



**HAL**  
open science

# Unification et création d'indicateurs de pilotage pour les activités de logistique

William Drugeon

► **To cite this version:**

William Drugeon. Unification et création d'indicateurs de pilotage pour les activités de logistique. Ingénierie, finance et science [cs.CE]. 2014. dumas-01377527

**HAL Id: dumas-01377527**

**<https://dumas.ccsd.cnrs.fr/dumas-01377527>**

Submitted on 7 Oct 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

CENTRE REGIONAL DE TOULOUSE

MEMOIRE

présenté en vue d'obtenir

le DIPLOME D'INGENIEUR CNAM

SPECIALITE : Informatique

OPTION : Architecture et ingénierie des systèmes et des logiciels (AISL)

par

**William DRUGEON**

---

**Unification et création d'indicateurs de pilotage pour les activités de  
logistique**

**Soutenu le 05 Décembre 2014**

---

**JURY**

<b>PRESIDENT :</b>	<b>M. Yann POLLET</b>	<b>Professeur, CNAM Paris</b>
<b>MEMBRES :</b>	<b>M. Hadj BATATIA</b>	<b>Maitre de conférences, INPT Toulouse</b>
	<b>M. Xavier CREGUT</b>	<b>Maitre de conférences, INPT Toulouse</b>
	<b>M. Xavier XIOL</b>	<b>Directeur site Easydis, Toulouse</b>
	<b>M. Christian SORBA</b>	<b>Directeur site Aldi, Toulouse</b>



## Remerciements

Il n'est pas facile pour un jeune tout juste sorti du système scolaire de trouver un emploi en alternance, c'est pourquoi je remercie M. XIOL, directeur du site d'Eurocentre, qui a rendu possible mon intégration au sein du groupe Casino, et m'a accueilli durant ces trois années.

Je tiens à remercier tout particulièrement M. SORBA, responsable d'exploitation, pour son soutien sur le projet, sa pédagogie sans égal, sa critique constructive et sa présence remarquable, et M. BATATIA, professeur à l'IPST Cnam et Directeur de mon mémoire qui m'a guidé dans mon travail et m'a aidé à trouver des solutions pour avancer. Merci également à l'ensemble du Pôle Etudiants avec qui j'ai pu réaliser une alternance enrichissante et sans qui ce projet n'existerait pas.

Suivre cette formation d'ingénieur informatique avec le CNAM Midi-Pyrénées a été un excellent choix, j'ai énormément appris et j'ai surtout été conforté dans mon projet professionnel qui se concrétise aujourd'hui.

## Glossaire

Terme/Acronyme	Définition
AJAX	Asynchrones JavaScript and XML, est une architecture combinant plusieurs langages
APC	Alternative PHP Cache, est un module PHP
APS	Advanced Planning System
CIT	Casino Informations Technology
CP	Congés Payés
CRUD	Create Read Update Delete, désigne les quatre opérations de base pour la persistance des données
CSV	Comma-separated values
DAO	Data Access Object
DQL	Date Query Language
DRH	Directeur Ressources Humaines
ERP	Enterprise Resource Planning
GEPC	Gestion prévisionnel des emplois et des compétences
HTML	HyperText Mark-Up Language
IFS	International Food Standard
IHM	Interface Homme Machine
JEE	Java Enterprise Edition, est une spécification pour la technique Java
JSP	Java Server Pages
KPI	Key Performance Indicator
MVC	Modèle Vue Contrôleur
MYSQL	Système de gestion de bases de données relationnelles (SGBDR)
ORM	Object-relational mapping, technique de programmation informatique
PBS	Product Breakdown Structure
PGI	Progiciel de gestion intégré
PHP	Hypertext Preprocessor, langage de programmation
RAM	Désigne la mémoire vive d'un ordinateur
RFID	Radio Frequency Identification
RH	Ressources Humaines
RTT	Réduction du temps de travail
SCEM	Supply Chain Event Management, concept de gestion des aléas dans la chaîne logistique
SI	Système d'information
ScrumMaster	Personne en charge de la supervision sur un projet utilisant une méthode type Scrum
SOA	Services Oriented Architecture, est une forme d'architecture de médiation
SQL	Structured Query Language
SRM	Supplier Relationship Management, gestion de la relation fournisseur
TMS	Transport Management System, progiciels de gestion et d'optimisation du transport
UO	Unité d'œuvre, unité de valeur opérationnelle
VBA	Visual Basic Application, langage de programmation
VPN	Virtual Private Network, système permettant de créer un lien direct entre des ordinateurs distants
WMS	Warehouse Management System, catégorie de progiciels destinés à gérer les opérations d'un entrepôt

## Table des matières

Remerciements.....	3
Glossaire.....	4
Table des matières.....	5
Introduction.....	9
<b>I ETAT DES LIEUX.....</b>	<b>11</b>
I.1 LES SYSTEMES D'INFORMATION DES ENTREPRISES.....	11
I.1.1 Rôle et importance.....	11
I.1.2 Composition.....	12
I.2 LE CAS PARTICULIER DU GROUPE CASINO.....	15
I.2.1 Présentation générale du groupe.....	15
I.2.2 Présentation de leur système d'information.....	16
I.2.3 Les problématiques rencontrées.....	21
I.3 PROPOSITIONS DE SOLUTIONS AUX PROBLEMATIQUES.....	22
<b>II INITIALISATION DE LA REALISATION D'UN SYSTEME INTEGRE.....</b>	<b>23</b>
II.1 ANALYSE DES BESOINS.....	23
II.1.1 Les approches d'analyse.....	23
II.1.2 Méthodes de définition.....	23
II.1.3 Besoins identifiés.....	24
II.1.4 Méthode de suivis des exigences.....	27
II.2 DEFINITION DU PROJET.....	29
II.2.1 Buts et objectifs du projet.....	29
II.2.2 Définition de la portée.....	30
II.2.3 Produits livrables.....	31
II.2.4 Etude des risques.....	32
II.3 STRUCTURE DE L'EQUIPE DE PROJET.....	33
<b>III ETUDES ET CONCEPTION DU SYSTEME LOGICIEL.....</b>	<b>35</b>
III.1 ETUDE DE FAISABILITE.....	35
III.1.1 Etude des solutions technologiques.....	35
III.1.2 Comparatif et choix de la solution.....	39

III.1.3	Architecture fonctionnelle.....	42
III.1.4	Architecture logicielle .....	54
III.1.4.1	Architecture en couches.....	54
III.1.4.2	Architecture Framework Symfony 2 : .....	55
III.1.4.3	Couche métier .....	57
III.1.5	Modèle du domaine .....	58
III.1.6	Intégration des données .....	69
III.1.7	Gestion des accès et sécurité.....	72
III.1.8	Architecture technique.....	75
III.2	GESTION DU PROJET .....	75
III.2.1	SCRUM .....	75
III.2.2	Macro planning .....	77
<b>IV</b>	<b>DEVELOPPEMENT DE L'APS .....</b>	<b>79</b>
IV.1	COMMUNICATION SUR LE PROJET .....	79
IV.2	PHASE DE TEST .....	80
IV.3	MISES EN PRODUCTION .....	82
IV.4	DEVELOPPEMENT ET PROCESSUS AGILE .....	83
IV.5	AVANCEMENT DU PROJET .....	84
<b>V</b>	<b>SUPPORT, MAINTENANCE, ET EVOLUTIONS .....</b>	<b>85</b>
V.1	SUPPORT ET FORMATION.....	85
V.1.1	Manuels d'utilisation.....	85
V.1.2	Formation des utilisateurs.....	85
V.2	EVOLUTIONS .....	86
V.2.1	Fonctionnalités envisagées .....	86
V.2.2	Projet de test multi site.....	87
V.2.3	Projet de déploiement national.....	87
V.2.4	Perspectives .....	88
V.2.4.1	Etude des prévisions de vente .....	88
V.2.4.2	Démarche Big Data .....	89
	Conclusion .....	93
	Bibliographie.....	95
	Liste des figures.....	97





## Introduction

Je commencerai par citer Henri Bergson qui disait : "L'homme devrait mettre autant d'ardeur à simplifier sa vie qu'il en met à la compliquer".

C'est avec cette idée directrice que la majorité des projets informatiques voient le jour, l'objectif premier étant toujours de "simplifier" une tâche, de rendre celle-ci plus efficace, performante, reproductible. Et c'est aussi l'idée mère de ce projet.

Toutes les entreprises disposent aujourd'hui de systèmes d'information plus ou moins complexes. Ces systèmes sont le cœur du fonctionnement de la plupart d'entre elles. Par ailleurs, de nouvelles tendances, notamment à devenir « DataCentric », ne font qu'accentuer cet état de cause.

La complexité et l'importance de ces systèmes augmentant, un certain nombre de problématiques ont vu et voient encore le jour.

Ce mémoire se place comme un modèle à la résolution de certaines d'entre elles, plus particulièrement celles liées à l'intégration de briques métier et à l'aide à la prise de décision.

Les problématiques traitées sont :

- comment et pourquoi unifier les outils d'un système d'information ?
- comment créer des indicateurs pour la prise de décisions ?

Avant toute chose, nous ferons un état des lieux sur les systèmes d'informations en général, puis plus spécifiquement sur celui du groupe Casino. Nous dégagerons de ce constat les problématiques principales ainsi que des propositions de solutions.

La suite du mémoire nous permettra de voir la réalisation ainsi que la mise en œuvre de la solution choisie.

Ainsi nous débuterons par l'analyse détaillée des besoins et l'étude de la phase d'initialisation du projet. Cette partie mettra en avant la transcription fonctionnelle des problématiques posées.

Dans un deuxième temps, nous consacrerons du temps à la description du travail de conception et d'architecture. Cette partie proposera les différents choix d'architecture possibles et ceux réalisés pour le cas du projet.

Ensuite nous traiterons des éléments d'organisation propres au développement d'un projet, illustré par l'Advanced Planning System (APS). Nous détaillerons notamment les principales phases d'un processus de développement dans le cadre d'une approche agile.

Enfin, la dernière partie composant ce mémoire vise à traiter de l'ensemble de la post production du projet. Nous verrons ainsi les étapes de suivi, de formation, de déploiement global, et terminerons l'ensemble par les perspectives offertes par ce projet, mais également à un grand nombre de systèmes d'information d'entreprises du domaine de la logistique.

# I Etat des lieux

## I.1 Les systèmes d'information des entreprises

Dans cette partie, nous allons voir l'importance des systèmes d'information dans les entreprises d'aujourd'hui.

### I.1.1 Rôle et importance

Qu'est-ce qu'un système d'information ?

D'après l'ouvrage « Les systèmes d'information en réadaptation » de Courcy R (1992) :  
« *Un système d'information (SI) est un ensemble organisé de ressources (matériels, logiciels, personnel, données et procédures) qui permet de collecter, regrouper, classifier, traiter et diffuser de l'information dans un environnement donné.* »

On pourrait le comparer à notre système nerveux. En effet, tous deux jouent un rôle similaire : celui de transmettre les informations, le système nerveux au cerveau, le système d'information aux décisionnaires. Cette comparaison a d'autant plus de sens que si leurs rôles sont semblables, il en va de même de leur importance : tous deux sont indissociables de leur environnement.

En définitive, les entreprises sont complètement dépendantes et indissociables de leur système d'information : certaines deviennent même aujourd'hui « Data Centric » autrement dit, basées uniquement sur des données.

Ces données contiennent forcément le savoir-faire de l'entreprise, mais elles représentent surtout un élément stratégique capital. De plus, l'arrivée de nouvelles technologies, telles que le Big Data, ne fait qu'accroître cette réalité.

Si nous en sommes là aujourd'hui, c'est bien parce que les systèmes d'information, et plus particulièrement l'ensemble des outils qui les composent, se proposent de résoudre

de très nombreuses problématiques rencontrées, ou bien même spécifiques à un métier donné.

De nombreux outils existent aujourd'hui dans le monde des progiciels de gestion et d'optimisation dont font partie les Enterprise Resource Planning (ERP) ou progiciel de gestion intégré (PGI). Certains de ces outils sont en réalité des systèmes de complexité variables. Toutefois ils constituent généralement le gros des systèmes d'information auxquels ils appartiennent.

### I.1.2 Composition

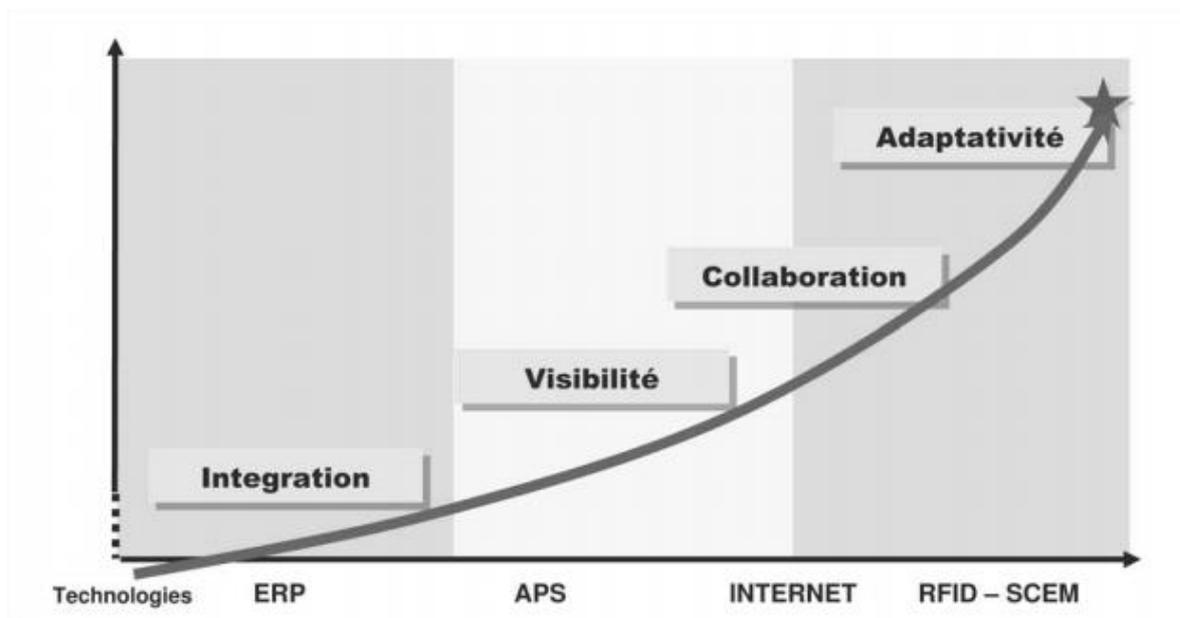


Figure 1: Le système d'information : de l'intégration à l'adaptativité.

Le schéma ci-dessus représente un rapport entre quatre technologies et leur action au sein d'un système d'information. Les trois premiers concernent toutes les entreprises (ERP, APS, Internet). Le dernier se place plus particulièrement dans le cadre d'une supply chain (concerne principalement des entreprises de logistique).

Les deux premiers éléments présentés ne sont pas à proprement dit des technologies, mais plus exactement des systèmes logiciels complexes.

- les ERP ont pour première fonction l'intégration.

Cette fonction est essentielle pour la mise en place par l'entreprise d'un modèle orienté en processus. En fait, l'intégration intra-entreprise est le principe de base qui assure une visibilité totale au sein de l'organisation. Cela passe par la mise en place d'un outil dont le rôle principal est le support aux processus métier mais qui, dans un même temps, permet de standardiser ces processus, et donc d'assurer cette visibilité globale.

- les APS pour Advanced Planning System.

L'APS est un outil complémentaire aux ERP. Il propose des services basés sur la planification des ressources et offre une visibilité globale sur celles-ci.

Il n'est pas toujours facile de déterminer la frontière entre ERP et APS. Bien souvent, les ERP, tels que SAP, intègrent un module APS pour la planification. De plus, les objectifs visés par ces systèmes sont similaires. De nouveau, on retrouve l'importance de la visibilité globale.

La corrélation entre visibilité et pilotage est simple à réaliser, de même que celle entre pilotage et stratégie. Finalement la visibilité offerte par ces systèmes sert à tous les niveaux de décision de l'entreprise.

Toutefois la « visibilité » se concrétise par des éléments tels que les indicateurs de performance, en anglais key performance indicator (KPI). La construction et le choix de ces KPI constituent alors un élément essentiel pour la chaîne de décision d'une entreprise.

- Internet

S'il est une technologie qui a fondamentalement changé nos communications, c'est bien internet. Ce dernier joue un rôle fondamental : il permet non seulement de relier l'entreprise à son environnement extérieur, mais également de créer un réseau intranet en fusionnant plusieurs réseaux distants entre eux.

- RFID

La RFID n'est qu'un exemple de technologie intégré aux systèmes d'information, qui diffère un peu des solutions traditionnelles cités précédemment. Moins généralement utilisée, elle est notamment très appréciée pour tout processus logistique. Ainsi la RFID accélère considérablement la circulation des informations et s'impose là où les autres méthodes manuelles de collecte de données atteignent leurs limites.

(Reix R. (2002), «Système d'information et management des organisations», Vuibert, 4ème édition, Paris)

(Cigref, (2002), Gouvernance du système d'information – Problématiques et démarches, Disponible sur [<http://www.cigref.fr/gouvernance-du-systeme-dinformation-problematiques-et-demarches>])

## I.2 Le cas particulier du groupe Casino

### I.2.1 Présentation générale du groupe

Créé en 1898, Casino est un acteur historique du commerce alimentaire en France. Avec près de 12 000 magasins, le Groupe est aujourd’hui l’un des leaders mondiaux de la distribution et réalise plus de 50% de son chiffre d’affaires à l’étranger, notamment en Amérique Latine et en Asie du Sud-Est.

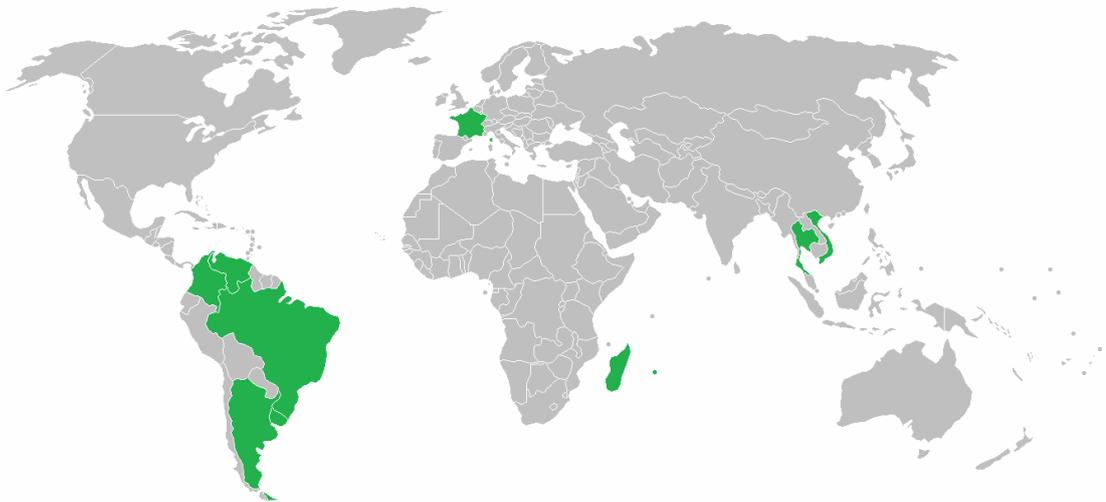


Figure 2 : Cartographie du Groupe Casino

Casino a décidé en 2000 de créer sa filiale dédiée à la logistique : Easydis.

Cette filialisation marque une évolution importante dans l’histoire du groupe Casino. Elle a permis à Easydis de renforcer son efficacité et sa rigueur dans la maîtrise de ses coûts et de se rendre plus compétitif vis-à-vis des concurrents.

Les activités principales de la société sont :

- l’entreposage, avec en charge la gestion physique des stocks au sein de ses entrepôts
- la préparation de commandes

- l'organisation des transports pour l'approvisionnement des magasins dans un souci permanent de maîtrise des coûts et de qualité de service.

Easydis est en lien direct avec ses clients avals, ce qui lui permet de recevoir en temps réel les commandes passées et à les préparer dans la journée.

Afin de gérer efficacement son activité et d'être opérationnelle en toutes circonstances, Easydis s'organise autour de trois axes :

- La typologie des produits (produits secs, frais et surgelés ; le bazar ; le textile)
- Le format de magasins GMS (Grande et Moyenne Surface) et réseau Proxi (Proximité)
- Le schéma de distribution (régional ou national)

Easydis compte 22 sites logistiques répartis sur l'ensemble du territoire et positionnés au plus près de ses clients. Parmi ces 22 sites, le site d'Eurocentre est celui sur lequel s'est porté l'ensemble des études pour la réalisation du système illustrant ce mémoire. A titre anecdotique, le service « méthodes et organisation », centralisé au siège de la société Casino, utilise souvent ce centre pour tester les nouveaux outils et processus créés.

## **1.2.2 Présentation de leur système d'information**

Les systèmes d'information des Supply Chain font parti de ceux qui ont le plus fortement évolué avec l'arrivée de systèmes comme : ERP (Enterprise Resource Planning), TMS (Transportation Management System), WMS (Warehouse Management System), APS (Advanced Planning System), SRM (Supplier Relationship Management) ... sans oublier l'arrivée de technologie comme le Big Data ou même le RFID.

Concrètement, pour le groupe Casino, l'ensemble de l'activité purement logistique repose sur deux WMS, respectivement GOLD, développé pour Casino, et REFLEX plus ancien.

