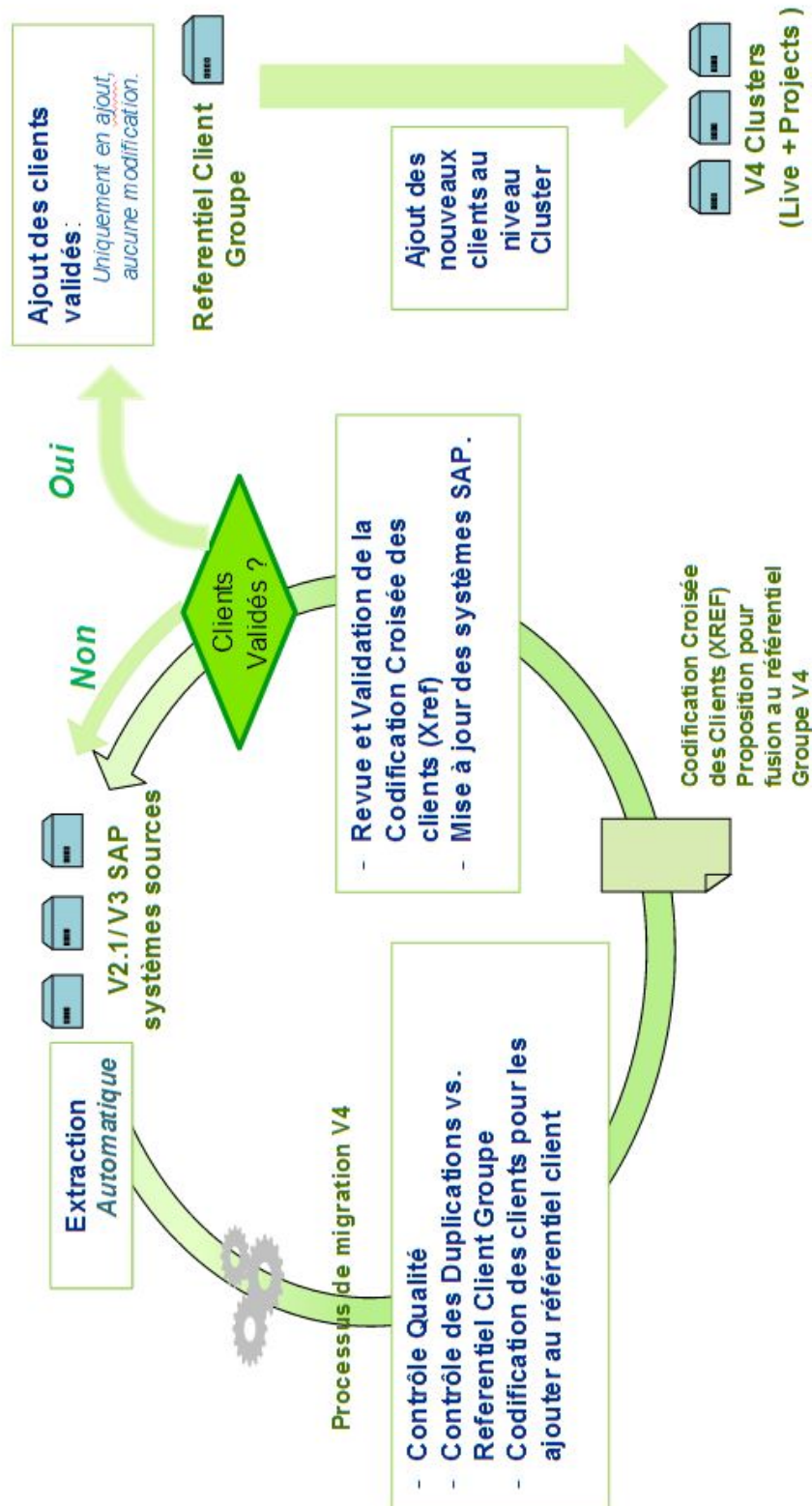


Figure 5.6 – Processus de migration de Comp@ssV2 à Comp@ssV4

5.7 Explication du travail effectué

5.7.1 Définition de l'objectif

Afin de mener à bien la migration de Comp@ssV2 à Comp@ssV4, il nous a donc été demandé de réaliser le nettoyage des données Clients et d'en effectuer la reprise pour Comp@ssV4.

La reprise consiste à faire correspondre autant que possible les clients déjà codifiés par les autres sites dans le référentiel Client Groupe de Comp@ssV4. Pour les autres, nous serons tenus de les codifier en respectant les règles de codification afin de les intégrer dans le système.

La problématique qui s'est donc posée à nous est la suivante :

“Comment réaliser une comparaison efficace entre les clients Comp@ssV2 de notre système et ceux du référentiel groupe déjà codifiés dans le nouveau système ?”

Effectivement, il est difficilement pensable de recouper et de comparer plus de 2500 clients de notre système avec les 6000 clients déjà existant dans le référentiel à la main.

Il nous a donc fallu trouver un outil pour effectuer cette comparaison.

5.7.2 L'algorithme de Ratcliff et Obershelp

Grâce à un cours sur la détection de la redondance des données que j'avais suivi en Erasmus à Vienne, j'ai pu proposer l'algorithme mis au point par **Ratcliff et Obershelp** sur la détection de séquence dans les chaînes de caractères.

Cet algorithme renvoie **un pourcentage de ressemblance** entre deux chaînes de caractères qu'il reçoit en entrée.

Cet algorithme permet de répondre à notre besoin pour les raisons suivantes :

- Possibilité "d'absorber" de faibles variations : Chaque site Valeo n'a pas forcément identifié l'adresse ou le nom complet du client de la même manière. Il est donc nécessaire d'établir une **marge d'erreur** lors de la reconnaissance automatique.
- Algorithme de relativement faible complexité : N^2 .

Principe de l'algorithme

L'algorithme de Ratcliff et Obershelp calcule la similarité entre deux chaînes de caractères de la manière suivante :

$2 * (\text{nombre de caractères identiques}) / (\text{nombre de caractères dans les deux chaînes})$

Correspondance des caractères dans un premier temps de manière relative à leur position (*le caractère 1 de la chaîne 1 correspond-il au caractère 1 de la chaîne 2 ?*), le tout sur la plus longue sous-séquence en commun (les chaînes n'ont pas forcément la même taille). A cela s'ajoute de manière récursive la correspondance des caractères non associés toujours en cherchant la plus grande zone correspondante et le tout jusqu'à ce que plus aucune association ne soit possible.

Exemple :

La similarité entre ALEXANDRE et ALEKSANDER est égale à :

$$2 * (3+3+1+1) / (9+10) = 0.84 \text{ (Corresponds : ALE (3), AND (3), E(1), R(1))}$$

5.7.3 Objectifs de notre méthode

L'objectif est d'être capable de **détecter** une forte similarité entre deux clients. Mais il ne nous semblait pas judicieux de réaliser l'association de manière **automatique** pour des raisons de contrôle des données (l'impact d'une erreur est important car cela touche les clients de Valeo).

Deux objectifs apparaissent donc :

Limiter les "faux positifs"

Afin d'effectuer notre travail de manière efficace il est important d'effectuer un **tri**, entre les clients qui ne correspondent **pas**, ceux qui correspondent **un peu**, et ceux qui correspondent **beaucoup**.

Par souci de temps, nous avons choisi de nous focaliser sur ceux qui ont un fort taux de ressemblance (avec la méthode précédemment expliquée).

Il paraît donc essentiel de limiter ce que nous appellerons un **faux positif** : Un faux positif c'est deux clients qui semblent identiques mais qui en réalité ne le sont pas (Exemple : Renault Toulouse et Renault Toulon).

Nous devons donc définir un **seuil** de détection de manière à ne pas générer trop de pollution dans nos résultats.

Aucun "faux négatif" ou presque

Encore plus gênant qu'un faux positif, nous avons le faux négatif. Cette fois ci nous avons deux clients identiques et malheureusement identifiés de manière trop différentes et le taux de ressemblance est en dessous de notre seuil de détection. Conséquence : Nous perdons une association et ce client sera donc recodifié, ce qui va à l'encontre du principe même du projet (limiter le nombre de client et une codification unique pour un même client au sein de tout Valeo).

5.7.4 La problématique du seuil de détection

Nous avons effectué de nombreux tests afin de déterminer quel était le **meilleur** seuil de détection afin de ne pas générer trop de pollution (faux positifs) et surtout de ne rater aucune correspondance ou presque (faux négatifs).

Suite aux tests effectués nous en avons déterminé le seuil de : 60%

5.7.5 Principe de l'outil que nous avons développé

Afin de répondre au besoin, nous avons donc établi un outil permettant de recouper les clients V2 avec les clients V4.

Cet outil cherche à réaliser des paquets de clients V2 et V4 ayant une forte ressemblance. Ces paquets sont constitués dans un fichier excel tierce qui pourra être utilisé comme fichier de travail. Le fichier des clients extrait reste inchangé pour des raisons de corruption de données en cas d'erreur de manipulation.

Cet outil a été développé par mes soins en VBA et implémenté sous forme de macro **instable** dans Excel par le biais du fichier spécialisé excel : *.xla.