L'exploitation « Epicerie » intègre pleinement GOLD. Il lui permet de gérer l'ensemble des stocks ainsi que les activités de préparation assistées vocalement voir visuellement.

L'exploitation « Frais », elle, se base encore sur REFLEX qui, bien que plus ancien dans sa conception, dispose de fonctions indispensables, comme le mode de préparation « allotissement ».

REFLEX est une solution développée sur AS400 alors que GOLD, plus récent, est développé en Java. Il faut noter que seul GOLD possède des outils dédiés au décisionnel. Il intègre un ensemble de fonctions d'exportation et de tableaux de bords des commandes des magasins, mais aussi plus généralement de la productivité.

Certaines interopérabilités existent entre les deux systèmes. L'ensemble de la chaîne d'approvisionnement est gérée par GOLD Central qui permet la génération de tous les ordres de préparation traités en entrepôts.

Casino cherche depuis plusieurs années à supprimer complètement REFLEX pour le remplacer par GOLD, mais n'y est pour le moment pas parvenu. Cependant, il est fort probable que cette transition se fasse dans les années à venir.

En plus des WMS, sont également présents des systèmes tels que SAP, pour toute la gestion financière, mais également Etemptation sur lequel nous allons nous arrêter.

Etemptation permet de gérer de nombreuses données associées aux Ressources Humaines (RH) : les heures réalisées par le personnel, notamment grâce au système de pointage par badge, mais également la gestion du planning, congés payés, repos, absences ou autre rythme horaire du personnel. En plus de cela, il intègre la gestion des contrats.

L'ensemble du personnel du groupe dédié à la modification et à la maintenance de ces systèmes est centralisé dans un département nommé CIT. Ce dernier se situe près du siège du groupe Casino. L'intégration de nouveaux outils parmi le système d'information de la société est donc soumise aux processus de validation de ce département.

### Diagramme des flux

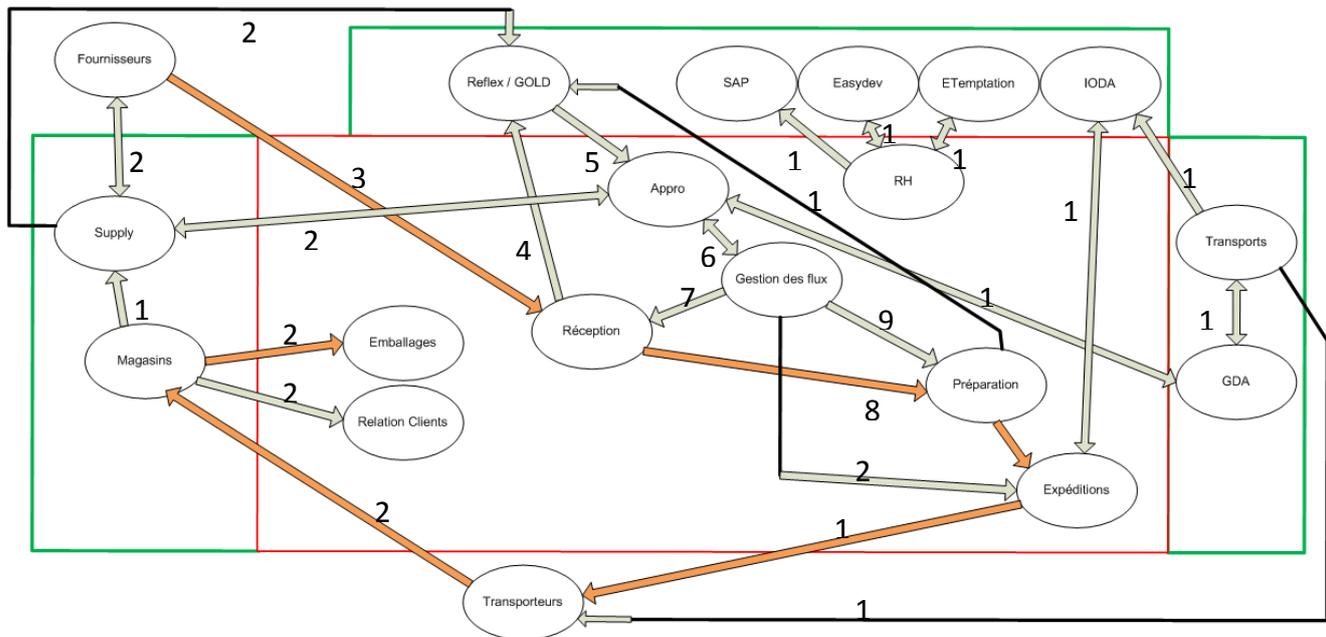


Figure 3 : Diagramme des flux existants

Le schéma ci-dessus présente une vision globale simplifiée des différents flux du sur-système Casino est ici représenté par un rectangle vert. Le rectangle rouge représente celui d'Easydis. A noter également la présence en orange des flux physiques et en gris des flux d'information.

Chacun de ces flux est numéroté dans un ordre chronologique par rapport au scénario du traitement d'une commande magasin.

Ci-dessous, le détail de chacun de ces flux :

1. Commande groupée aux fournisseurs
2. Livraison de la marchandise commandée
3. Remontée des PDA de la réception
4. Descente des ODP (ordre de préparation) correspondant aux commandes magasins
5. Les approvisionneurs font redescendre les ODP à la gestion des flux

6. La gestion des flux contrôle
7. Transfert de la marchandise réceptionnée pour préparation
8. Gestion de l'activité Boucherie/Volaille – Gestion des implantations
9. Remontée de la préparation vocale
10. Entrée de la compatibilité
11. Gestion des productivités
12. Saisie des Heures
13. Gestion des rythmes de livraisons / TACA
14. Récupération des groupages
15. Planification des groupages
16. Gestion des rythmes de livraisons
17. Expédition de la marchandise préparée
18. Affrètement
19. Livraison des magasins
20. En cas de réclamation : Prise de contact avec la relation client
21. Retour des supports vides depuis les magasins
22. Les approvisionneurs communiquent avec la supply
23. Contrôle

Nous retrouvons donc tous les services présentés précédemment, ainsi que les outils WMS et ERP, mais également des éléments nouveaux comme Easydev, que nous présenterons dans la section suivante, et IODA (le TMS) de la société.

Comme spécifié, les transports sont autonomes dans leur fonctionnement, et même si le BRT sud se situe physiquement sur le site d'Eurocentre, nous n'avons que très peu d'interactions directes avec eux. C'est pourquoi cette analyse ne détaillera en rien cette partie du système de l'entreprise.

Nous avons vu la partie centralisée du système d'information constitué des ERP et WMS. Toutefois, celui-ci s'avère incomplet. De nombreuses opérations réalisées en entrepôts ne sont pas gérées par ces systèmes. Pour pallier à cela, chaque entrepôt a développé un vaste nombre d'outils, principalement sous Excel, complètement indépendamment des uns des autres. Il en résulte, au niveau du siège, un contrôle impossible des outils mais également une vision centralisée incomplète.

L'étude réalisée a permis de dresser une liste, malgré tout non exhaustive, des outils Excel présents sur Eurocentre.

Notons la présence des documents suivants :

- Ressources Humaines
  - Tableaux de bord Journaliers (Gestion)
  - Tableaux de bord Productivité
  - Gestion Intérimaires
  - Tableaux de bord qualité (Taux de service)
  - Registre du personnel
- Exploitation frais/épicerie
  - Plannings par services
  - Gestion des expéditions (lignage)
  - Tableaux de bord de la gestion des flux
  - Stock emballages
  - Attribution matériel
  - Gestion des litiges
  - Gestion des clefs magasins
  - Polyvalence
  - Suivis Fournisseurs
  - Advanced Planning System

Parmi ces documents, on constate la présence de nombreux tableaux de bords : ceux-ci constituent la majorité des outils décisionnels utilisés. On trouve également des outils plus spécifiques à l'organisation interne du site, c'est par exemple le cas du document de gestion du lignage.

Même en interne sur un site unique, certains outils sont redondants entre les différentes entités organisationnelles.

### **I.2.3 Les problématiques rencontrées**

Nous avons vu que le système d'information de la Supply Chain de Casino repose sur GOLD, qui offre une visibilité globale toutefois insuffisante.

Certes, le groupe fonctionne de manière performante, et dispose donc déjà d'indicateurs de performances centralisés et efficaces. Mais la visibilité, notamment sur l'organisation interne des différents sites, est très réduite. Il faut ajouter à cela que les indicateurs existants ne sont pas toujours adaptés pour le pilotage d'un site, ce qui a pour conséquence de voir émerger, sur chacun d'eux, des outils complètement uniques, réalisés sous Excel.

Actuellement, aucune communication inter sites n'existe concernant le fonctionnement et l'organisation d'un site, notamment sur son pilotage.

Nous avons vu que l'intégration d'un ERP a pour premier objectif la standardisation pour adopter un modèle de processus. Mais plus que cela, standardiser des outils jugés performant permet le partage de ressources internes au groupe et donc un gain potentiel de performance.

Ce raisonnement a toutefois des limites et certaines particularités liées à une organisation sont trop spécifiques pour gagner à être standardisées. Il est donc toutefois normal que de tels outils existent pour gérer ces cas.

Il faut favoriser des outils standards qui offrent une visibilité et une performance théoriquement accrues.

Il faut ajouter à cela que l'ergonomie des outils et notamment des indicateurs de performance est d'une réelle importance. Certaines technologies permettent de réaliser des outils plus ou moins ergonomiques et performants.

### **I.3 Propositions de solutions aux problématiques**

Il faut commencer par poser le problème comme un problème de systémique. Le système d'information est le sur-système. Il se compose alors de systèmes tels que les ERP, WMS, TMS que nous avons vus, sans oublier les outils Excel qui répondent à des besoins du sur-système.

Il existe plusieurs solutions à la problématique système d'information telle que la standardisation des outils : il est possible de faire évoluer un système existant pour intégrer des services supplémentaires, ou bien de réaliser et intégrer un nouveau système. Deux options majeures donc : l'achat et l'intégration, ou la réalisation et l'intégration. Dans les deux cas, on retrouvera l'intégration d'un ensemble de services qui doivent venir combler des besoins du système d'information.

Il est évident que, dans les deux cas, ces opérations impliquent un investissement financier plus ou moins important, et nécessite une analyse poussée des besoins et de l'intégration entre systèmes.

Une intégration implique plusieurs problématiques qu'il advient de résoudre. Il faut veiller, si nécessaire, à l'interopérabilité avec les autres systèmes, et donc concevoir une architecture en conséquence. Mais il faut avant tout gérer les problèmes liés à l'hétérogénéité des données.

Pour répondre aux problématiques évoquées dans le cadre du système d'information de Casino, nous avons choisi de réaliser en interne un système répondant aux besoins. L'un des services majeurs de ce système est celui de la planification. Pour cette raison, nous l'avons nommé Advanced Planning System (APS).

## **II Initialisation de la réalisation d'un système intégré**

### **II.1 Analyse des besoins**

La définition des besoins est la première étape pour la réalisation d'un système. Elle se décompose en deux étapes :

- l'analyse du problème, qui doit permettre de déterminer une finalité pour le système
- la spécification du besoin, qui permet de définir un référentiel des exigences

#### **II.1.1 Les approches d'analyse**

Il existe deux approches pour récupérer des besoins métier. Il est possible d'adopter une approche descendante à partir d'un problème donné, ou bien une approche montante à partir d'un système existant.

L'approche descendante part du domaine du problème et vise à le définir, ainsi que les besoins et contraintes qui le composent. A l'inverse, l'approche montante part du domaine de la solution, et vise à construire le système en intégrant ses constituants existants.

Le choix de l'approche dépend évidemment de la configuration de votre solution. Il se peut également qu'il soit nécessaire d'utiliser les deux approches. C'est notamment ce que nous avons réalisé dans le cadre du projet APS.

#### **II.1.2 Méthodes de définition**

L'initialisation d'un projet est due à l'apparition d'un problème duquel découlent des besoins émanant du métier. C'est donc à partir d'interviews réalisés avec le métier qu'on les détermine.

Ces éléments sont ensuite raffinés avec un expert provenant du métier et l'architecte du système, ce qui permet de définir ses exigences.

Le recueil des besoins est une étape cruciale pour la réalisation d'un système aussi complexe soit-il. L'importance de cette étape varie toutefois en fonction du modèle de développement choisi. Une mauvaise définition des besoins entrainera un échec du projet, en tout cas il ne satisfera pas le client final comme escompté.

Pour notre projet, nous avons choisi de suivre une méthode agile, notamment pour réduire les risques lors la définition des exigences.

### **II.1.3 Besoins identifiés**

Dans le cadre du projet APS, nous avons réalisé des interviews de l'ensemble des services d'un site.

Comme l'analyse du système d'information l'avait déjà mis en lumière, le grand nombre d'outils Excel génère un fort besoin de simplification, une simplification non seulement des outils eux même qui manquent d'automatisation, mais également de l'ensemble, dans le sens où il est devenu trop complexe d'accéder à une source unique d'information. Vu à plus haut niveau, les même outils génèrent un besoin de standardisation dans le but de partager ces ressources, et de récupérer un nouvel ensemble d'informations centralisées.

Le plus gros de l'analyse s'est porté sur des outils existants, avec pour objectif la standardisation et la simplification. Toutefois, pour ne rien omettre, une analyse globale des différents processus métier est recommandée. Dans le cadre du projet, nous avons profité d'une démarche de Lean Management pour récupérer l'étude de ces processus dont font partie les différents flux d'information.

Finalement, nous avons regroupé les exigences par sous-système qui sont détaillés ci-après.

### ***Prévision journalière :***

La réalisation d'une prévision journalière est la finalité auquel l'APS Excel répondait : celle de permettre aux décideurs (maitrises et cadres) de visualiser une journée d'activité à J+1 afin d'optimiser celle-ci.

L'étude a montré un besoin d'ajout d'indicateurs de suivis, notamment visuels, comme :

- ✓ La productivité exploitation
- ✓ La productivité des préparateurs

Mais l'intérêt principal d'une intégration de cet outil dans le nouveau système est motivé par sa standardisation, qui doit permettre d'avoir une vision globale de l'ensemble des prévisions réalisées par les différents sites.

De plus, il a été formulé le besoin de réaliser des prévisions sur de plus longues périodes, comme à la semaine ou au mois. Un outil tel que celui-là offrirait une nouvelle capacité d'anticipation pour les décideurs, notamment dans le but de planifier les ressources humaines.

### ***Planification :***

Dans l'analyse du système d'information du groupe, nous avons vu que la planification des ressources humaines est géré à deux niveaux, dans un premier temps grâce à des plannings Excel personnalisés par chaque service, dans un second temps plannings récupérés et saisis sur Etemptation par le service des ressources humaines.

Le premier besoin relevé est une demande de simplification pour les RH qui consacrent un temps conséquent à la saisie des plannings. Le système doit alors centraliser et

proposer directement l'ensemble des informations, notamment récapitulatives, pour faciliter ces saisies.

Les plannings sont une source d'informations utiles à de nombreux autres systèmes. Une partie de ces données est accessible de manière globale, car centralisée sur Etemptation. La problématique est que le modèle du domaine sur lequel se base Etemptation ne contient pas toutes les données nécessaires à l'alimentation du système.

Ne pouvant utiliser comme source Etemptation, notre système a besoin d'un sous-système de planification pour fonctionner.

En plus de ce besoin émis par le système lui-même, on retrouve un besoin de standardisation de l'outil, et principalement de ses indicateurs sur le personnel.

#### ***Demande intérimaire :***

Les demandes d'intérimaire suivent un processus hiérarchique de validation qui n'est pas automatisé. Les problèmes relevés ont amené un besoin de restructuration de ce processus par un outil.

En plus de l'automatisation des étapes de validation, l'outil a besoin d'intégrer un ensemble d'indicateurs permettant de valider ces demandes. Pour cela, les principaux éléments à prendre en compte sont le budget alloué et les commentaires justificatifs.

#### ***Formation :***

Ici le besoin se situe surtout sur l'archivage de toutes les formations réalisées et l'amélioration de leur suivi. Auparavant gérés sous Excel/VBA, un certain nombre de tâches avaient déjà été automatisées, comme le publipostage des convocations. Toutefois, certaines opérations, comme la proposition des formations à planifier pour chaque salarié avec des ordres de priorité, s'avéraient compliquées.

Un autre besoin était de permettre d'afficher les formations du personnel de manière automatique dans les plannings des services et donc de centraliser l'information.

### ***Suivi du personnel :***

Le suivi du personnel, comme on a pu le voir, est fait en grande partie sur Excel, notamment sur la base d'extractions réalisées depuis Etemptation. Les outils disponibles sont assez complets et ont été simplifiés grâce à l'utilisation du VBA. Toutefois, les maitrises n'ont pas accès à ces informations, certaines étant sensibles, comme les salaires par exemple. Bon nombre de ces outils se voient réservés aux RH. Les maitrises se retrouvent alors à devoir appeler régulièrement le service RH pour des demandes d'information.

Le besoin serait ici de rendre accessible certaines informations pour tous, afin de réduire les flux de demandes d'informations. De plus, certains outils bénéficieraient d'une intégration au sein du système, ce qui permettrait de proposer une version centralisée et standardisée de ceux-ci. C'est le cas de la Gestion Prévisionnelle des Emplois et des Compétences (GPEC).

#### **II.1.4 Méthode de suivis des exigences**

Je cite « Une exigence exprime une capacité ou une contrainte à satisfaire par un système. Elle peut exprimer une fonction que devra réaliser le système ou une condition de performance technique, physique, de sécurité, de fiabilité, d'ergonomie, d'esthétisme... ».

Les exigences sont un élément clé porté sur le cahier des charges fonctionnel.

Deux méthodes existent pour assurer le suivi des contraintes et des exigences. Un formalisme textuel est le plus souvent utilisé. Mais il est également possible lors de la réalisation d'ingénierie approchée par les modèles, et grâce au SysML, de les représenter par un diagramme aussi nommé « requirement diagram ».

Le diagramme d'exigences permet tout au long d'un projet de relier les exigences avec d'autres types d'éléments SysML par plusieurs relations.

Lors du projet, nous avons fait le choix d'adopter le formalisme textuel.

Pour assurer le suivi, nous avons simplement utilisé un outil réalisé grâce à Excel dont voici un extrait :

Système	F5	oui	Le système doit permettre à un administrateur non informaticien de configurer l'organisation d'un site
	C5.1	oui	Le système doit pouvoir être déployé sur tous les sites de la société
	C5.2	oui	Le système doit être centralisable
	F6	oui	Le système doit permettre d'identifier les utilisateurs
	C6.1	oui	Les accès des utilisateurs au sein du système doivent être configurables par un administrateur non informaticien
	F9	oui	Le système doit permettre de réaliser des simulations d'une journée d'activité d'une exploitation
Services previsionnel activité	C9.1	oui	Ces simulations doivent pouvoir comprendre des commentaires
	F10	oui	Le système doit permettre de visualiser ces simulations (F9) sous la forme d'un tableau de bord
	C10.1	oui	Ces simulations doivent être conservée et rester consultables à tout moment par les utilisateurs
	C10.2	oui	Le tableau de bord doit être synthétique et présenter les informations concernant : les heures, les uo, la productivité et la performance
	C10.3	non	Le tableau doit être paramétrable pour pouvoir afficher les simulations suivant 4 niveaux : exploitation, site, région et France
	F41	non	Le système doit être capable d'évaluer une journée d'activité d'une exploitation
Services previsionnel activité	C41.1	non	Le système doit permettre de comparer l'évaluation avec la simulation réalisée pour la même journée d'activité
	F50	non	Le système doit permettre de comparer l'activité d'une exploitation entre deux années
	C50.1	non	Le système doit prendre en compte les périodes de vacances pour permettre de comparer les pics d'activité

Figure 4 : Extrait de l'outil de suivi des exigences

On voit que chaque fonction notée en bleu peut posséder des contraintes, alors notées en orange. Elles sont classées par « catégorie » pour une meilleure lisibilité. Nous aborderons plus précisément le rôle de ce document dans la méthode de travail, plus loin dans ce dossier.

Ainsi, ce sont 54 fonctions et 40 contraintes qui ont émergé de l'analyse des besoins comme exigences pour le système.

Ce document ne permet toutefois pas d'évaluer la charge de travail. Il permet d'établir des priorités pour les différents sprints, afin de s'assurer que rien n'est oublié.

## II.2 Définition du projet

### II.2.1 Buts et objectifs du projet

N°	Buts	Objectifs	Résultats opérationnels
1	Standardisation du pilotage	<ul style="list-style-type: none"><li>Uniformiser les outils sur les entrepôts</li></ul>	<ul style="list-style-type: none"><li>Vision globale</li></ul>
2	Réduction des tâches administratives	<ul style="list-style-type: none"><li>Optimisation du système d'information</li></ul>	<ul style="list-style-type: none"><li>Réduction du temps consacré aux tâches de pilotage</li></ul>
3	Amélioration des performances sur SITE	<ul style="list-style-type: none"><li>Refonte des KPI</li></ul>	<ul style="list-style-type: none"><li>Gain de productivité</li><li>Gain de visibilité</li></ul>

Tableau 1: Buts, Objectifs et Résultats opérationnels

Le tableau ci-dessus récapitule les buts, objectifs et résultats opérationnels visés par le projet Advanced Planning System.

Pour le siège, l'enjeu est de centraliser un maximum d'outils et ainsi bénéficier d'informations détaillées sur le fonctionnement interne des sites, mais également de permettre un partage des outils entre site. Cela est représenté par le but de « Standardisation du pilotage », mais également celui d'amélioration des performances.

Les entrepôts bénéficieront de nouveaux indicateurs ainsi que d'une refonte d'indicateurs existants qui doivent leur permettre d'affiner leur pilotage. De plus, l'intégration de ce système simplifiera significativement le système d'information propre à leur site.

Finalement on souhaite remédier aux défauts constatés d'Excel, notamment sa solidité et sa faible automatisation, dans le but de réduire les tâches administratives.

## II.2.2 Définition de la portée

Pour présenter la portée, nous l'avons partagée en deux domaines : un qui a pour fonction la standardisation du pilotage au niveau de l'exploitation, le second se consacre aux fonctions des ressources humaines.

### ***Domaine des exploitations :***

---

- Outil de pilotage des équipes de préparation (EasyProd)
- Ensemble de tableaux de bord pour le pilotage à différents niveaux que sont le groupe de services géré par un maitrise de niveau 6, la gestion de l'exploitation par un cadre niveau 7, mais également pour les niveaux supérieurs, les visions globales Site, Région et France.
- Un système de planification des effectifs, intégrant la notion de polyvalence.
- Un système de gestion des intérimaires allant de la planification de ceux-ci en passant par le processus d'embauche jusqu'à leur suivi complet, conformément à notre projection budgétaire.

### ***Domaine des ressources humaines :***

---

- Outil de communication interne au SITE
- Outil de gestion des compétences
- Outil de gestion des formations
- Outil de synthétisation des données planifiées via le premier système
- Outil de gestion des mesures applicables au personnel

Vu globalement, le projet doit être la solution à une problématique de standardisation des outils et offrir donc de nouvelles possibilités quant à la vision de haut niveau sur le pilotage des entrepôts.

Vu localement par un site, il doit être un nouveau moyen de communication directement intégré dans son Système d'Information, mais également un outil d'aide à la prise de décisions pour tous les niveaux du pilotage. Par la même, il doit offrir de nouvelles possibilités quant à l'analyse et à l'exploitation des données métiers.

L'objectif est que tous les entrepôts disposent d'un ensemble commun d'outils nécessaires pour piloter leur activité.

### II.2.3 Produits livrables

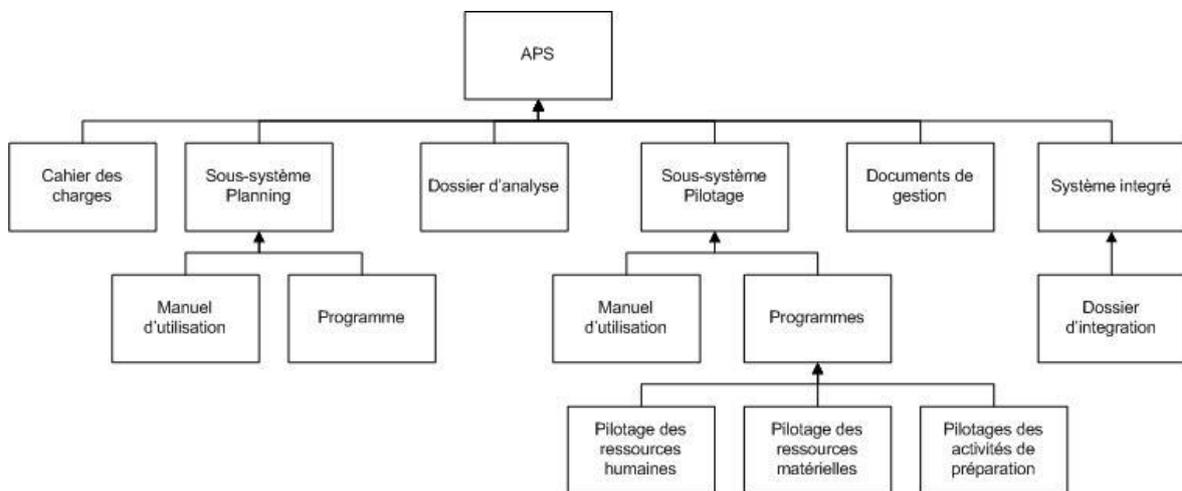


Figure 5 : Product Breakdown Structure - APS

Ci-dessus un Product Breakdown Structure (PBS) global du projet. Les livrables du projet ont été définis au début du projet et son représentés par le schéma ci-dessus. Il présente assez globalement les livrables, surtout au niveau des parties logicielles, celles-ci ayant évolué avec le projet.

On retrouve ainsi les éléments constituant la documentation du projet sur trois niveaux, ainsi que deux sous-systèmes, les plannings et le pilotage. On ne voit pas par contre ici le détail des différentes versions réalisées comme des livrables.

Le sous-système de pilotage est détaillé en trois sous programmes assez globaux qui regroupent les fonctions dédiées aux ressources humaines, aux exploitations et un élément que nous n'avons pas encore abordé, la gestion du matériel. Cette sous partie a en réalité été mise de côté et n'a finalement pas été ajoutée au cahier des charges du projet. Toutefois, une étude avait été réalisée et se présente comme un bon axe d'amélioration pour le projet.

#### **II.2.4 Etude des risques**

La première règle avant de débiter l'étude des risques est de terminer la définition du système. Une fois cela accompli, il est possible de passer à l'étape de l'identification des risques.

Lors de cette étape, il faut décomposer le système en sous-systèmes, composants et fonctions. Il faut ensuite identifier les modes de défaillance, puis établir les conséquences possibles.

On identifie et modélise les risques grâce à la représentation prédictive du fonctionnement du système et des relations de causalité entre chaque risque. Ils apparaissent notamment dans les scénarios obtenus à partir des cas d'utilisation du système.

L'ensemble des risques obtenus, grâce à ces modèles prédictifs, est ensuite analysé quantitativement et qualitativement. L'analyse quantitative consiste en l'attribution de probabilité d'occurrence, alors que l'analyse qualitative cherche à affecter un niveau d'importance.

Il existe un grand nombre de méthodes d'analyse des risques.

L'étude réalisée pour l'APS a permis d'établir un risque commun mais avéré dans le cadre de plusieurs projets au contexte similaire : l'acceptation au changement des utilisateurs.

En effet, il a été établi que la mise en place forcée d'un ensemble d'outils, comme ceux intégrés dans le système, n'aboutirait qu'à un refus et un échec.

Pour réduire ce risque, nous avons favorisé une implication directe des utilisateurs lors de la conception du système. De plus, il a été établi que la formation serait un facteur important pour la réussite du projet.

### II.3 Structure de l'équipe de projet

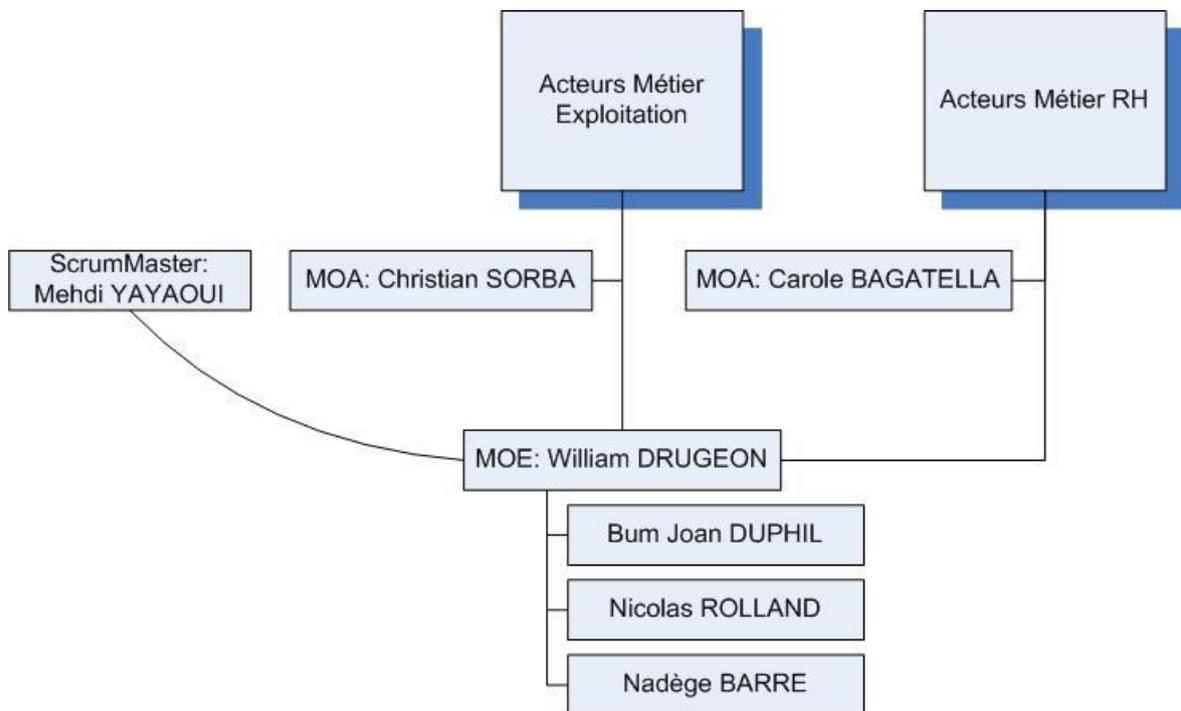


Figure 6 : Schéma organisationnel

Le schéma ci-dessus présente les différents acteurs du projet. Au commencement de celui-ci, les seuls acteurs étaient M. SORBA, mon responsable direct et moi-même. Cependant, avec l'agrandissement du périmètre du projet, l'équipe s'est agrandie. Il y a eu notamment l'intégration de Mme BAGATELLA comme MOA pour le métier RH et de M. YAYAOUI (cadre méthodes et organisations) pour l'encadrement méthodologique.

Ultérieurement, au cours du projet, j'ai eu à intégrer, former et gérer trois développeurs. Ces derniers ont aujourd'hui la charge de sa continuité. M. DUPHIL, anciennement affecté aux ressources humaines pour développer des outils spécifiques, s'est vu chargé de

développer le sous-système RH notamment. M.ROLLAND assurera mon remplacement et à la charge d'achever le projet. Pour finir, Mlle BARRE a participé aux dernières phases du développement du projet.

## **III Etudes et conception du système logiciel**

### **III.1 Etude de faisabilité**

#### **III.1.1 Etude des solutions technologiques**

Après l'étude préliminaire, une solution type « client lourd » a rapidement été exclue, et ce pour plusieurs raisons.

Tout d'abord il s'avère difficile de déployer des logiciels sur l'ensemble des postes. En effet, cela nécessite une intervention de la part de CIT qui n'a pas de temps alloué au maintien d'application supplémentaire de ce type.

Ensuite, l'architecture du système d'information tend clairement à s'orienter vers des services web et du « client léger ». En conséquence de quoi le choix d'une technologie web était tout indiqué.

Les technologies les plus utilisées pour développer des applications de ce type sont certainement Java EE et PHP. Ce sont deux technologies maintenant éprouvées qui disposent de très grandes communautés et d'une utilisation libre de droit. CIT dispose de plusieurs compétences pour le maintien de certaines applications composant le système d'information, notamment en Java.

La dernière version de PHP est numéroté 5. Elle permet une conception objet, ce qui n'était pas le cas des versions précédentes. Il faut savoir que le modèle objet de PHP a été fortement inspiré de celui de Java : on retrouve donc certains concepts et certaines règles communes, comme par exemple l'interdiction de l'héritage multiple, pour ne citer que celle-là. Cependant, et c'est probablement le défaut de PHP le plus souvent cité, c'est un langage plus permissif que Java, ce qui peut avoir des conséquences concernant notamment la maintenabilité du système.

Toutefois, comme on le fait en Java, on suit généralement un modèle type « modèle vue contrôleur version 2 », qui se veut structurant et permet donc une forte augmentation de la maintenabilité du système. Appliquer ces modèles sans utiliser de framework en PHP est possible mais très complexe, et reviendrait à en développer un finalement. Pour cela, il existe plusieurs Frameworks comme Zend et Symfony, pour ne citer que les plus connus. Zend nécessite d'acheter une licence pour être utilisée, mais dispose d'un support professionnel, alors que Symfony, aujourd'hui en version 2.5, est totalement gratuit, et se base plus sur sa communauté au niveau support, même si SensioLabs, qui en est à l'origine, apporte de constantes améliorations à celui-ci.

### ***Présentation du Modèle MVC :***

Le Model-View-Controller, est un modèle de conception. Il est donc complètement indépendant des langages de programmation. Très répandu il fut introduit dans les années 80. Une seconde version existe aujourd'hui : le MVC2.

Comme son nom l'indique clairement, le principe de ce modèle est assez simple : il consiste à segmenter l'application en trois couches logicielles :

- ✓ Le modèle
- ✓ La vue
- ✓ Le contrôleur

Une fois segmentée, ces trois composants ont des rôles bien établis.

Le modèle est une représentation des données et de l'ensemble des règles métier. C'est la seule couche autorisée à accéder aux données directement stockées en base par exemple.

La vue elle, a la charge de réaliser la mise en forme de ces données. Elle représente l'IHM, et récupère donc également les informations saisies et les actions effectuées par l'utilisateur.

Pour finir, le contrôleur est une sorte de routeur. Il se charge d'intercepter les requêtes de l'utilisateur, d'appeler le modèle, puis finalement de rediriger sur la vue : l'IHM. En fait, il ne fait que de l'interception et de la redirection.

Dans sa première version, le MVC nécessitait d'implémenter une multitude de contrôleurs, ce qui pouvait être lourd à mettre en place. C'est pourquoi, et c'est bien là l'amélioration apportée par sa deuxième version, le MVC2 ne dispose lui que d'un seul contrôleur qui se charge donc d'intercepter l'ensemble des requêtes et de les rediriger. C'est cela que les Frameworks, que ce soit en PHP ou JEE, permettent de réaliser facilement en plus d'offrir de nombreux autres services.

Représentation du modèle :

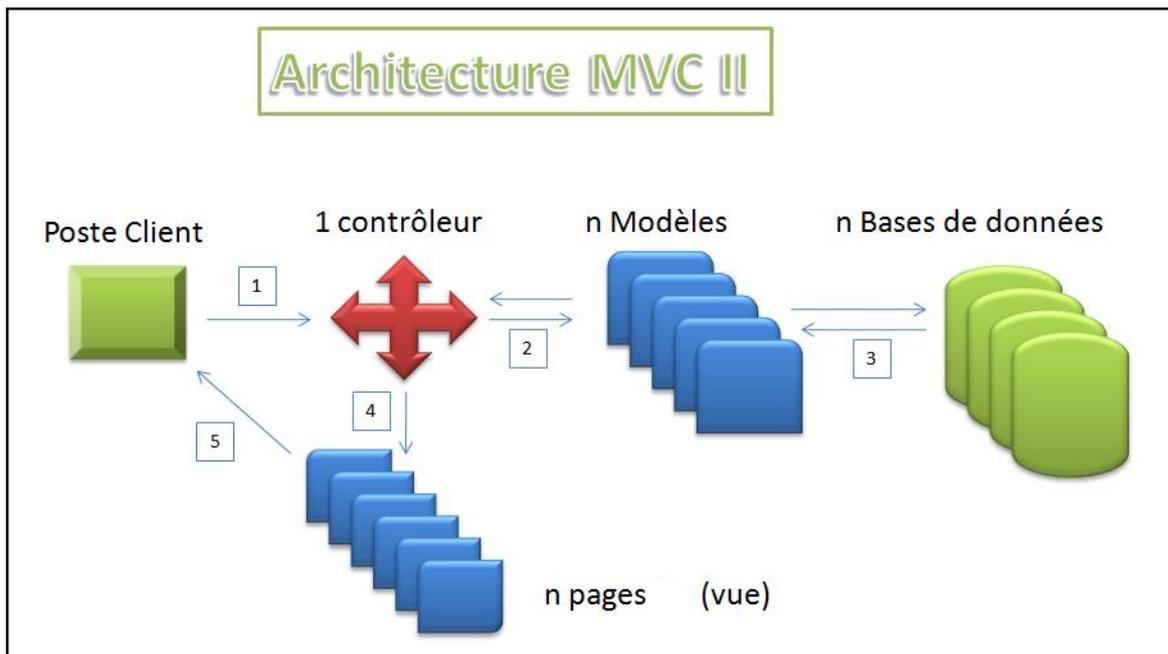


Figure 7: Schéma architecture MVC II

Cinématique du schéma ci-dessus :

1. Le client émet une requête
2. Le contrôleur intercepte la requête puis la redirige vers la partie du modèle correspondante
3. Le modèle effectue les transactions nécessaires avec la ou les bases de données, puis applique les règles métier

4. Le contrôleur sectionne la vue et lui transmet les données
5. La vue met en forme les données : elle construit l'IHM

Concrètement, ce modèle cherche à proscrire certaines pratiques de développement, celles qui visent à mettre du code de traitement dans des composants de présentation (JSP, PHP, ASP...). Cela pour plusieurs raisons.

Tout d'abord, la factorisation des traitements permet de découpler totalement l'IHM du modèle, et donc apporte de la flexibilité à l'application, ainsi que de la réutilisabilité. Ainsi, si une règle métier change, il suffira de la modifier une fois pour que l'ensemble des vues en bénéficie.

De plus, cela facilite le travail en équipe, notamment avec des spécialistes comme les infographistes qui ne voient alors que la partie présentation, et ne sont donc pas gênés par les traitements sur lesquels on peut travailler en parallèle.

Le contrôleur lui aussi permet une plus grande souplesse dans l'application. C'est lui qui assemble les différents éléments du système à partir d'une requête.

Toutefois, le MVC n'a pas que des avantages. Il peut se révéler trop complexe pour des applications de petite envergure. En conséquence de quoi le temps accordé à l'architecture peut ne pas être rentable. De plus, la séparation en trois couches implique tout de même l'augmentation du nombre de micro composants, malgré la factorisation du code.

En conclusion, le MVC favorise les bonnes pratiques de développement et la maintenabilité. Sur de gros projets, avec ou sans grandes équipes de développement, une architecture telle que celle-ci s'avère indispensable.

Il n'est toutefois pas le seul modèle existant, même si c'est le plus répandu. On peut citer par exemple, SOA pour Service Oriented Architecture, aussi nommé en français « architecture orientée services ». Celui-ci met en avant les problématiques

d'interopérabilité, de réutilisabilité, de réduction de couplage entre les systèmes, et cherche à les résoudre en se basant sur des standards tels que les très populaires « Services Web ».

### **III.1.2 Comparatif et choix de la solution**

On ne peut comparer un langage et un Framework. Il est donc préférable de comparer Symfony avec Struts par exemple, qui est un Framework permettant l'implémentation d'une architecture MVC2 en JEE. Sans rentrer dans les détails, les deux Frameworks proposent des services très similaires, même si on peut noter de légères différences sur la gestion du routing, par exemple. Toutefois, ces Frameworks ont un même objectif et permettent donc de mettre en place des classes « Controller » ainsi que des « Actions », qui sont au cœur de ces Frameworks et du MVC.

Nous allons voir qu'en réalité les services proposés par ces Frameworks sont semblables en bien des points.

Pour commencer par la vue, nous avons noté que les bonnes pratiques sont de ne pas y écrire de code Java ou PHP. Pour cela, il existe des libraires telles que la JSTL en Java, et des moteurs de Template tels que Twig en PHP. Ce sont des langages simples, à part. Ceux-ci, une fois compilés, transcrivent ensuite du Java ou du PHP. Ils offrent de nombreux avantages, comme la protection des affichages, ou bien même, pour Twig, l'héritage de Template par exemple. Ce dernier permet, comme pour des classes, de définir des blocks surchargeables, et ainsi « d'emboîter » des Templates très facilement.

Au niveau des services proposés, on retrouvera les notions de « Form », qui permettent de factoriser le code des formulaires HTML. Le reste est lié aux classes principales des Frameworks que sont les « Controller ». Ces contrôleurs font bien parti de la couche du modèle au niveau du MVC. En effet, la couche du « Controller » est complètement prise en charge par les Frameworks, et seul un fichier de configuration des routes reste à la charge du développeur. Ainsi, de nombreux services de redirection, de génération de

Template, et plus généralement de gestion des requêtes sont offerts dans les classes héritant de « Controller ».

Finalement, la principale différence réside dans leurs notions du routing, et leurs configurations.

### ***Intégration d'un ORM :***

Il faut savoir que par défaut Symfony intègre l'ORM Doctrine2 pour la gestion de la persistance. Il est également possible d'utiliser un ORM tel que JPA avec Struts.

Les ORM ont, comme toute chose, des avantages mais également des inconvénients. Ils permettent, par exemple, de générer depuis des entités une base de données, ou bien, depuis une base, l'ensemble des entités, en plus de gérer leur persistance. Doctrine est configurable, mais fonctionne par défaut de la même manière que JPA via un système d'annotation du code. Ce sont ces annotations qui permettent à l'ORM de transformer des classes en table dans une base de données.

La première force d'un ORM, celle complètement visée dans ce projet, c'est l'abstraction qu'il offre de la base de données utilisée : il permet de facilement en changer. Se pose toutefois toujours le problème de migration des données. Malgré cela, si le code au niveau des repositories a été réalisé proprement, la migration n'induit pas de modifications.

En effet, pour détailler plus précisément, les repositories sont l'équivalent des DAO : ce sont donc eux qui réalisent l'ensemble des requêtes que l'on souhaite écrire sur le modèle du domaine. La problématique se pose alors : chaque moteur de base de données ayant un langage SQL avec des mots clefs potentiellement propres, une même requête SQL peut s'exécuter par exemple sur MySQL mais pas sur Oracle.

Cependant, Doctrine offre une solution à cette problématique : le Data Query Language (DQL). En effet, le développeur peut ainsi, en utilisant uniquement du DQL, laisser Doctrine générer le SQL qui sera alors compatible avec le moteur configuré.