Voici un bref descriptif de son fonctionnement :

1. Génération de la clé de comparaison par concaténation des champs **importants** dans notre analyse : *nom, ville, adresse*
2. Réalisation d'une boucle $x \in V4$ qui analyse la ressemblance entre $y \in V2$ et x .
3. Si ressemblance $(x,y) > 60\%$ → Constitution d'un paquet dans le fichier excel de sortie.
4. A chaque fois que l'outil trouve une ressemblance entre V2 et V4, il procède également à l'analyse de V4 par rapport à V4. Cela permet de balayer dans les deux sens la liste des clients et de constituer des paquet plus cohérent et non des associations 1 :1.

5.7.6 Les principales difficultés que j'ai rencontrées

Je ne connaissais pas le VBA et ce ne fut pas forcément aisé d'obtenir le résultat escompté immédiatement voir même de *débugguer* ...

De plus la structure de l'algorithme n'étant pas simple, j'ai dû passer beaucoup de temps en phase de conception afin d'obtenir le schéma le plus efficace et fonctionnel possible.

Le langage VBA, celui des macros excel, n'est pas forcément très rapide. Les accès mémoire et encore plus les écritures ne se font pas rapidement. Ainsi après réalisation de la macro j'ai procédé à une longue phase de recherche et d'optimisation sur la rapidité des différentes fonctions pour effectuer certaines tâches. J'ai divisé le temps d'exécution par un facteur 3. Et malgré tout l'exécution a tout de même duré près de deux heures et demi.

5.7.7 Quel futur pour cet outil

A ma grande surprise, l'outil mis en place intéresserait les responsables du projet au niveau Groupe. Il m'a été demandé de leur faire une présentation de l'outil à la fois dans le principe et dans l'utilisation.

Il m'a été demandé en ce sens de le modifier afin qu'il soit **industrialisable** et **utilisable par un non informaticien** : comprendre sans modification du code.

Cette nouvelle a récompensé les efforts que l'équipe a pu mettre dans le projet et permet de mettre un point final, pour ma part, sur ce projet au regard de l'échéance de mon stage.

Conclusion

C'est avec plaisir que maintenant je peux regarder en arrière, rire de mes erreurs, sourire de mes nombreuses questions mais surtout être satisfait du chemin parcouru.

Les nombreux problèmes qui se sont posés à moi m'ont forcé à toujours aller plus loin dans mes raisonnements et mes recherches, à trouver *la parade* ou *l'astuce* qui fera que *cette fois-ci, ça marchera*. Un moment où j'ai pu tirer profit de l'enseignement reçu à l'UTC et en Erasmus à Vienne et mettre en difficulté mes connaissances afin de consolider mes acquis.

En particulier dans le cadre de mon travail sur SAP, j'ai eu la chance de créer un outil informatique qui sera par la suite réutilisé et peut être améliorer par les équipes Valeo ultérieurement.

J'en profite, encore une fois, pour remercier tout ceux qui ont participé au bon déroulement de ce stage qui m'aura permis d'avoir un vaste aperçu du monde de l'entreprise, de me familiariser avec un environnement complexe comme celui de Valeo, de découvrir des sujets passionnants. Ce stage fut l'occasion de confronter des connaissances théoriques à des connaissances pratiques et au monde de l'entreprise.

Je ne saurais faire la liste de tout ce que j'ai pu apprendre et j'aurais souhaité que ce stage dure un peu plus longtemps pour avoir le temps de me pencher sur le logiciel BW, module SAP de Fouille de données.

Ce stage m'a donc beaucoup appris dans des domaines techniques et concernant la vie en entreprise, et cette expérience me sera très utile pour m'adapter dans différents types d'organisations. Il est un premier élan dans le monde du travail en tant qu'ingénieur. Une première vraie prise de responsabilité professionnelle que m'a offerte Valeo.

Bibliographie

- [1] <http://www.valeo.com/le-groupe/profil-du-groupe-valeo.html>.
- [2] <http://www.valeo.com/nos-activites/>.
- [3] <http://www.valeo.com/nos-activites/systemes-de-confort-et-d-aide-a-la-conduite/>.
- [4] <http://commons.wikimedia.org/wiki/File:Nuage33.png>.
- [5] [http://commons.wikimedia.org/wiki/File:Cloud_Computing_-_les_différents modèles_de_service.svg](http://commons.wikimedia.org/wiki/File:Cloud_Computing_-_les_diff%C3%A9rents_mod%C3%A8les_de_service.svg).

Rédaction du Rapport en langage LaTeX

Ce rapport a été entièrement rédigé en **LaTeX**.

Les sources sont disponibles et téléchargeables sur Github à l'adresse suivante :

<https://github.com/JonathanDekhtiar/Rapport-TN09>.