Il faut toutefois noter le principal désavantage de ce service : l'optimisation des requêtes. En effet, chaque moteur, notamment Oracle, permet d'écrire des requêtes SQL en utilisant des fonctions propres très performantes, que Doctrine ne saura utiliser. De plus c'est Doctrine qui génère le code SQL mais il peut parfois se révéler peu optimisé. Dans ce cas, il n'est d'autre choix que de passer directement par du SQL à travers celui-ci.

A l'utilisation, l'ORM apporte également un gain de temps non négligeable au niveau du développement, permettant ainsi de récupérer très simplement certaines données. Toutefois, ici encore, il faut faire très attention, car plusieurs requêtes peuvent être exécutées sans que l'on s'en rende réellement compte, et cela peut rapidement en générer un très grand nombre.

Un très bon exemple de ce phénomène est le suivant : lors de l'appel à un getter d'une entité, proposant de récupérer une instance d'une autre entité, Doctrine ira de lui-même sélectionner, en base de données, l'instance de la table concernée pour retourner l'élément. Il fait donc de manière autonome une requête Select lors de l'appel à ce getter. Il est tout à fait possible et même simple, grâce au débogueur inclu dans Symfony, de tracer toutes les requêtes effectuées pour réaliser le rendu d'une vue. Il faut alors revoir le code et utiliser des jointures qui permettront de ramener un ensemble de résultats alors exploitables avec des traitements.

### III.1.3 Architecture fonctionnelle

L'ensemble des fonctionnalités du système a été décrit via des cas d'utilisation, qui ont permis de réaliser un modèle du domaine complet.

**Piloter service :**

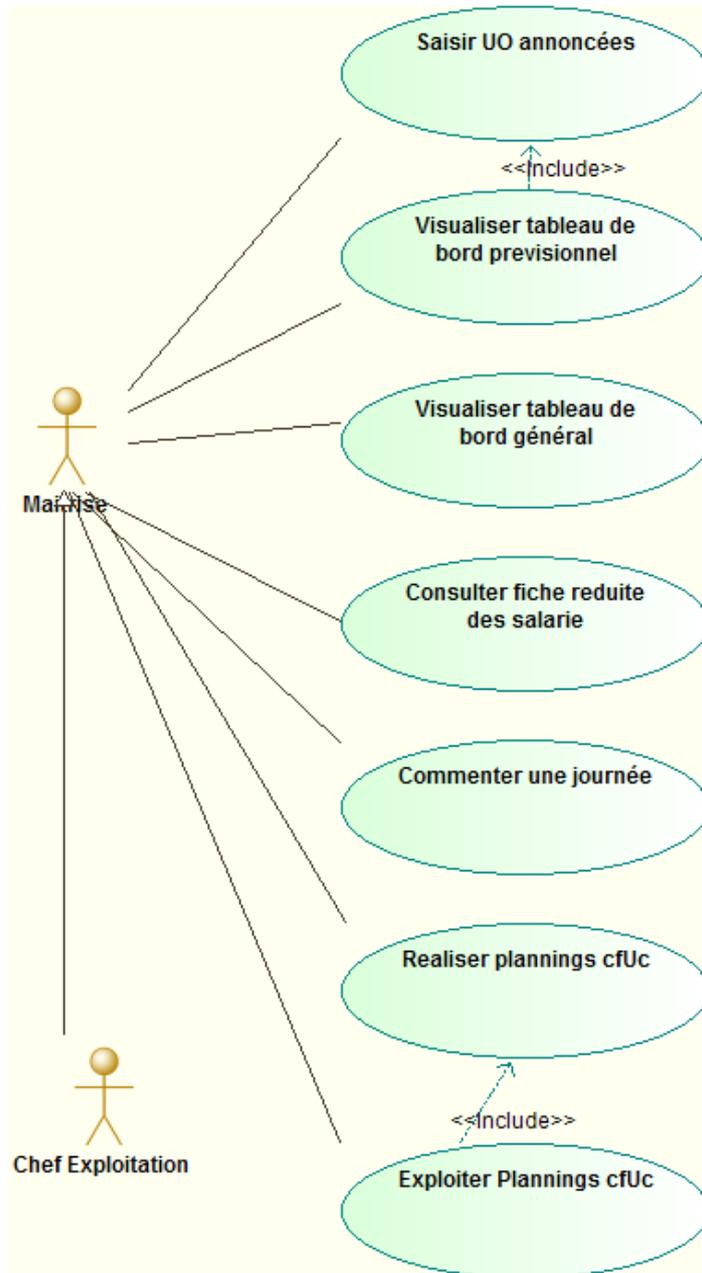


Figure 8 : Use case - Piloter Service

L'acteur principal ici est le maitrise. Il dispose d'un certain nombre de services pour l'assister au pilotage de son service. Ainsi, le cas d'utilisation montre un tableau de bord prévisionnel, qui est l'équivalent de l'APS Excel que nous avons évoqué précédemment. Celui-ci permet aux maitrises, après qu'ils aient saisi les UO annoncés, de retrouver un résultat pour le lendemain au niveau de l'exploitation.

Ce mécanisme de prévision serait très intéressant sur plus long terme, par exemple à la semaine ou au mois, et permettrait d'ajuster plus finement les ressources humaines, notamment intérimaires. Nous avons réalisé une étude sur cette fonctionnalité qui, expérimentée sous Excel, s'est révélée trop imprécise et imprédictible : le colisage y étant directement lié, cela s'apparente complètement à de la prévision de vente en magasin. Donc, même si certains calculs ou approximations s'avéraient précis certains mois, ils pouvaient être totalement erronés pour un autre. Cette fonctionnalité pour le système n'a donc pas été choisie.

Par contre, la supply nous annonce chaque jour les UO pour le lendemain. C'est sur cette base, ainsi que sur les heures placées grâce aux plannings, que la simulation a été réalisée. A partir de ce tableau de bord, un processus décrit par le système peut être mise en place pour son analyse. Celui-ci s'avère être assez simple, mais très important, pour réellement bénéficier des gains de l'analyse de cette prévision.

Le processus est le suivant : chaque maitrise réalise la prévision pour son service. Ensuite, l'ensemble des maitrises, ainsi que le chef d'exploitation, se réunissent pour un briefing journalier de la prévision du lendemain. C'est lors de cette réunion que les décideurs vont chercher à optimiser et aller chercher la performance, grâce à la polyvalence des équipes notamment. Sans cette réunion, nous avons pu constater que les gains étaient minimes.

En effet, les maitrises géraient déjà finement leurs services. Le problème est qu'il est nécessaire d'avoir un recul au niveau de l'exploitation pour ne pas être en déphasage avec le budget global.

C'est pour répondre à cette problématique que ce processus est maintenant décrit dans la documentation du système comme proposition de méthode d'utilisation de l'outil.

Tableaux de bords et fiche salarié :

Comme indicateur, les maitrises disposent de deux tableaux de bords :

- ✓ Le premier permet de visualiser la performance d'un groupe de service donné, et donc des informations telles que les budgets intérimaires mais également la performance objectivée
- ✓ Les services dits productifs disposent également d'un suivi détaillé de leur équipe, avec un suivi de la productivité par préparateur et par chemin de préparation

Par ailleurs, un ensemble de données fournies par les RH sont accessibles via une fiche salarié dans une version réduite qui présente, en plus des informations classiques (nom, adresse, téléphone..) des informations sur les sanctions, les formations et les bilans d'évaluation.

Pour finir, il est également possible de visualiser une série de commentaires saisis. Ils permettent en fin d'année, au moment de réaliser les budgets, de prendre en compte des éléments de décision supplémentaires pour cette tâche.

## **Plannings :**

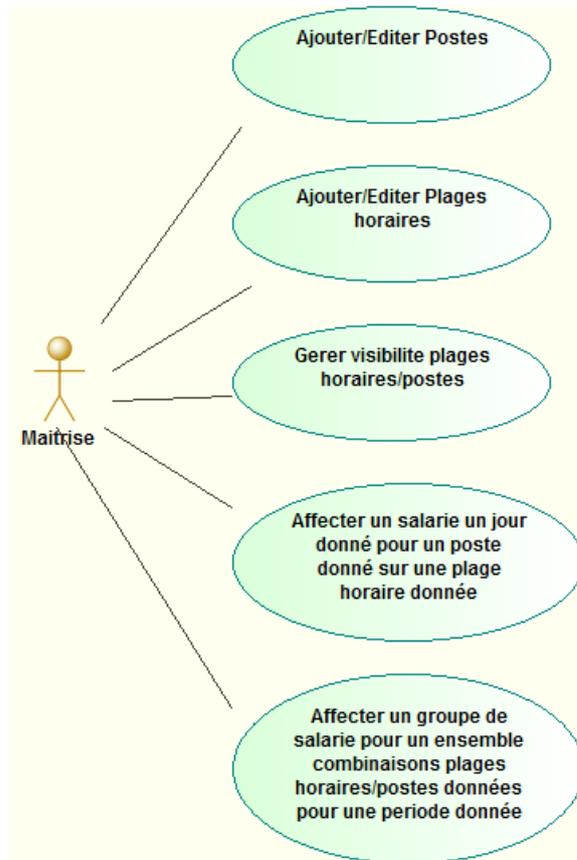


Figure 9 : Use case - Plannifier

Le cas d'utilisation ci-dessus présente les fonctions permettant la réalisation des plannings. On retrouve, comme acteur, le maitrise, qui peut affecter à des salariés une ou plusieurs combinaisons plage-horaire-poste pour une journée donnée.

Cela permet de créer une planification complète d'un groupe de services. La notion de visibilité est utilisée pour l'archivage des données, comme nous le verrons plus loin.

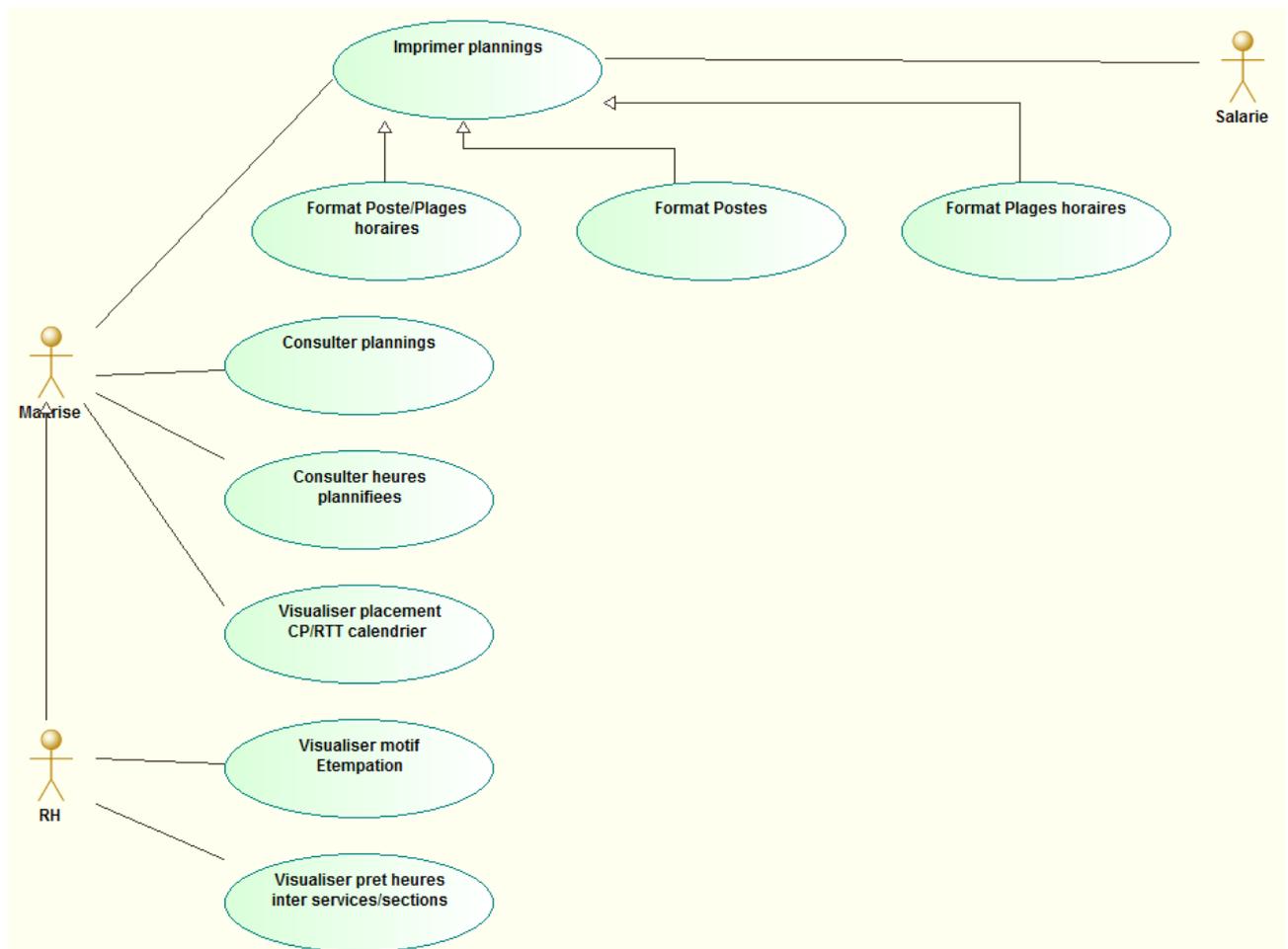


Figure 10 : Use case - Exploiter plannings

Les plannings sont une source de donnée centrale au système. Ils sont exploités grâce à des tableaux de bords utiles aux RH, et également directement par les maîtrises bien entendu. Il est ainsi possible de les imprimer sous différentes formes, de visualiser précisément les CP et RTT posés par un service, mais également de comparer les heures planifiées avec les budgets correspondants.

Formation :

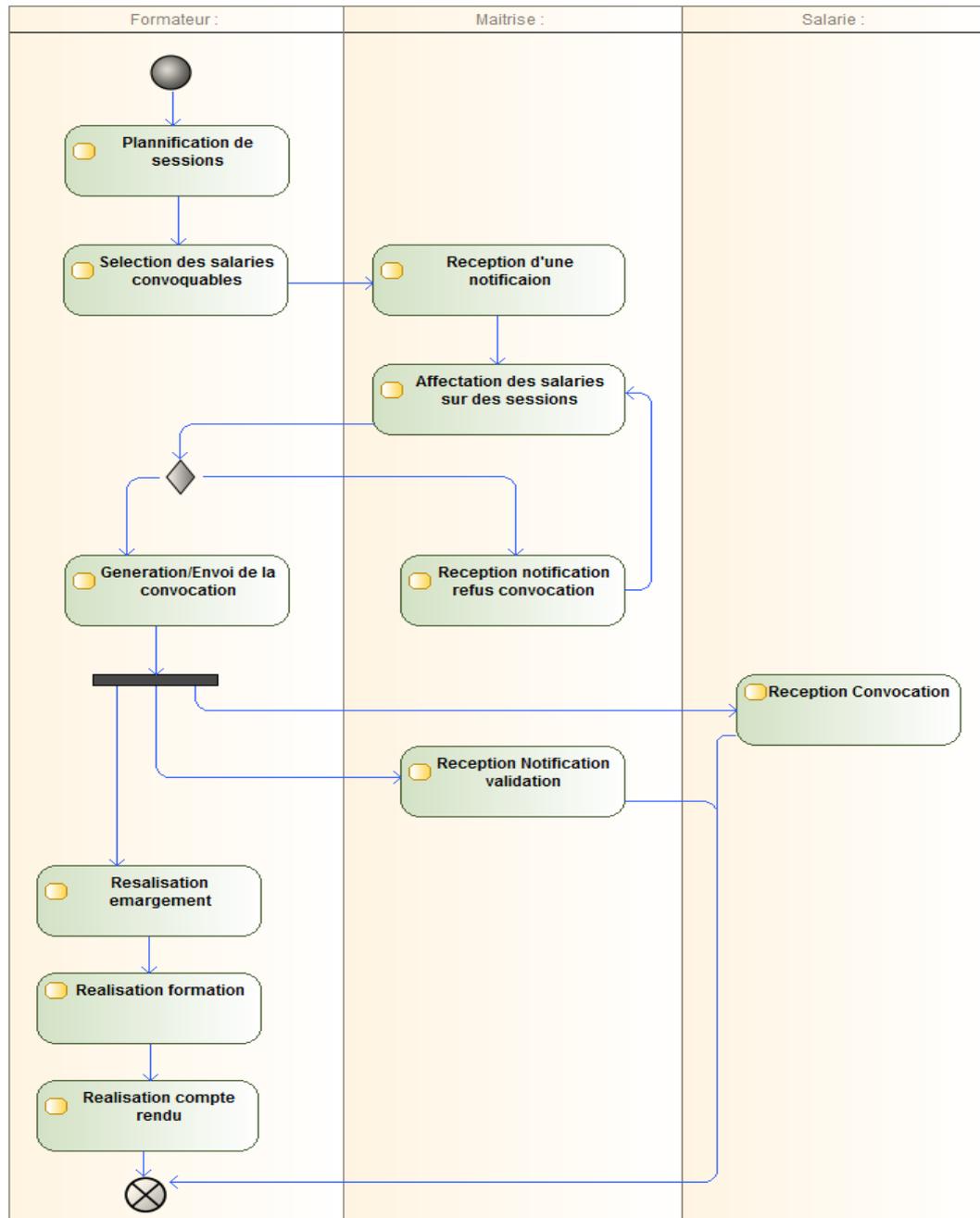


Figure 11: Activity diagram - Former salarie

Ici, le formateur va pouvoir créer des sessions de formation, pour un type de formation donné, qu'il pourra ensuite lier entre elles.

L'objectif est d'affecter une liste de salariés à former durant ces sessions, en laissant aux maitrises la possibilité de partager leurs effectifs dans les différentes sessions proposées. Chaque placement est validé par le formateur, ce qui aura pour conséquence de générer un document imprimable : la convocation.

Cette convocation est directement transmise au salarié, qui se voit alors dans l'obligation de suivre la formation.

Il est ensuite possible de réaliser le suivi, notamment de l'émargement, mais également de saisir des comptes rendu de formation pour archive.

Finalement, le formateur pourra consulter l'ensemble des formations réalisées, ainsi qu'une liste des formations à réaliser suggérée par le système. L'ensemble est archivé et consultable pour répondre au besoin précédemment évoqué d'inclure cet outil du système comme élément d'amélioration profitable à l'obtention de l'IFS. Les formations apparaissent également automatiquement dans l'ensemble des plannings des services concernés.

## Gestion des intérimaires :

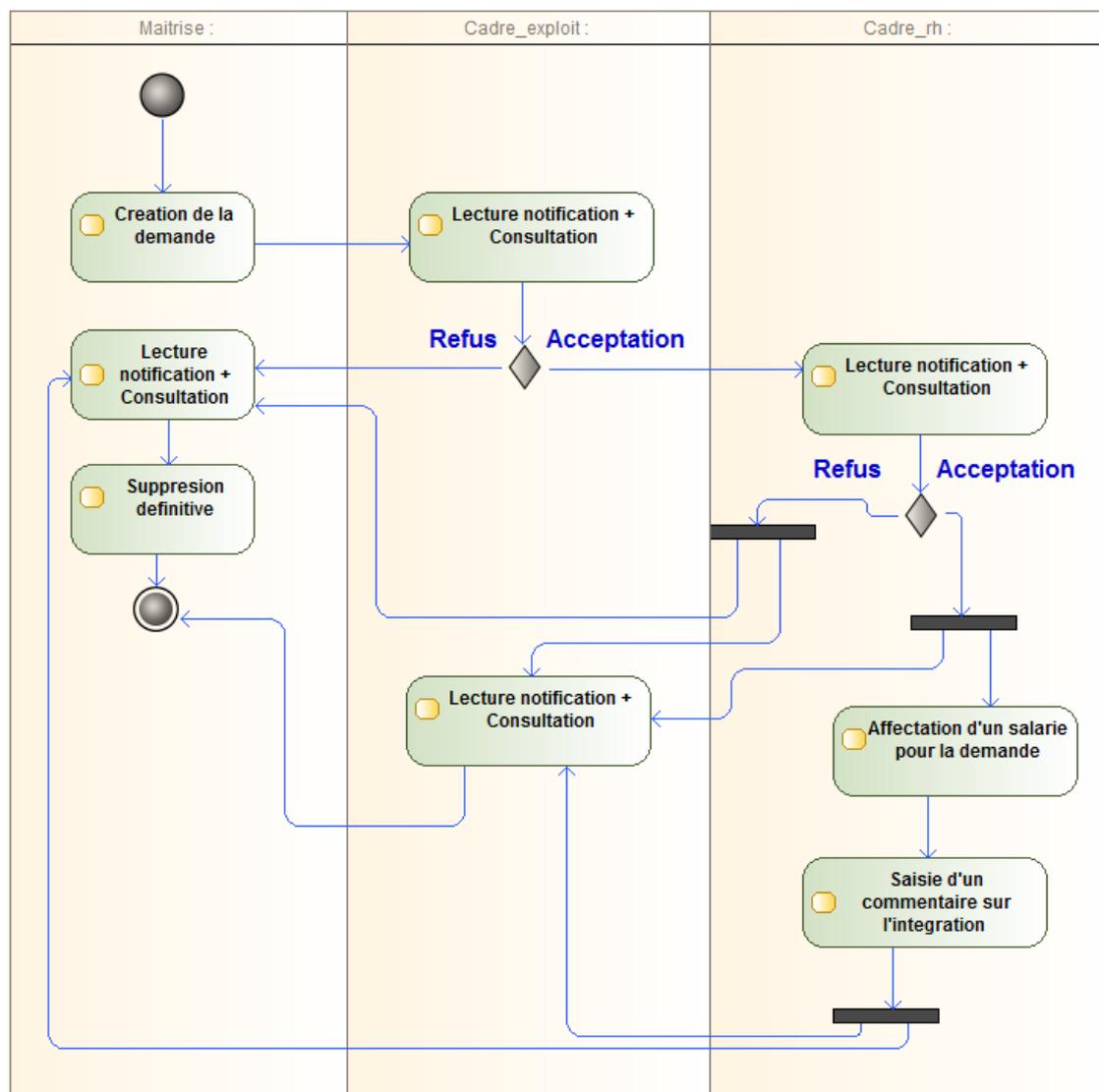


Figure 12: Activity diagram - Réaliser demande interimaire

Le système permet de gérer intégralement le processus de demande d'un intérimaire. Le maitrise crée la dite demande. Ici, le système doit intervenir pour intégrer un processus de validation à deux niveaux.

Tout d'abord, le cadre exploitation traite la requête. En cas de refus, celle-ci sera invalidée et le maitrise pourra alors la supprimer. En cas d'acceptation, la demande est transmise aux RH qui ont également le même traitement à effectuer.

Le maitrise peut alors affecter, dans son planning, cette ressource, même si elle n'est encore que théorique. Si elle est validée par les RH, il est alors possible de conserver ces informations pour les transférer sur le nouveau salarié, ou éventuellement de récupérer un salarié qui serait déjà passé par l'entreprise, et donc présent dans la base de données.

Il est possible de saisir plusieurs informations utiles, comme des commentaires pour chaque validation ou refus, mais également les dates de formation et d'intégration.

Un service de notification interne à l'application est sollicité pour informer tous les acteurs des différents changements d'état d'une demande.

Suivi du personnel :

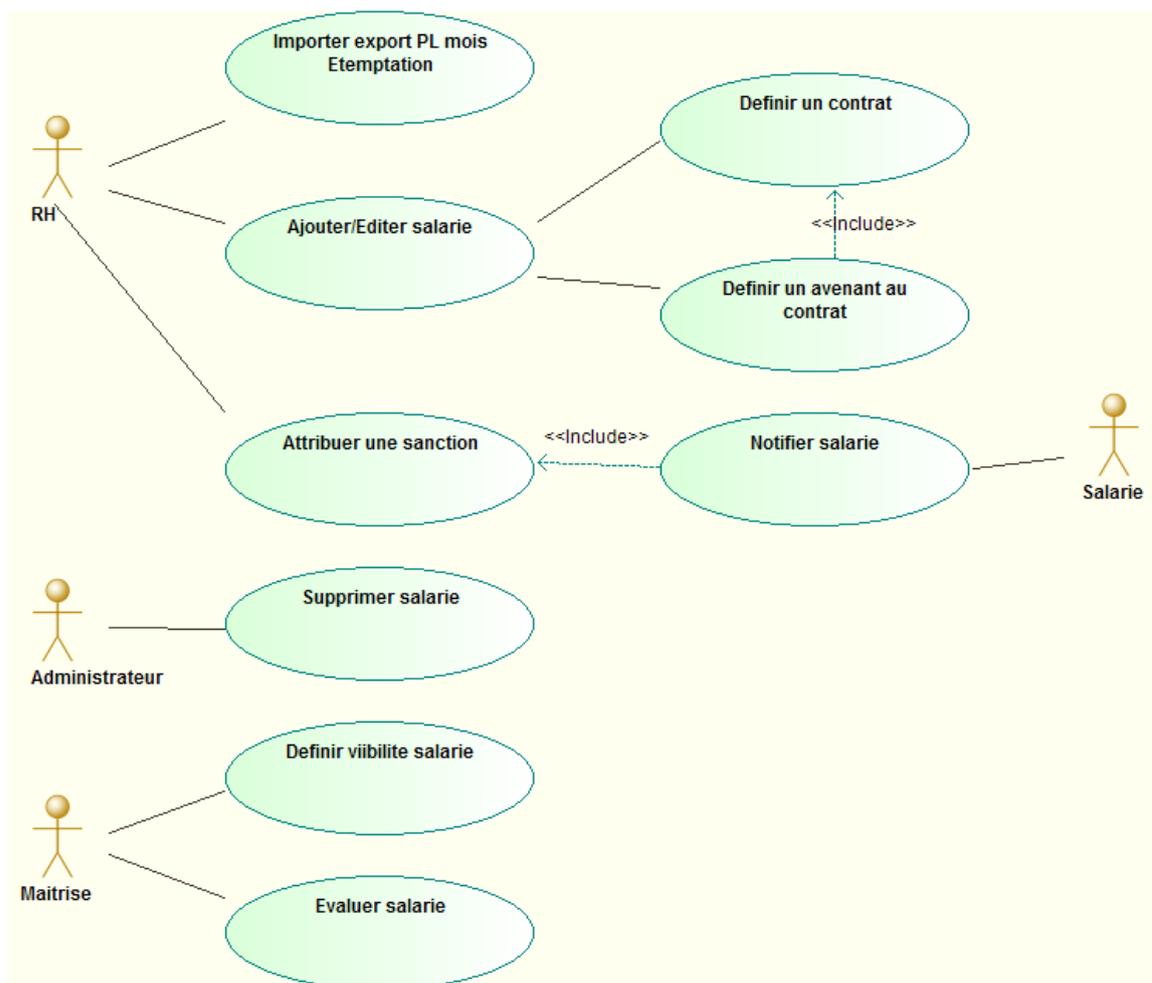


Figure 13: Use case - Gérer salariés

Le digramme de cas d'utilisation ci-dessus présente les fonctions offertes pour la gestion des salariés. Il est ainsi possible de réaliser un import depuis la base d'Etemption. Nous verrons plus précisément comment a été réalisée l'intégration du système au sein du SI et comment il inter-opère avec celui-ci. C'est une fonction qui permet l'intégration en une seule opération de plusieurs centaines de salariés, donc complètement indispensable pour la gestion d'un nombre de salarié aussi important que celui d'Easydis. Il est toutefois possible d'enregistrer manuellement des salariés et bien évidemment de modifier ces informations ultérieurement.

On voit ensuite qu'il est possible de créer des contrats pour ces salariés. Ces contrats n'ont pas un niveau de détail, comme c'est le cas dans Etemption. En effet le but de l'application est d'émettre des alertes sur les fin de contrats, ainsi que de suivre les augmentations reçues par les salariés, informations utiles lors de la réalisation des bilans d'évaluation annuels que nous retrouvons également dans ce diagramme.

Le système doit proposer un service de suivi des différentes sanctions appliquées par les ressources humaines. L'intérêt est l'historisation centralisée de l'information et son partage aux différents responsables concernés de manière automatisée.

Pour finir, on peut voir que seuls les administrateurs peuvent supprimer définitivement un salarié. En effet on souhaite, dans le cycle de vie du système, que les informations sur les différents salariés soient archivées, car elles peuvent présenter un intérêt ultérieur, par exemple : un intérimaire qui passerait dans la société.

Toutefois, il s'avère évident qu'on ne peut laisser visible, par exemple dans la liste des effectifs, les salariés ayant quittés la société. Pour cela la fonction de définition de la visibilité d'un salarié existe. Nous verrons que ce concept est une solution de conception qui a été appliquée à plusieurs endroits du système et que nous détaillerons plus avant.

Notifications :

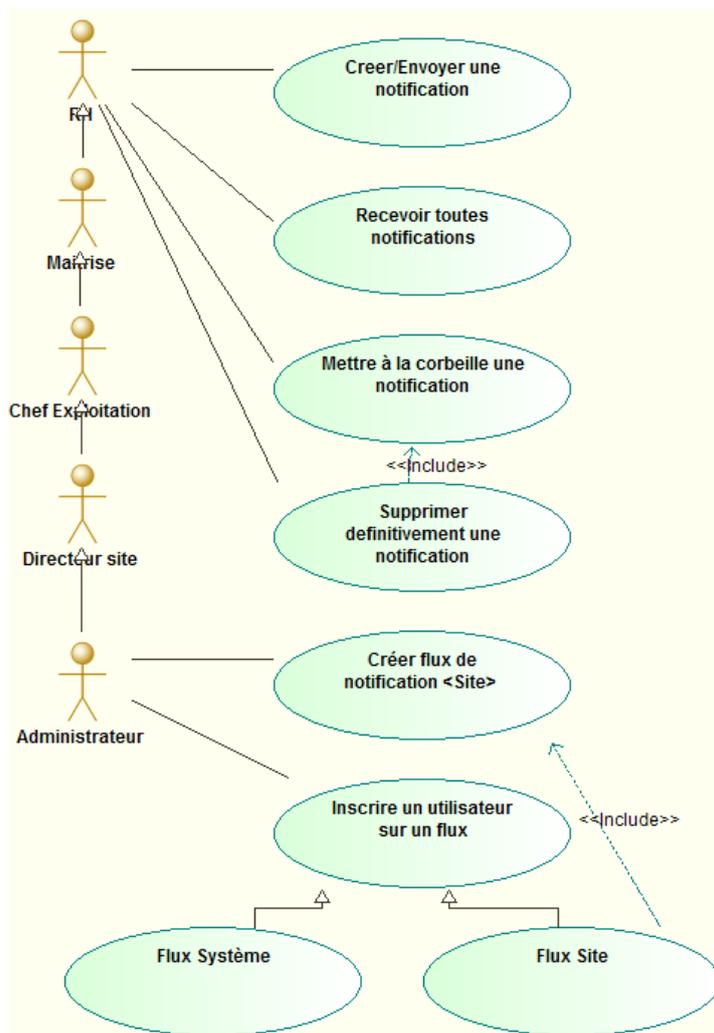


Figure 14: Use case - Gérer notifications

Pour terminer avec la présentation des fonctionnalités, le diagramme ci-dessus présente les services offerts par le sous-système de notification. Ici nous voyons que l'ensemble des acteurs de notre système a la possibilité de créer, envoyer et supprimer des notifications.

Ces notifications peuvent être envoyées à un ensemble de listes de diffusion du site auquel le dit utilisateur appartient. Cette restriction est due au fait que, pour l'instant, le système n'a pas vocation à être un outil d'échange, notamment inter site, mais plutôt à permettre de réaliser des notes d'information globales à un site de manière ponctuelle. Ici l'administrateur a également un rôle important, puisque c'est lui qui est chargé de gérer les flux des différents sites et d'y abonner les utilisateurs.

On distingue ensuite deux types de flux :

- ✓ Les flux site, qui permettent de personnaliser le sous-système de notification sur chaque site.
- ✓ Les flux système, qui permettent de notifier aux utilisateurs des évènements générés par l'application.

Par exemple : la création d'une demande intérimaire générera automatiquement une notification à destination du chef d'exploitation, pour demander une validation, et ainsi de suite dans la chaîne du processus de demande intérimaire que nous avons déjà pu voir.

## III.1.4 Architecture logicielle

### III.1.4.1 Architecture en couches

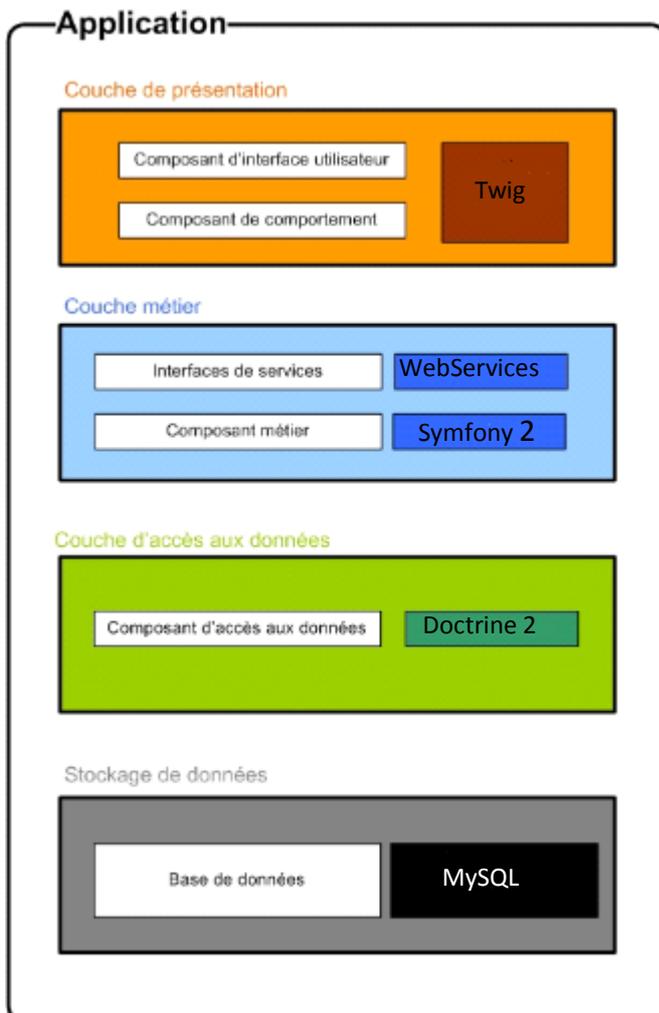


Figure 15 : Schéma architecture en couche

On retrouve dans ce schéma les couches composant le MVC, avec :

- ✓ La couche de présentation qui s'appuiera sur plusieurs technologies :
  - Twig : que nous avons évoqué
  - JavaScript : pour l'interaction avec le client
  - AJAX : pour des traitements dynamiques complexes
  - HTML5 / CC3 : pour la mise en page et la mise en forme
  
- ✓ La couche métier, qui intègre les traitements propres à l'application qui seront donc en PHP et bénéficieront des fonctionnalités de Symfony.

- ✓ La couche d'accès aux données, gérée par Doctrine 2 avec l'ensemble des requêtes réalisé en DQL.
- ✓ La couche de stockage des données, effectuée sur une base de données MySQL, qui sera, à terme du déploiement global, migrée sur une base Oracle.

### III.1.4.2 Architecture Framework Symfony 2 :

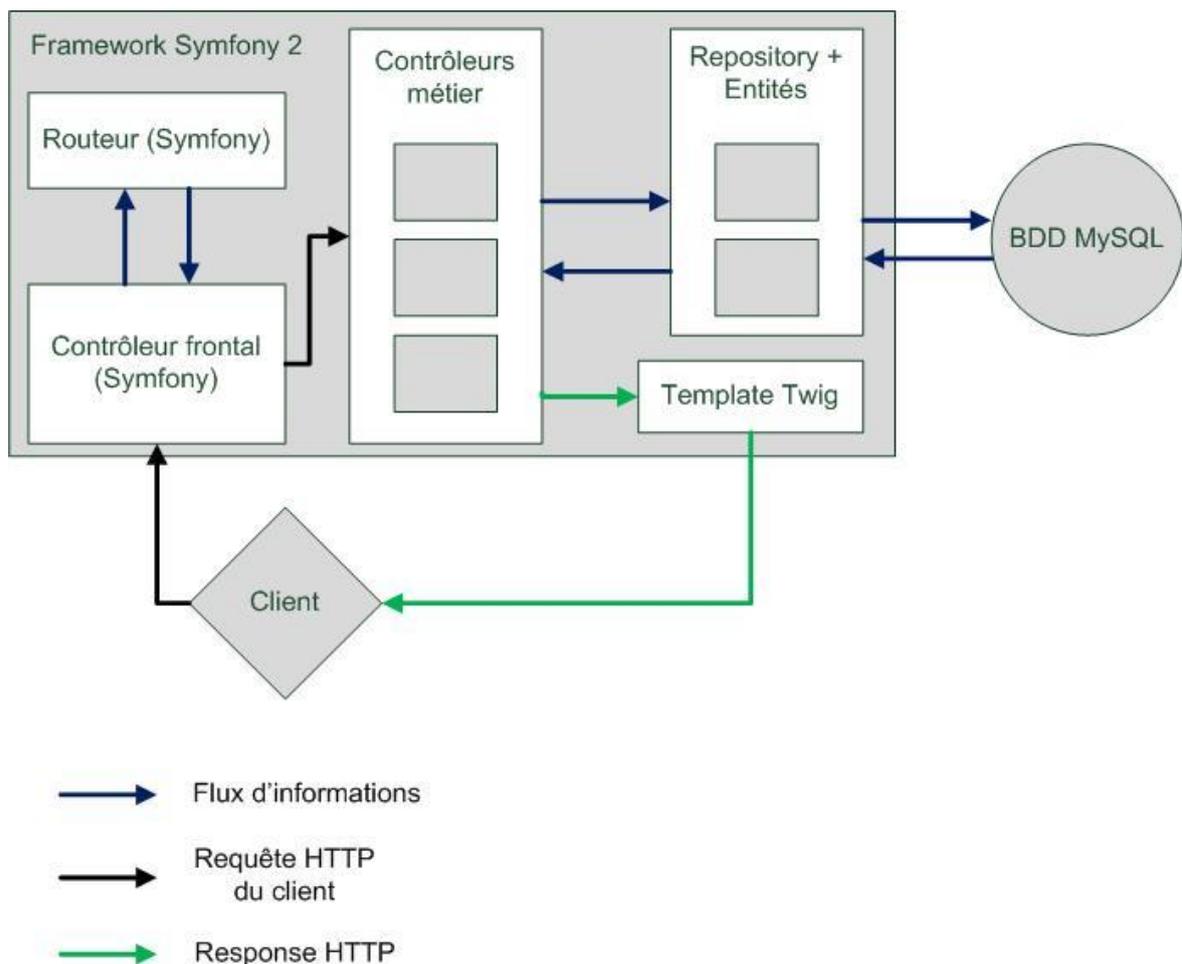


Figure 16 : Schéma architecture framework Symfony

Le schéma ci-dessus montre comment cette architecture se présente une fois appliquée avec le Framework Symfony. Il intègre les principales couches logicielles et les différents flux internes à l'application. Ce n'est pas un schéma d'architecture détaillé. Il vise à présenter globalement les principaux composants.

Le scénario présenté par le schéma est celui d'un client qui réalise une requête grâce à son navigateur. Le contrôleur principal correspond dans le schéma au contrôleur frontal, offert par Symfony. Il redirige, grâce au routeur configuré par un fichier de routing, les requêtes sur les différents contrôleurs métier.

Il est également important de noter que c'est grâce à ce contrôleur que le Framework permet de gérer le « rewriting d'url » qui n'est pas pris en charge nativement par PHP.

En réalité, le routeur possède deux niveaux de routing, disons deux tables de routage. La première sert à rediriger sur les bundle et la seconde sur les contrôleurs métier. Ici c'est bien la seconde action que nous présentons. La première est en réalité induite.

Une fois cette opération effectuée, une « Action » appartenant à un contrôleur métier est appelée pour réaliser le traitement nécessaire à la réalisation de l'action demandée par l'utilisateur (génération d'une vue, traitement métier ...).

Nous sommes ici dans la couche métier de l'application, en concordance avec le MVC. Ce contrôleur va pouvoir utiliser la couche d'accès aux données, c'est-à-dire nos DAO, qui prennent ici le nom de « Repository ». Ces repositories utilisent donc du DQL et Doctrine pour persister et récupérer des entités depuis la base de données. Si on ne nomme pas nos repositories DAO, c'est bien parce qu'en réalité c'est Doctrine qui les intègre. Les Repositories n'héritent que de la tâche de récupération des données. Le CRUD est complètement intégré à l'ORM, grâce aux annotations faites dans les entités.

Pour finir, la couche de présentation est représentée par les Templates Twig. Une fois les traitements effectués par la ou les actions, d'un ou de plusieurs contrôleurs, ces informations sont passées sous la forme d'une « Response http » au client, qui va alors pouvoir utiliser le Template adapté pour générer une page en HTML.

### III.1.4.3 Couche métier

L'organisation logicielle principale se situe au niveau de la couche métier, qui intègre les contrôleurs et actions métier. La réalisation de ce découpage est une étape importante de la conception, puisque cela aura un impact direct sur la maintenabilité de l'application.

Le découpage des contrôleurs a été effectué comme suit :

- Routing
- Dashboard
- DashboardRH
- RH
- Formation
- Planning
- Security
- Application

Nous verrons que la méthodologie appliquée au projet fait que ce découpage et de nombreux éléments du projet peuvent être amenés à évoluer. Toutefois, cette organisation a permis la réalisation du projet jusqu'à aujourd'hui.

Ainsi le contrôleur de routing, qui n'a rien à voir avec le routeur du Framework, permet de centraliser des actions génériques de choix.

L'objectif ici est de factoriser tout le code concernant la redirection sur des contrôleurs nécessitant des paramètres. Par exemple, on trouve dans ce contrôleur une fonction qui propose à un utilisateur de sélectionner un service, une exploitation ... L'utilisation est simple : si un utilisateur veut visualiser le tableau de bord d'un groupe de services, la requête va d'abord passer par ce contrôleur, qui la redirigera avec ses paramètres à l'action chargée de réaliser les traitements nécessaires à la génération de ce tableau de bord.

On trouve ensuite le contrôleur Dashboard, qui, comme son nom le suggère, a la charge de toutes les actions effectuant des traitements métier concernant la génération de tableaux de bord pour l'exploitation, de la même manière que DashboardRH le fait pour les ressources humaines.

Quant au contrôleur RH, celui-ci propose des actions permettant la partie gestion des ressources humaines, comme les contrats, les sanctions, les salariés, les demandes d'intérimaires ...

Les formations comme les plannings disposent d'un contrôleur propre pour assurer leurs traitements.

Pour finir, il subsiste deux contrôleurs, Security et Application. Le premier gère l'authentification des utilisateurs. Le second, un peu plus général, gère principalement les notifications.

### **III.1.5 Modèle du domaine**

Un modèle global a été conçu dès le début du projet afin de maximiser sa cohérence. Le modèle a été amené à évoluer durant le processus de développement, mais seulement de manière sporadique.

Il se compose de 42 classes. Il n'est donc pas possible de le présenter clairement dans sa globalité. C'est pourquoi j'ai choisi de le découper pour vous le présenter partie par partie.

Toutefois vous pouvez trouver le modèle dans son intégralité en annexe.

## Flexibilité et tableaux de bord :

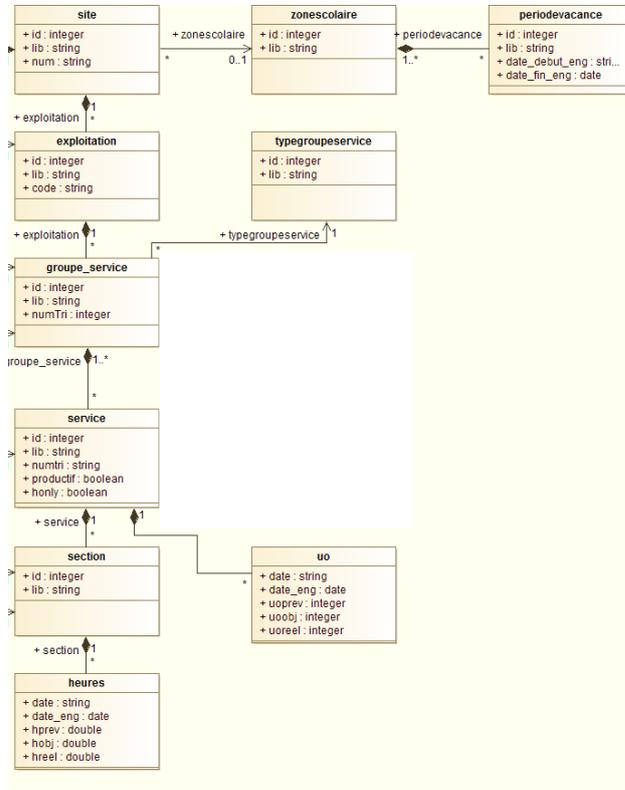


Figure 17 : Diagramme de classe - Généricité

L'extrait du modèle ci-dessus permet de répondre à une problématique majeure du système : sa flexibilité. Nous l'avons vu, le système doit pouvoir être déployé sur différentes structures d'Easydis. La première action a été de définir ce tronc commun qui permet de personnaliser l'application sur chaque exploitation.

Pour réaliser cela, le point de départ a été le plus bas niveau de gestion : les sections. Toutes les exploitations ont en commun de travailler avec des sections, car des systèmes comme Etemptation fonctionnent sur cette notion.

Nous l'avons vu, le système doit permettre de visualiser et gérer plus globalement à différents niveau toutes ses informations.

La solution a été de rajouter des niveaux, en premier lieu celui des services, qui peut contenir une ou plusieurs sections.

Ensuite, pour coller à la réalité métier, une classe groupe de services a été créée. Elle représente l'élément de gestion le plus utilisé pour visualiser et gérer, car c'est concrètement l'ensemble des services affecté à un maitrise. Les maitrises travaillent et pensent généralement en groupe de services.

Vient ensuite le chef d'exploitation et le responsable du site, qui respectivement pensent Exploitation et Site, d'où les classes de niveau supérieur.

Actuellement, le site est le plus haut niveau de gestion de l'application. Même s'il est prévu que des directeurs régionaux puissent utiliser l'outil, il n'est pas paru nécessaire de posséder un niveau régional. Cela s'explique par une raison simple : même si l'activité est identique sur chaque site, il n'est pas forcément cohérent de les comparer directement les uns par rapport aux autres, tant leurs organisations / contraintes peuvent différer.

Il reste quand même possible que le déploiement global amène le système à évoluer pour intégrer des niveaux supplémentaires, et permettre de créer des indicateurs plus centraux : cela aurait pour conséquence de faire évoluer ce modèle, mais ne devrait pas affecter le travail déjà réalisé.

Vous l'aurez compris, ces classes sont le cœur même du système, notamment pour tout ce qui concerne les tableaux de bord.

On trouve ainsi plusieurs classes comme UO (Unité d'œuvre) et Heures qui permettent de stocker des informations concernant l'activité d'une exploitation. Stockées au plus bas niveau, elles sont ensuite regroupées au niveau désiré pour former ces tableaux. D'autres informations permettent de générer des tableaux de bords. C'est le cas par exemple des plannings qui fournissent un grand nombre d'informations, mais que nous verrons séparément.

Ensuite on trouve également la notion de chemin. Cette notion est propre aux services assurant une activité de préparation. Ces derniers fonctionnent sur un système de prime individuelle qui est calculée par rapport à une productivité d'un préparateur sur un chemin.

Grace à cela, nous pouvons construire l'activité en terme de productivité pour chaque préparateur. Ces informations sont simplement regroupées et mise en forme par le système. Elles sont préalablement intégrées depuis un système externe : Easydev. Ce dernier permet justement le calcul de ces primes.

Une majorité des tableaux de bords de l'exploitation repose sur cette partie du modèle.

Nous n'avons pas fait figurer sur chaque sous-modèle la partie centrale qui est la structure composée des classes Section, Service, Groupe de services, Exploitation et Site. Toutefois celle-ci intervient nécessairement dans la spécialisation de chaque sous système.

## Ressources Humaines :

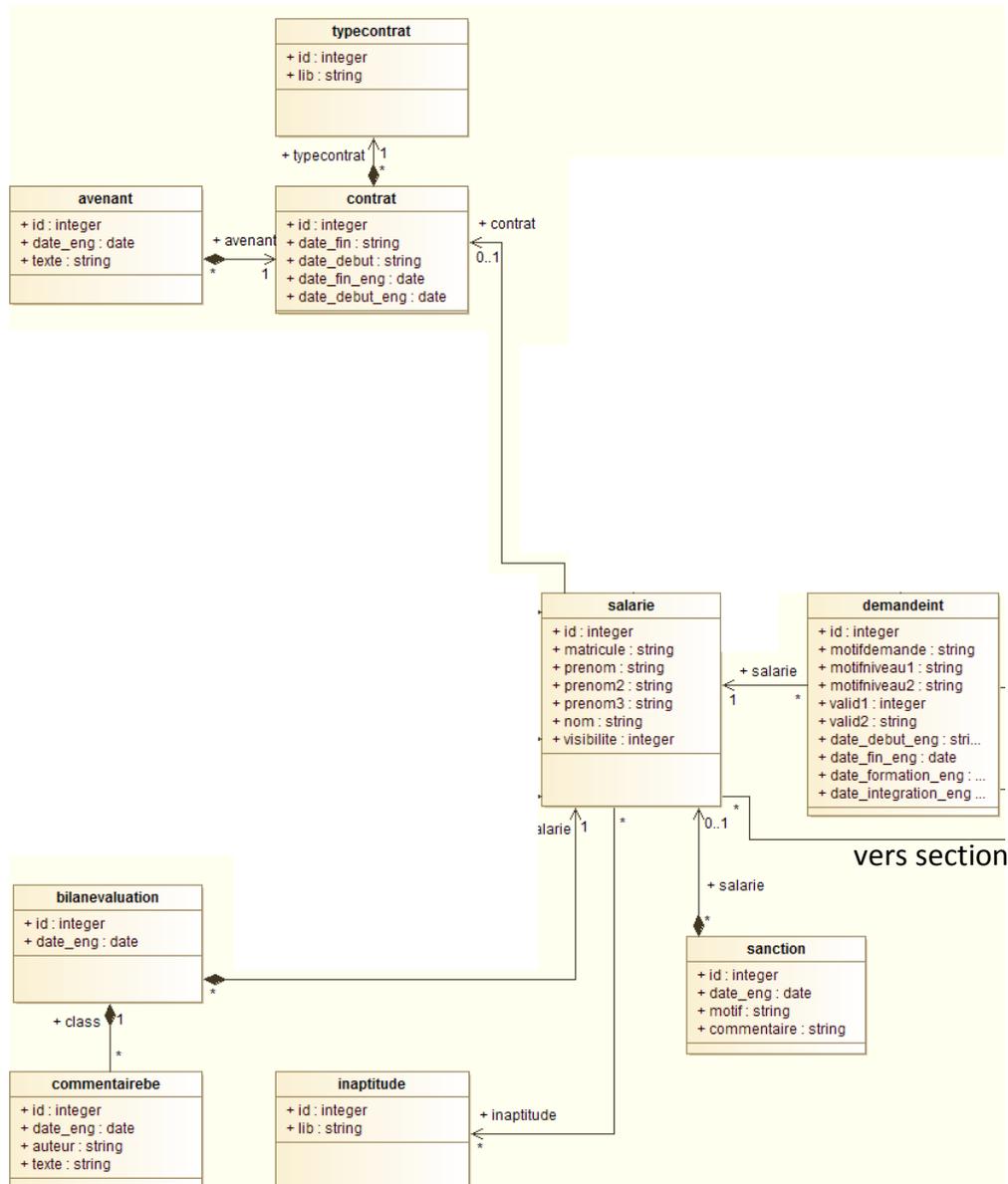


Figure 18: Diagramme de classe - Services ressources humaines

On remarque que dans cette partie la classe salarié est très centrale. De nombreuses relations sont ici masquées pour mettre en avant la gestion des fonctions RH par le modèle. Assez simplement, on retrouve les éléments des principales fonctions assurées par les ressources humaines.

La classe « demandeur » permet de créer des demandes d'intérimaires. On retrouve à l'intérieur des attributs permettant de réaliser les différents niveaux de validation précédemment vus dans le diagramme d'activité.

On trouve ensuite les bilans d'évaluation, inaptitudes et sanctions. Si la conception en l'état permet de gérer les salariés et leur type de contrat en cours, elle ne permet pas encore de définir des avenants aux contrats, ou encore de consulter leur historique. Les contrats sont un élément plus complexe à gérer et ne sont pas, pour le moment, pleinement mis en place. Ce point pourra certainement être amélioré ultérieurement.

Finalement, on remarque que les salariés possèdent une section de rattachement, ici notée « vers section ». C'est cette liaison qui permet au système de gérer les salariés par site.

**Formation :**

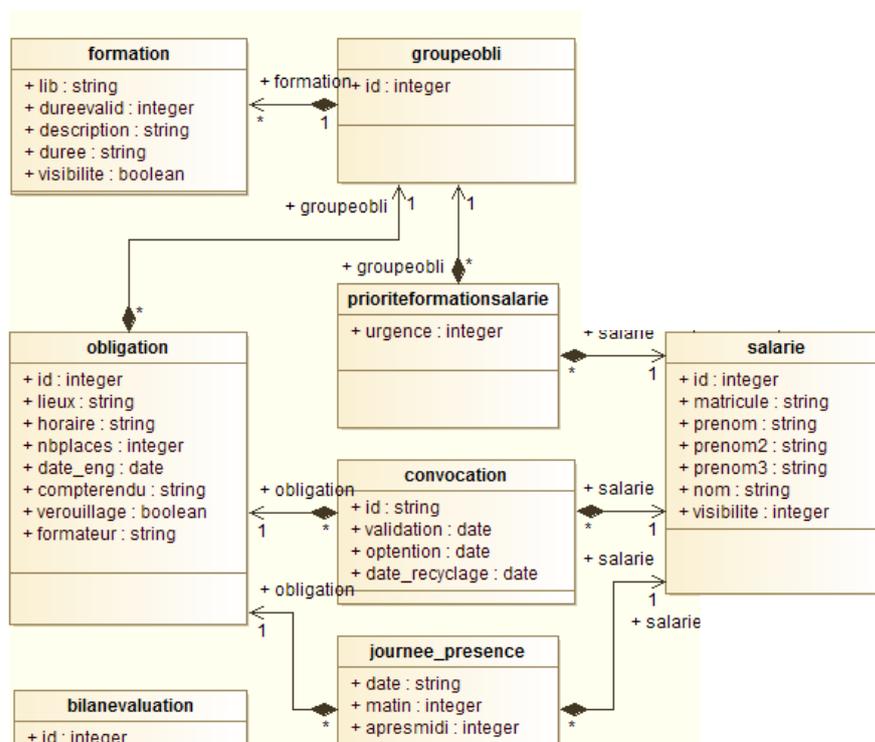


Figure 19: Diagramme de classe - Services Formation

Cet extrait met en avant la gestion des formations. Nous avons vu, dans le diagramme d'activité concernant la réalisation d'une formation, que celles-ci sont organisées en groupe de plusieurs sessions, puis proposées aux maitrises.

Ici, la classe formation n'est pas identifiable sur un site donné. En effet, les formations suivent un référentiel fourni par le siège, donc tous les sites l'utilisent. Ces formations sont, nous avons pu le voir, gérées par l'administrateur. Par contre, le responsable des formations de chaque site peut créer une session d'une formation qui correspond, dans le modèle, à une instance de la classe « obligation ».

La classe « groupeobli » permet de créer des groupes de sessions. Mais, pour une question d'accessibilité, cette notion est masquée. L'utilisateur ne crée pas un groupe de sessions mais plusieurs sessions qu'il peut ensuite lier les unes aux autres.

L'attribution des salariés sur une session de formation se fait automatiquement en fonction de la date de repassage de chaque salarié pour la dite formation. Il est toutefois possible d'ajouter manuellement des salariés à former. Elle est matérialisée par la classe « prioritéformationsalarie » car elle intègre une notion de niveau de priorité.

Pour finir, la classe « convocation » permet d'inscrire un salarié à une session donnée. Cette classe doit être validée pour être transformée en convocation. Une fois convoqué, un salarié se doit d'être présent à l'ensemble des journées de formation. Pour gérer cette présence, deux émargements par jours sont réalisés (matin et après-midi). Il est ensuite possible, grâce à la classe « journee\_presence », de conserver un historique de cette information.

## Plannings :

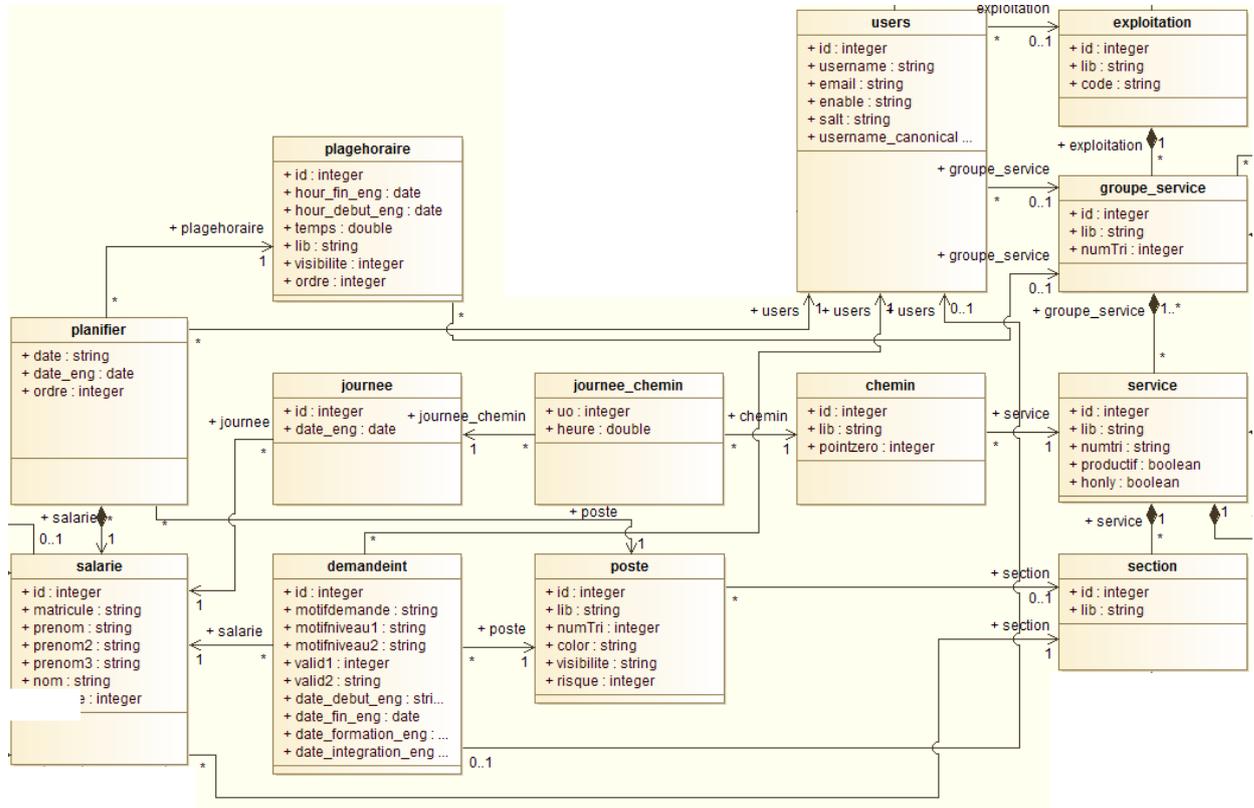


Figure 20: Diagramme de classe - Planification

Cet extrait présente l'aspect planification du système.

La conception des plannings fut compliquée. Les contraintes initialement imposées ont largement évolué au cours du développement, ce qui a eu pour conséquence de devoir reprendre complètement la conception de ces fonctions. Finalement, ce modèle permet de gérer l'ensemble des exigences imposées jusqu'à aujourd'hui.

La classe « planifier » permet l'association d'un salarié, d'un poste, d'une plage horaire et ce pour une date donnée. Les clés primaires sont ici fortes de sens. On remarque que le salarié, la date et la plage horaire forment la clé primaire composée.

En effet le système doit gérer le multiposte. En ce sens, il est possible de planifier plusieurs fois un salarié pour une date donnée. Par contre cela n'a pas de sens d'attribuer deux postes différents pour une même plage horaire. De plus, des contraintes comme l'amplitude horaire existent et doivent être vérifiées. Par exemple : une personne ne peut enchaîner deux plages horaires complètes sans une coupure d'un minimum de 12h.

On note également la présence dans la classe « planifier » d'une instance de la classe « user » qui permet d'assurer une traçabilité de chaque planification. Une fois saisie, il est possible d'identifier les prêts de personnels grâce à l'association d'un poste à une section, et d'un salarié à une section dite de rattachement. C'est une notion importante qui permet de détecter des anomalies. Ces informations sont également utiles pour les saisies réalisées par les ressources humaines sur Etemptation.

Les plages horaires et les postes sont définis par les maitrises qui peuvent ainsi personnaliser leur planning. L'affectation sur des sections se voit masquée par l'affectation de poste, ce qui permet de réaliser tous les tableaux de bord nécessaires, tout en gardant lisible les plannings.

### Notifications et utilisateurs :

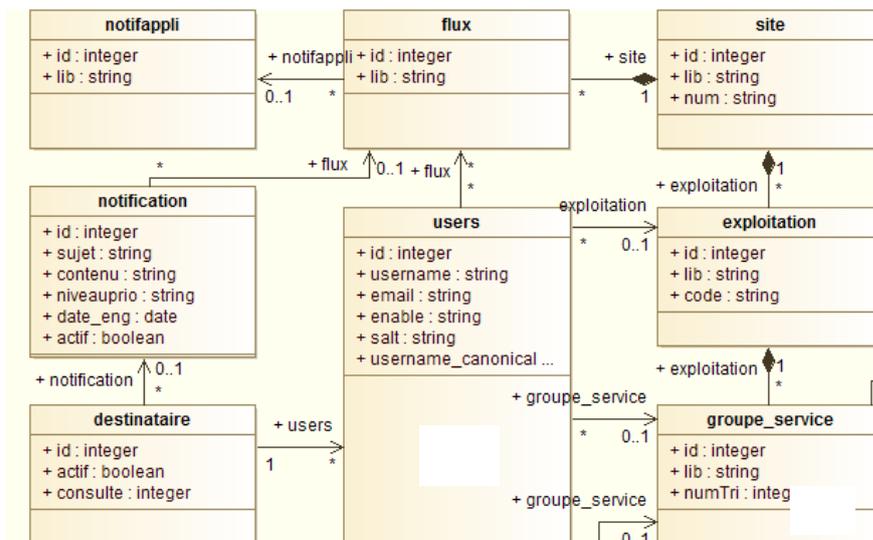


Figure 21 : Diagramme de classe - Notification

Pour terminer la présentation du modèle du domaine, l'extrait ci-dessus présente la gestion des notifications et des utilisateurs.

Nous avons déjà vu la classe « users » qui permet de gérer des comptes d'utilisateur du système. Finalement, la majorité des utilisateurs n'est concernée que par une partie restreinte au périmètre de leur exploitation. C'est pour cela que les utilisateurs peuvent être rattachés à une exploitation et un groupe de services.

Ces associations servent de choix par défaut pour la vision de tous les éléments du système. Ainsi, pour les notifications par exemple, les listes de flux affichées sont récupérées grâce à la liaison d'un utilisateur à une exploitation.

En ce qui concerne les notifications, premièrement, les administrateurs créent pour chaque site différent flux qui peuvent être interprétés comme des listes de diffusion, puisque ces derniers viennent ensuite abonner les utilisateurs sur ces listes. Conceptuellement, cela permet au sous-système de notification d'être générique, et ce sont ces attributions qui le spécialisent.

Les utilisateurs peuvent ensuite émettre des notifications, non pas à destination d'un utilisateur unique, car là n'est pas le but du système, mais à destination d'un flux, bien qu'un flux ne peut en réalité n'avoir qu'un seul abonné. Par exemple : un flux « CadreRH » sera à destination du DRH d'un site, et donc seul le DRH sera abonné par l'administrateur à ce flux.

Les notifications représentent une instance de la classe « notification », et une ou plusieurs de la classe « destinataire ». Cette notion de séparation a été nécessaire pour permettre aux utilisateurs de gérer notamment la suppression des notifications de manière indépendante. Un problème se posait alors quand à la suppression de cette notification, car même l'émetteur possède un élément de la classe « destinataire ». Une règle métier a donc été mis en place. Elle vérifie, lors de la suppression d'un élément de la classe destinataire, la présence d'autres éléments ayant pour source la même

notification. S'il est le seul, alors il supprime également la source, c'est-à-dire la notification.

Les notifications système sont représentées par la classe « notifappli ». En réalité, cette classe est fixée par les développeurs et n'est jamais amenée à évoluer, à moins que des ajouts de fonctionnalités soient faits sur le système. Cela se présente sous la forme d'une liste de services proposés par l'application. Ces services sont appelés par le système dès lors que des évènements prédéfinis se produisent.

Par exemple : « valid\_demandeint\_cadreRH » qui a pour description « Emet une notification de demande de validation à destination du cadre RH responsable de la validation d'une demande intérimaire ». Ce service sera invoqué dès lors qu'une demande intérimaire aura été validée par le cadre exploitation.

Ensuite, les administrateurs rattachent ces services à des flux auxquels sont abonnés des utilisateurs. L'évènement associé au service se produit, l'application récupère la liste des destinataires qui se voient alors notifiés de l'évènement.

Il faut noter que toutes les notifications émises par le système le sont par un compte utilisateur « system », ce qui permet de facilement trier ces éléments.

### III.1.6 Intégration des données

Nous avons vu comment était construit le modèle du domaine et la quantité de données que celui-ci intègre. Nous allons maintenant voir comment il est alimenté.

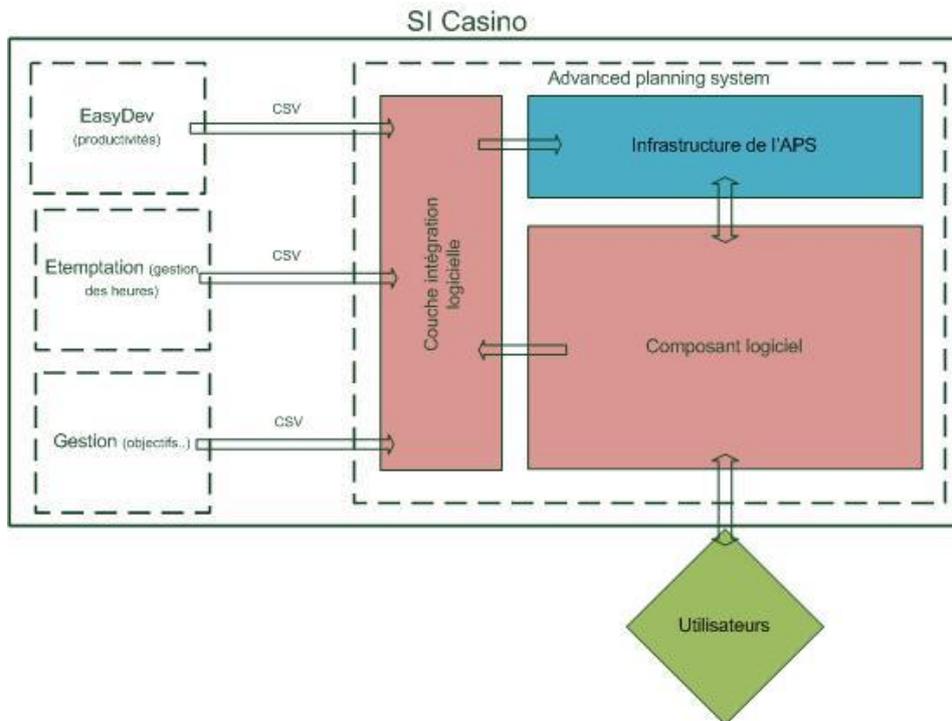


Figure 22: Schéma intégration système

Le schéma ci-dessus présente l'intégration des données du SI de Casino par le système. Nous avons déjà pu évoquer les outils que sont :

- Easydev qui gère le système de prime pour la productivité,
- Etemption, l'ERP de nos ressources humaines
- La « gestion » qui, plus largement, au niveau du siège, travaille sur SAP, mais, sur les sites, remonte les informations sur des tableurs Excel.

Nous verrons un peu plus loin un schéma plus détaillé pour ce qui concerne l'intégration de la partie Gestion.

On retrouve une décomposition interne du système en trois éléments.

Le composant logiciel représente l'ensemble des fonctions offertes par le système. Elles peuvent faire des échanges avec l'infrastructure, qui ici représente la couche d'accès ainsi que la couche de stockage des données.

Finalement, la couche d'intégration logicielle correspond en fait au contrôleur « Application » que nous avons pu voir, mais pas seulement puisqu'on y trouve aussi des documents Excel. Ce contrôleur dispose de l'ensemble des outils pour intégrer plusieurs types de fichiers en CSV.

Un certain nombre de problématiques s'est posé très tôt dans le projet concernant l'intégration des données. Il aurait fallu avoir accès aux systèmes existants, ce qui s'est avéré impossible. Seule option possible : l'intégration d'exportations réalisées à partir de ces systèmes externes. Pour des systèmes comme Etemptation, cela s'est avéré tout à fait réalisable, car celui-ci est commun à l'ensemble des sites et propose tout un ensemble d'exports formalisés. Il est donc précisé, dans le manuel d'utilisation du système, comment exporter depuis Etemptation, puis comment l'importer pour alimenter le modèle.

Toutefois cela s'est avéré beaucoup plus difficile de déterminer une source commune de données, pour tout ce qui concerne la gestion et même d'EasyDev. Pour la gestion, tout est géré sous Excel, même si c'est ensuite repris par le siège sur SAP. Hors les informations de SAP sont confidentielles, celles-ci ne contenant pas uniquement des résultats d'activité mais l'ensemble des finances de la société.

Il fallait donc se contenter des documents Excel du site Eurocentre. Sur celui-ci, il existe un fichier qui lui est propre, qui permet de gérer les budgets, et les résultats de performance. Malheureusement, ces fichiers ne sont en l'état aucunement exploitable pour réaliser un import en CSV.

La solution trouvée fut de créer des classeurs intermédiaires pour l'intégration des données. Ceux-ci ont pour rôle de venir centraliser et rendre exploitables les informations de la gestion.

L'idéal, pour un déploiement global, serait bien évidemment que l'ensemble des sites utilise les mêmes outils, mais cela reste compliqué avec de l'Excel et du VBA.

Ci-dessous une représentation de cette intégration :

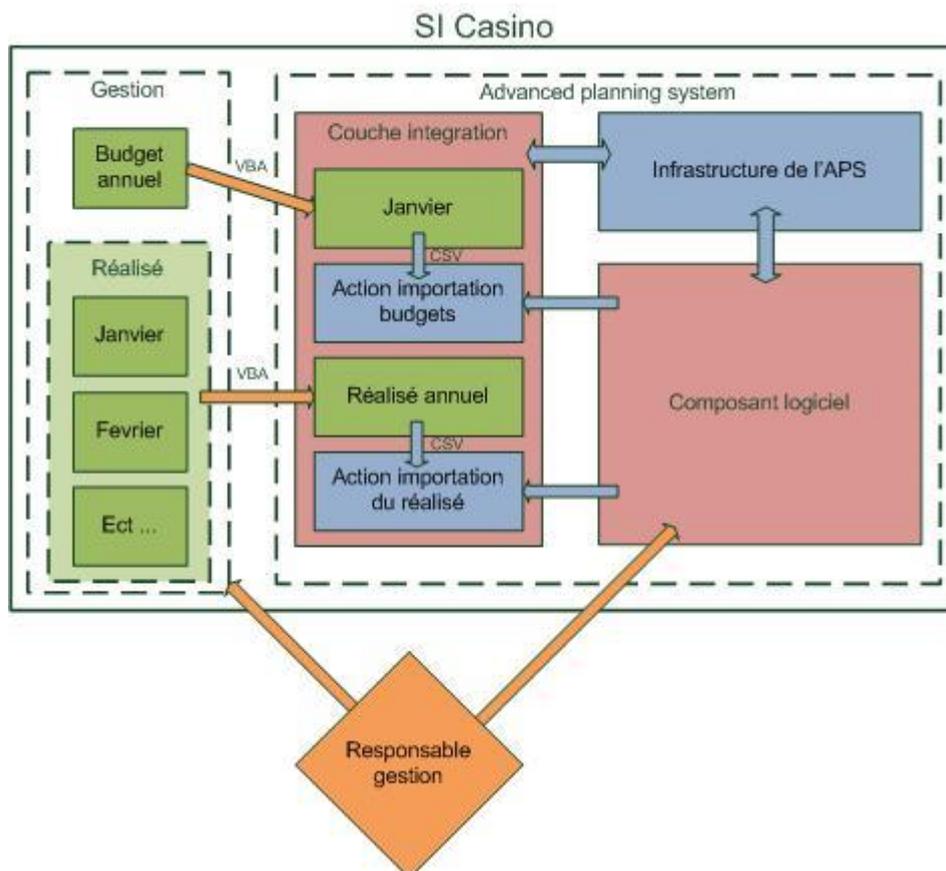


Figure 23: Schéma couche intégration détaillée

Dans ce schéma, les rectangles verts représentent des classeurs Excel, ceux en bleu sont des actions. On retrouve l'architecture détaillée plus haut.

On voit ici clairement que l'intégration des données de la gestion s'avère plus complexe. L'acteur mentionné ici est le responsable gestion, mais tous les salariés du service gestion peuvent effectuer ces tâches.

Pour commencer, il faut distinguer les actions concernant les budgets (ou objectifs) de ceux du réalisé. En effet, la mise à jour du réalisé est faite quotidiennement pour la veille, alors que l'objectif est défini en début d'année pour l'ensemble de celle-ci : on ne fait donc qu'un seul import par an.

Le schéma montre également les différents classeurs. Par exemple, on voit que l'ensemble des budgets son compris sur un classeur, alors que le réalisé est divisé avec un classeur par mois. Il existe donc deux classeurs qui servent à récupérer l'ensemble de ces informations et qui les met en forme pour les rendre exploitable.

Ces actions de mise en forme et de centralisation ne sont pas réalisées automatiquement : c'est l'utilisateur qui doit les exécuter. Une fois ces classeurs mis à jour, il faut utiliser les services d'importation qui eux récupèrent des fichiers CSV, fichiers intégrés ensuite dans la base de données.

L'intégration de transformation Excel en VBA au sein du système était un choix de conception. Il devait permettre de palier aux différences entre les sources de données. En agissant comme un connecteur, qui reste en réalité spécifique à chaque source, il sera nécessaire soit d'uniformiser les classeurs sources de tous les sites, soit de développer des connecteurs compatibles pour les intégrer au système. Il faut savoir que l'avantage est que ces connecteurs, réalisés en VBA, sont rapides à développer. Par contre, comme toujours en VBA, sont trop dépendants de la structure du document.

Si le déploiement global est réalisé, il sera sûrement nécessaire de « reconcevoir » cette intégration, en donnant, au système, des accès plus haut niveau à ces données.

### **III.1.7 Gestion des accès et sécurité**

L'objectif, ici, n'est pas de détailler la sécurité au niveau du réseau informatique du groupe , ne disposant pas de ces informations.

Toutefois, le serveur d'application du système n'est pas accessible de l'extérieur, excepté si l'on possède une VPN configurée. Les accès sont donc restreints en intranet.

Pour protéger le système, les services inclus dans le Framework Symfony concernant les accès et la Security ont été utilisés, cela en combinaison avec un bundle open source, du nom de « FOSUserBundle », qui lui offre des services pour la gestion de comptes utilisateurs.

Pour présenter cela, nous allons analyser un schéma issu directement de la documentation de Symfony.

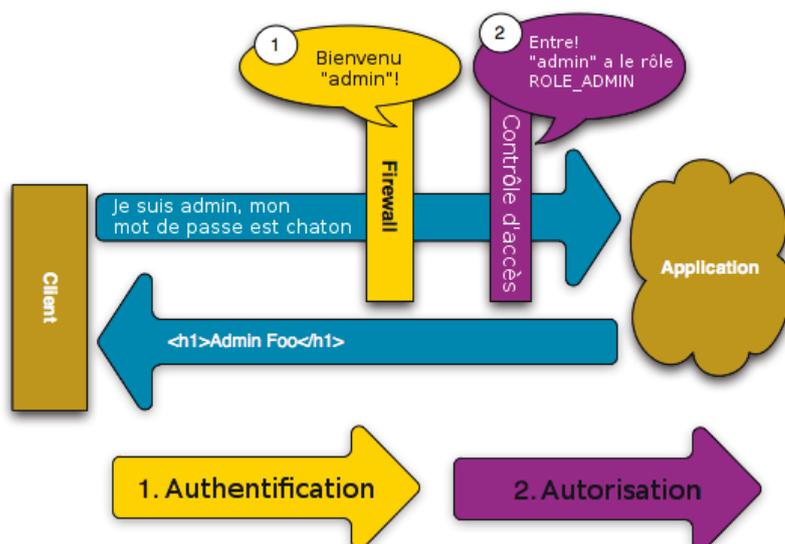


Figure 24: Scénario de gestion des accès par Symfony 2 (Source: Documentation officielle Symfony 2)

Le schéma ci-dessus est intéressant car il met en évidence les deux concepts de l'architecture de gestion des accès de Symfony.

Le premier est l'authentification qui va permettre, via un formulaire, de vous identifier. L'application peut ou pas accepter les connexions anonymes. Ce n'est toutefois pas le cas de l'APS qui n'est accessible qu'à des personnes possédant un compte utilisateur.

Dans le système, nous passerons donc toujours par le firewall pour identifier les utilisateurs. Une fois authentifié, on voit que le contrôleur des accès vérifie les droits d'accès aux ressources : c'est la partie autorisation. Pour définir les droits, il faut en fait créer des rôles. On peut voir dans le schéma un ROLE\_ADMIN. On peut ensuite faire hériter à un rôle d'autres rôles. Par exemple, dans le projet, un cadre exploitation peut hériter du rôle de maitrise. Il suffit ensuite de spécifier les rôles autorisés pour l'accès à chaque ressource ou, autre possibilité, créer une zone restreinte grâce au système de routing de Symfony. Par exemple, dans notre système, toutes les routes débutant par « /aps/administration/ » n'acceptent que des utilisateurs disposant d'un rôle d'administrateur.

Pour le moment, le système ne possède que deux restrictions d'accès. La première, nous venons de la proposer comme exemple, concerne toute la partie administration du système. La seconde concerne un ensemble de fonctions propres aux RH. Elles sont regroupées sous des routes commençant par « /aps/RH/secured/ ». C'est ici que seront accessibles les fiches de description complètes des salariés, avec des informations confidentielles, comme les salaires par exemple.

Le système de gestion des accès de Symfony est très complet. Toutefois, la gestion des utilisateurs reste à la charge du développeur. Pour cela, nous avons utilisé un bundle qui offre tous ces services. Il permet notamment la création d'utilisateur, la récupération des mots de passe et la gestion de nombreuses informations, comme les dernières dates de connexion notamment. Il suffit alors de surcharger les Templates du bundle pour les remplacer par les nôtres.

Nous avons fait le choix d'encoder les mots de passe via une méthode « SHA 512 » pour garantir la sécurité de ceux-ci en base de données.

### **III.1.8 Architecture technique**

Pour l'architecture, nous avons effectué une étude quant au choix du matériel. L'objectif était de pouvoir héberger l'APS mais également Easycom (projet réalisé en J2EE).

Nous avons préconisé l'achat d'un serveur IBM d'entrée de gamme pour assurer une redondance minimale, c'est-à-dire une alimentation de secours et deux disques en raid 0. Le budget ne fut pas accordé. Le choix fut porté sur un poste plus « classique », moins performant. Ce choix se justifie toutefois par le fait, qu'à terme, ces applications doivent être centralisées et disposeront donc de l'architecture technique qu'il se doit au sein du CIT. Cependant, jusque-là, le système s'avère sensible aux pannes.

Nous avons donc fait une demande, qui fut accordée, pour investir dans deux disques de stockage externe, afin de programmer des backups quotidiens et ainsi se prémunir d'une panne du disque principal.

Par la suite, au fur et à mesure de l'avancement du développement des deux projets, nous avons pu réaliser des benchmarks de performance du serveur. Globalement, les performances sont satisfaisantes, et le facteur le plus limitant s'avère être la RAM qui est limitée à 4GO, ce qui, pour un nombre supérieur d'utilisateurs, serait insuffisant.

## **III.2 Gestion du projet**

Nous avons vu le contexte du projet et l'ensemble de sa conception. Dans cette partie, nous nous intéresserons aux éléments de gestion de projet qui ont pu être établis au commencement de celui-ci.

### **III.2.1 SCRUM**

Le contexte du projet était, dès le départ, très favorable à l'adoption d'une méthode agile. En effet, le contact direct avec le métier, et les contraintes posées sur la communication sur le projet, en tiraient fortement parti. Nous avons vu que l'un des

risques majeurs du projet était la non acceptation de celui-ci par les acteurs métier. Pour atténuer ce risque, nous avons décidé de réaliser une forte communication sur le projet, et de placer le métier au cœur du système. La contrepartie était l'évolution permanente des exigences.

De plus, nous avons estimé que des sorties régulières de version permettraient également d'introduire le projet par étape, et donc plus facilement.

Tous ces facteurs nous ont amené à choisir, comme méthode de suivi de projet, « Scrum », méthode agile reconnue. Nous avons alors articulé ce dernier autour des concepts de cette méthode.

Après une première étude assez poussée sur l'ensemble du projet, nous avons priorisé les fonctionnalités au fur et à mesure du développement, en fonction des contraintes.

(Scrum.org, Avril 2014, « Empirical Management Explored », consulté le 25 mai 2014)

(Scrumlop.org, « Published Patterns », consulté le 29 mai 2014)

### III.2.2 Macro planning

Comme élément du cahier des charges et de pilotage du projet, un macro planning nous a permis de définir les principales étapes de celui-ci.

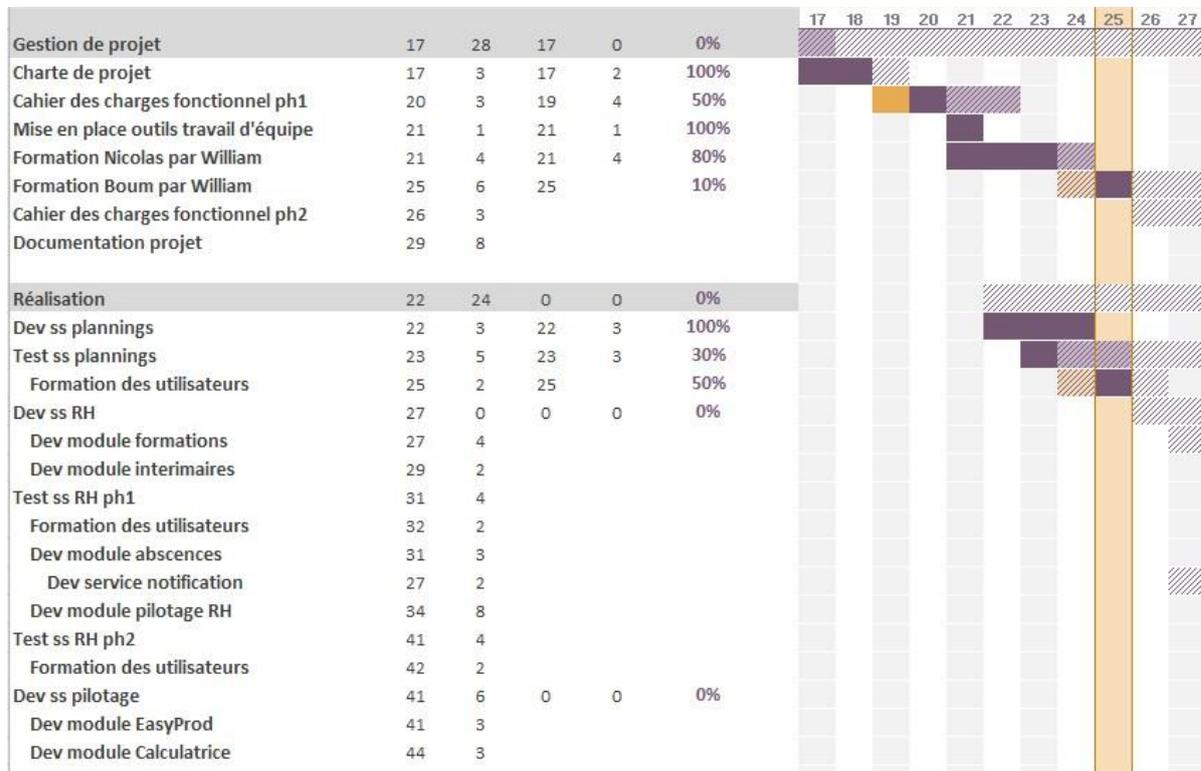


Figure 25: Extrait planification projet

Nous avons suivi ce planning à la semaine, avec des réunions de débriefing hebdomadaires. Cela a permis de réajuster ce planning en fonction de l'avancement global du projet et des missions de support à assurer.



## **IV Développement de l'APS**

### **IV.1 Communication sur le projet**

La communication sur le projet est généralement assurée par le chef de projet. Il a à sa charge la réalisation et présentation des comptes rendu d'avancement pour le comité de direction, mais également d'assurer la relation entre l'équipe de développeurs et les experts métier.

Précédemment, nous avons identifié la communication sur le projet comme une réponse au risque de sa non acceptation.

Nous avons donc impliqué le plus possible les utilisateurs finaux lors des étapes de conception du système. Pour cela, nous organisons des réunions régulières afin de recueillir les nouveaux besoins et les demandes de ces acteurs.

L'ensemble du site compte jusqu'à 33 maitrises. Le projet, lors de son développement, a été concentré sur l'activité « frais » hors fonctionnels, ce qui a réduit ce nombre à 15 acteurs métier majeurs, même si l'ensemble du site a été consulté.

La difficulté était de réaliser des réunions constructives. Nous les organisons autour d'un ordre du jour pré établi, et récupérons les informations via un compte rendu. Ce dernier étant brut d'information, il était repris pour en extraire les exigences du système.

Ce fonctionnement nous a beaucoup apporté pour la conception des différentes fonctionnalités du système. Il se veut aujourd'hui fidèle aux besoins des utilisateurs finaux. Le principal défaut de cette méthode de travail est qu'elle prend énormément de temps. Au-delà de la préparation et du suivi des réunions d'informations elles-mêmes, il faut consacrer du temps à l'écoute des acteurs.



L'idée est de réaliser de manière incrémentale le développement d'un sprint, puis, au cours du sprint suivant, le test du sprint précédent, et de la même manière avec les résolutions de bug. On remarque toutefois que les résolutions de bugs critiques se font en parallèle des tests, étant donné qu'ils peuvent complètement bloquer ceux-ci.

Il faut également préciser que les tests (ici en bleu), incluent des tests préalables, réalisés par l'équipe de développement. Habituellement, ceux-ci sont pris en charge par une équipe dédiée à la qualité, puis ensuite par les acteurs métier eux même.

Nous avons restreint tous les premiers tests à un seul service, pour réaliser, en priorité, les corrections de bugs critiques. Ceci fait, nous testions plus largement le produit pour avoir un retour d'erreur le plus large possible et le plus rapidement.

Dans le cadre du projet, la résolution des bugs a fortement impacté les délais de réalisation du sprint suivant, en raison de la taille de l'équipe de développeur ; mais cela a permis de garantir la qualité du produit. Par exemple, il est arrivé, une fois dans le projet, que ces phases de test ne soient pas concluantes.

En effet, lors des premières phases de test des plannings, nous nous sommes rendu compte que certains cas spéciaux devaient être intégrés. Nous avons donc décidé de reprendre sur une itération de développement pour ces fonctionnalités.

C'est également un bon exemple de ce que permet une méthode agile par rapport à un cycle en V classique : reprendre le développement implique forcément un délai, voir un budget supplémentaire, mais permet de s'assurer que le produit final convient parfaitement aux besoins, et donc garantir sa qualité.

### **IV.3 Mises en production**

Nous avons vu le déroulement des phases de test qui sont, en cas de succès, le passe-droit pour une mise en production de la version.

Il faut savoir que Symfony propose trois environnements d'exploitation d'un projet.

Le premier est l'environnement de développement. Il intègre un certain nombre d'outils très utiles aux développeurs, mais ignore certains warnings qui deviennent des erreurs dans un environnement de développement. Ensuite, cet environnement est important, car il supprime tous les systèmes de cache qui empêchent totalement la résolution des bugs.

Enfin, il y a un environnement dédié aux phases de test, dit de « qualité ». Il propose un juste milieu entre l'environnement de production et de développement car il n'intègre pas les outils dédiés aux développeurs (qui peuvent être une gêne les utilisateurs). Par ailleurs, il n'intègre pas non plus les systèmes de cache, qui compliquent toujours la résolution des bugs et affectent lourdement les performances.

Le dernier environnement est celui de la production. Il intègre tous les systèmes de cache proposés par Symfony. Il permet également l'activation des extensions comme APC (accélérateur PHP) sur le serveur de production.

La mise en production en elle-même est assez simple à réaliser. Il faut surtout prendre garde aux erreurs qui peuvent être émises dans cet environnement et pas dans les autres.

Le changement d'environnement se fait en modifiant un fichier de configuration de Symfony. Il faut toutefois également prendre en compte la configuration du serveur, qui change entre un serveur de développement et un serveur de production.

## IV.4 Développement et processus agile

La phase de développement a été découpée en sprints, comme le suggère SCRUM.

L'enchaînement des sprints est soumis aux conditions suivantes :

- Le sprint précédent est terminé et livré.
- Les prochaines user stories sont plus prioritaires

A la fin de chaque sprint, un bilan et une réunion doivent permettre de valider le sprint et lancer le suivant.

La durée d'un sprint peut varier entre une et trois semaines. Lors du projet, nous avons travaillé sur une base de deux semaines.

Parmi les sous-systèmes développés, la conception puis la réalisation de la planification fut la plus complexe. C'est le seul sous-système qui n'a pas été validé lors de nos phases de tests. Les exigences initialement approuvées se sont retrouvées incomplètes. Le principal problème rencontré fut qu'il a fallu revoir des éléments importants du modèle de données et de certains composants logiciels.

Toutefois, le processus agile nous a permis de réagir assez tôt et de réaliser les modifications étudiées directement. Néanmoins, le planning du projet a dû être réévalué en conséquence. Le reste du développement s'est déroulé normalement en suivant les plannings établis.

Nous notons les pratiques Agile suivantes :

- Recherche de solutions pouvant aider à débloquer une situation par le Scrum Master.
- Travail en équipe sur des tâches difficiles.
- Adaptation et priorisation du besoin face au contexte et aux contraintes (suspension de tâches).

## **IV.5 Avancement du projet**

Le développement et la mise en place du système a demandé plus de temps que le planning initial le prévoyait. Les délais de réalisation n'étaient toutefois pas un élément déterminant du succès du projet. De plus, plusieurs facteurs concernant les ressources sur le projet expliquent les changements et la réévaluation des délais.

L'organisation autour du projet a permis sa continuité. En effet, l'intégration et la formation de personnel a été anticipée.

Parmi les réalisations mises en production, on retrouve :

- Spécification de l'organisation des sites
- Indicateurs de pilotage d'une l'exploitation
- Indicateurs de pilotage d'un groupe de services
- Système de planification des ressources
- Gestion des salariés
- Gestion des demandes intérimaire
- Système de notification

## **V Support, maintenance, et évolutions**

### **V.1 Support et formation**

#### **V.1.1 Manuels d'utilisation**

Les manuels d'utilisation sont un élément important pour l'exploitation de tout le système. C'est également le cas pour l'APS, car même si nous avons pu réaliser des formations avancées et assurer un suivi individuel des utilisateurs d'un site, il n'est pas possible de conserver cette approche dans le cadre d'une mise en place nationale.

Les manuels doivent donc représenter un élément important du support aux utilisateurs.

Pour le projet, nous avons eu la possibilité de confier la réalisation de ces documents à une équipe différente de celle de développement. A terme, ces documents doivent être tenus à jour pour l'équipe que sera chargée de la maintenance du système.

#### **V.1.2 Formation des utilisateurs**

Former les utilisateurs est une tâche d'une complexité très variable. Elle dépend forcément des outils mais également des utilisateurs.

Toutefois, nous avons vu que nous voulions particulièrement accentuer l'aspect formation autour du projet, pour assurer sa bonne intégration et son acceptation par l'ensemble des utilisateurs. Pour garantir cela, nous avons conçu et encadré des formations, ainsi qu'un suivi individuel, à la fin de chaque sprint majeur.

Dans le cas d'un déploiement global, cela posera un problème. Le suivi de cette méthode de formation nécessiterait trop de ressources pour être appliquée sur chaque site. Toutefois, le déploiement global fera l'objet d'un projet à part entière qui traitera cette question.

## **V.2 Evolutions**

### **V.2.1 Fonctionnalités envisagées**

Au cours de l'étude des besoins, nous avons mis de côté l'intégration de la gestion du matériel.

Nous n'avons pas pu terminer l'étude concernant la gestion des ressources matérielles par manque de ressources et de temps. Mais les données récoltées permettent tout de même d'établir qu'il serait intéressant d'intégrer cela au système.

En effet, intégrer la gestion du matériel permettrait notamment la création de nouveaux indicateurs concernant leur gestion. De plus, il existe une liaison entre ressources matérielles et ressources humaines, qu'il n'est pas possible d'établir avec nos outils, mais qui est réalisable au sein du système intégré.

En fait, les objectifs de l'APS que sont la standardisation et l'aide au pilotage, sont assez globaux et devront amener le système à évoluer au plus près des processus métier.

Nous avons également étudié l'intérêt de la réalisation d'une interopérabilité entre notre système et Easycom.

Easycom est un système logiciel développé en Java sur un framework Struts 2. Ce système permet la gestion des retours clients. Ces retours sont gérés par un service dédié : la relation client. Ce service enregistre chaque jour des dizaines de réclamation qui sont la conséquence d'une erreur dans le processus logistique. C'est la mise en relation entre les réclamations client et la qualité qui a été étudiée.

De nombreux facteurs peuvent agir sur la qualité à tous les niveaux du processus logistique. Hors il est intéressant, pour les décideurs, de pouvoir évaluer et étudier ces facteurs dans le but de les réduire.

### **V.2.2 Projet de test multi site**

En prévision d'un déploiement national, il a été établi que le système devrait passer les phases de test sur deux sites.

Le premier site est celui sur lequel il a été conçu, un autre site sera choisi pour son déploiement. L'objectif est de valider la généricité du système. En effet, l'étude de la conception du système même, si elle se veut adaptable à plusieurs organisations, doit être éprouvée.

La question de l'architecture matérielle se pose. Le premier site détient du matériel nécessaire pour l'exploitation du système, il est toutefois le seul site à en disposer. En conséquence, le système devra être déployé sur une architecture matérielle centralisée par CIT.

Pour être intégré à PARIS CIT, le système passera une série de test d'intégration qui leur est propre.

### **V.2.3 Projet de déploiement national**

Si la mise en œuvre du système sur deux sites distincts est réussie, l'objectif du projet reste qu'il soit déployé sur l'ensemble des sites compatibles.

Le système a été conçu dans ce but. Toutefois, plusieurs problématiques sont soulevées par le déploiement global. Ces problématiques doivent faire l'objet d'un projet à part entière qui permettra, en accord avec CIT, d'établir un plan de déploiement.

## **V.2.4 Perspectives**

### **V.2.4.1 Etude des prévisions de vente**

Durant l'étude des besoins, et plus précisément la création des indicateurs, nous avons évalué la faisabilité de réaliser des prévisions d'activité à la semaine et au mois.

Pour évaluer cette faisabilité, nous avons développé des outils qui, par des méthodes mathématiques, devaient estimer le colisage mensuel. L'idée était de suivre des tendances et de les appliquer à notre colisage de référence qui est le budget.

Lors de nos différents essais, la méthode des « moindres carrés » s'est avérée la plus proche de la réalité. Toutefois, le résultat restait peu fiable, tant le nombre de facteurs est important.

L'évaluation de notre activité est directement liée aux prévisions de vente des magasins car nous assurons leur réapprovisionnement. Hors, nous ne possédons pas de modèle fiable pour estimer cela.

Il existe toutefois une technologie qui pourrait, dans ce cadre, s'avérer pertinente : l'étude des Big Data.

#### V.2.4.2 Démarche Big Data

Casino, et plus généralement la grande distribution, ont trouvé de nombreux intérêts dans l'utilisation de cette technologie.

On parle aujourd'hui de démarche Big Data, car les entreprises qui l'ont adopté se sont vite rendu compte que cela posait des problèmes organisationnels. En effet le Big Data se veut, pour être efficace, transverse aux services d'une entreprise, c'est-à-dire que l'on va chercher à utiliser au mieux toutes les données de l'entreprise afin d'en tirer des données nouvelles. Pour cela, un processus décisionnel transverse est nécessaire, ce qui induit de fait une approche globale du système, avec la donnée en son centre.

Les entreprises pionnières ont alors adopté différents modes organisationnels dis « data-driver organisation »

- L'option « décentralisée » : consiste à déployer le Big Data « service par service », c'est-à-dire que la DSI se charge de la mise en place des outils dans chaque service en priorisant le déploiement en fonction de l'importance des données. Généralement les experts métier sont alors sollicités, voir formés, pour réaliser l'exploitation des données. Les entreprises peuvent posséder également un service BI dédié, qui a la responsabilité d'assurer le support sur les projets menés et assister les acteurs décisionnels.
- L'option « centralisée » : consiste, comme son nom l'évoque, en la création d'une entité dédiée au Big Data. Elle a donc la charge de centraliser les données de l'entreprise puis d'effectuer les traitements et l'exploitation de celles-ci. Cette entité doit fonctionner en synergie avec les autres services métier mais également IT. Elle permet alors de diriger l'entreprise vers un mouvement « datacentric » et sera à même de proposer des projets transversaux en profitant de la vision globale qui leur ait donnée.

- L'option de la sous-traitance : c'est l'option actuellement la plus utilisée et probablement la plus simple à adopter. Elle consiste à confier à une entreprise ce qu'elle ne souhaite pas prendre à sa charge. Nous avons vu que les offres de services et solutions sont très nombreux et couvrent l'ensemble du spectre du Big Data depuis le stockage, en passant par l'analyse des données et même jusqu'à leur gestion. Cependant, du fait d'une intégration bien moindre des équipes opérationnelles, l'aspect de la vitesse de transmission de l'information peut en être affecté.

Pour le moment, on ne voit pas apparaître d'organisation type. Toutefois, cela paraît logique, du fait que toutes les entreprises ne partagent pas les mêmes besoins et n'utilisent pas leurs données de la même manière. Encore une fois, cela dépend notamment de leur domaine.

Casino, par sa chaîne de magasin « Monoprix », fut l'un des précurseurs dans ce domaine. L'option choisie par le groupe est celle de la centralisation. Il possède depuis une organisation et des compétences permettant l'intégration de cette technologie.

Grâce à cette démarche, il est possible d'utiliser, sur d'immenses flots de données, des techniques d'analyse comme l'analyse des séries temporelles notamment.

L'utilisation de ces techniques, sur des données de masse, permet de dégager des tendances et des modèles prévisionnels.

La plupart de ces techniques et méthodes demandent des compétences particulières. Les analystes de données massives doivent posséder des connaissances en mathématiques et statistiques avancées mais également en programmation.

L'étude des Big Data d'un système d'information pourrait théoriquement permettre d'établir un modèle pour évaluer les ventes des magasins, ce qui serait pour l'ensemble

du groupe un indicateur très important et qui s'intégrerait parfaitement dans le cadre d'un projet comme celui présenté dans ce dossier.

(Christophe BRASSEUR, « Enjeux et usages du Big Data : Technologies, méthodes et mise en œuvre », Publié par Lavoisier)

(Mike LYNCH, « Les origines de Big Data », 15/03/2012, Disponible sur [http://www.decideo.fr/Les-origines-de-Big-data\_a4987.html])

(Lionel BOUR, Clément MORIZOT, « Big data : comment intégrer les technologies de stockage ? », 26/03/2013, Consultable à l'adresse :

http://www.solucominsight.fr/2013/03/big-data-comment-integrer-les-technologies-de-stockage/])



## Conclusion

Force est de constater que le meilleur des groupes peut encore et toujours améliorer son système d'information. Cette amélioration, en dehors de la "simplification" évoquée dans l'introduction, passe par l'optimisation des outils le composant. L'étude réalisée au sein du groupe Casino nous a permis de comprendre comment l'information peut avoir un impact sur l'entreprise et mis en évidence que l'accessibilité de l'information est un levier de développement pour elle.

Aujourd'hui, le système d'information est une composante cruciale pour la compétitivité des entreprises et des économies. A court terme, une entreprise bien gérée sera une entreprise qui va valoriser l'information initiale en la faisant circuler et en la stockant, d'où l'importance de l'unification des outils du système d'information.

Toutes les entreprises ne sont pas confrontées de façon identique à l'utilisation de son SI. Plus la taille de l'entreprise est conséquente, plus l'utilisation des SI est importante, que ce soit pour la gestion de ses activités métier, le suivi de ses objectifs stratégiques, ou encore la capitalisation d'un savoir-faire. Les industries globales sont donc les plus à même de disposer d'un SI développé.

Pour faciliter l'utilisation de ces SI, un ensemble de technologies se sont développées. Tout au long de ce mémoire, nous avons décrit ces nouveaux outils mis à la disposition des entreprises dans le domaine du management de l'information. Ils sont nombreux et variés. Lequel choisir ?

L'analyse des besoins est la phase essentielle, l'étape primordiale avant toute mise en route de projet. C'est elle qui déterminera le choix de l'outil le mieux adapté. De plus, l'analyse des besoins combinée à l'étude de l'existant permet d'évaluer la justesse des différents indicateurs de pilotage.

Comme nous avons pu le voir, il est ensuite possible soit de refondre un indicateur, soit d'en créer de nouveaux, sachant que la création d'un indicateur n'est jamais aisée, tout d'abord parce qu'il est difficile d'en évaluer sa justesse et sa pertinence avant d'avoir un réel retour sur son utilisation. Ensuite, un indicateur n'est pas qu'une donnée. Enfin, le visuel a son importance. L'ergonomie des tableaux de bords qui regroupe les KPI est essentielle car elle peut aller jusqu'à changer la perception d'un même indicateur.

Créer un outil de pilotage est sans aucun doute un véritable défi technique au vu de l'ensemble des approches nécessaires et des différentes difficultés rencontrées mais l'outil développé permet à l'entreprise d'être performante, et donc de perdurer. C'est un outil au service de l'homme, de l'équipe, de l'entreprise et du groupe. Alors oui, si c'était à refaire, je le referai et n'hésiterai pas à conseiller à de nombreuses entreprises de se lancer dans ce type de projet.

## Bibliographie

- (Reix R. (2002), «Système d'information et management des organisations», Vuibert, 4ème édition, Paris)
- (Cigref , (2002), Gouvernance du système d'information – Problématiques et démarches, Disponible sur [<http://www.cigref.fr/gouvernance-du-systeme-dinformation-problematiques-et-demarches>])
- (Scrum.org, Avril 2014, « Empirical Management Explored », consulté le 25 mai 2014)
- (Scrumplop.org, « Published Patterns », consulté le 29 mai 2014)
- (Christophe BRASSEUR, « Enjeux et usages du Big Data : Technologies, méthodes et mise en œuvre », Publié par Lavoisier)
- (Mike LYNCH, « Les origines de Big Data », 15/03/2012, Disponible sur [[http://www.decideo.fr/Les-origines-de-Big-data\\_a4987.html](http://www.decideo.fr/Les-origines-de-Big-data_a4987.html)])
- (Lionel BOUR, Clément MORIZOT, « Big data : comment intégrer les technologies de stockage ? », 26/03/2013, Consultable à l'adresse : <http://www.solucominsight.fr/2013/03/big-data-comment-integrer-les-technologies-de-stockage/>)



## Liste des figures<sup>1</sup>

Figure 1: Le système d'information : de l'intégration à l'adaptativité.....	12
Figure 2 : Cartographie du Groupe Casino .....	15
Figure 3 : Diagramme des flux existants.....	18
Figure 4 : Extrait de l'outil de suivi des exigences .....	28
Figure 5 : Product Breakdown Structure - APS.....	31
Figure 6 : Schéma organisationnel.....	33
Figure 7: Schéma architecture MVC II.....	37
Figure 8 : Use case - Piloter Service.....	42
Figure 9 : Use case - Plannifier .....	45
Figure 10 : Use case - Exploiter plannings.....	46
Figure 11: Activity diagram - Former salarié.....	47
Figure 12: Activity diagram - Réaliser demande interimaire .....	49
Figure 13: Use case - Gérer salaires .....	50
Figure 14: Use case - Gerer notifications.....	52
Figure 15 : Schéma architecture en couche.....	54
Figure 16 : Schéma architecture framework Symfony.....	55
Figure 17 : Diagramme de classe - Généricité.....	59
Figure 18: Diagramme de classe - Services ressources humaines .....	62
Figure 19: Diagramme de classe - Services Formation.....	63
Figure 20: Diagramme de classe - Plannification .....	65
Figure 21: Schéma intégration système .....	69
Figure 22: Schéma couche intégration détaillée .....	71
Figure 23: Scénario de gestion des accès par Symfony 2 (Source: Documentation officielle Symfony 2) .....	73
Figure 24: Extrait planification projet.....	77
Figure 25: Schéma de mise en œuvre de phases de test au sein d'un processus agile.....	80

---





## ***Unification et création d'indicateurs de pilotage pour les activités logistiques***

**Mémoire d'Ingénieur C.N.A.M., Toulouse 2014**

---

### **RESUME**

Ce document se place dans le contexte d'un système d'information lié aux activités de logistique, et détaille un cas particulier qui est celui du groupe de grande distribution Casino.

Il traite des problématiques d'unification et de pilotage liés aux systèmes d'information. Pour cela, il propose l'étude du contexte de celles-ci, puis l'illustration par un projet de leur gestion par l'intégration de briques métier et à l'aide à la prise de décision.

**Mots clés : Système d'information, Unification, Pilotage, Simplification.**

---

### **SUMMARY**

This document takes place in the context of an information system bound to the activities of logistics, and details a particular case which is the one of the retailing group Casino.

It handles problems of unification and piloting bound to information systems. For that purpose, it proposes the study of the context of these then the illustration with a project of their management by integration of business building blocks and decision support.

**Key words : Information system, Unification, Steering, Simplification.**