

Dédicace

“

Je dédie ce travail :

À ma chère mère et à mon cher père qui n'ont jamais cessé de me supporter, me soutenir et m'encourager durant mes années d'études.

Qu'ils trouvent ici le témoignage de ma profonde gratitude et reconnaissance.

À mes frères, mes grands-parents et ma famille qui me donnent de l'amour et de la vivacité.

À tous ceux qui m'ont aidé - de près ou de loin - et ceux qui ont partagé avec moi les moments d'émotion lors de la réalisation de ce travail et qui m'ont chaleureusement supporté et encouragé tout au long de mon parcours.

À tous mes amis qui m'ont toujours encouragé, et à qui je souhaite plus de succès.

Merci !

”

- Younes

Remerciements

Je tiens, avant de présenter mon travail, à exprimer ma grande reconnaissance envers les personnes qui m'ont - de près ou de loin - apporté leurs soutiens. Qu'ils trouvent ici collectivement et individuellement l'expression de toute ma gratitude et ma reconnaissance.

Je tiens à remercier tout particulièrement et à témoigner toute ma reconnaissance à M. **AMRI Faouzi** et à Mlle. **HADI Halima**, pour l'expérience enrichissante et pleine d'intérêt qu'ils m'ont fait vivre durant la période du stage, pour tous les conseils et les informations qu'ils m'ont prodigués et pour le temps qu'ils ont consacré à l'encadrement et le suivi de ce travail.

Je tiens à remercier également mes professeurs encadrants M. **AMNAI Mohamed** et M. **NASRI Mohammed**, en dépit de leurs multiples charges, leurs aides et les renseignements précieux qu'ils m'ont fournis ainsi que pour tous les conseils et les informations qu'ils m'ont prodigués avec un degré de patience et de professionnalisme sans égal.

Je tiens aussi à adresser mes plus sincères remerciements à l'ensemble du corps administratif et enseignant de l'ENSA Khouribga, pour avoir porté un vif intérêt à notre formation, et pour avoir accordé de l'attention et de l'énergie, et ce, dans un cadre agréable de respect.

Que les membres de jury trouvent, ici, l'expression de mes remerciements pour l'honneur qu'ils me font en prenant le temps de lire et d'évaluer ce travail.

Résumé

Le présent rapport constitue une synthèse de mon projet de fin d'études effectué au sein de la société SQLI Rabat et ayant comme objectif la mise en œuvre d'une solution PIM (Product Information Management) en faveur d'une maison de produits de luxe à base de la plateforme e-commerce SAP Hybris, et ce en vue de l'obtention du diplôme d'ingénieur d'état en informatique.

Ce projet vise à permettre à la société cliente qu'est une maison multinationale de vente des produits de luxe, de pouvoir gérer les informations relatives à ses produits d'une façon centralisée, afin de les utiliser dans plusieurs endroits, que ce soit au niveau de ses réseaux intranet et extranet ou pour alimenter les divers sites web, applications mobiles et points de vente qu'elle possède à travers le monde. Avec la solution proposée la société cliente peut exercer un contrôle de qualité et remédier aux problèmes liés à la redondance et la non-cohérence des informations de ces produits qui peuvent être parfois assez critiques pour la réputation de l'entreprise vis-à-vis de ses partenaires et clients.

L'étude technique a été faite conjointement entre SQLI et la Maîtrise d'Ouvrage, vu l'expertise de SQLI dans les outils utilisés qui ont prouvé une efficacité inégale sur d'autres projets similaires. Durant ce stage, nous avons opté pour la méthodologie SCRUM pour la gestion et la conduite du projet.

Mots clés : é-commerce, gestion de l'information produit, PIM, Java EE, SAP commerce, Hybris, SCRUM, SQLI.

Abstract

In order to get my degree in computer sciences, this report summarizes the work carried out within SQLI-Rabat Company for my graduation project, which consists of implementing and customizing a PIM (Product Information Management) solution for a luxury products company based on the SAP Hybris eCommerce platform.

The project basically aims to enable the client company to manage their products information in a centralized way in order to be used in several places such as the extranet and the intranet, as well as in the different websites, mobile applications and selling points they have across the world. This will allow the company to control the quality of those information and avoid the problems resulting from their redundancy and non consistency, especially that they might be sometimes quite critical for the company to maintain a good reputation in front of its customers and partners.

The technical study was made by SQLI in cooperation with the client, since SQLI gained a huge experience in the tools used. The project was managed through an agile methodology which is SCRUM, in order to gain in flexibility and to better identify and respond to our client's needs.

Keywords : e-commerce, product information management, PIM, Java EE, SAP commerce, Hybris, SCRUM, SQLI.

Table des matières

| | |
|------------------------------------------------------|-----------|
| Introduction | 10 |
| 1 Contexte général du projet | 12 |
| 1.1 Présentation de l'organisme d'accueil | 13 |
| 1.1.1 Carte d'identité | 13 |
| 1.1.2 Métiers SQLI | 14 |
| 1.1.3 Organigramme | 14 |
| 1.1.4 Fondamentaux SQLI | 16 |
| 1.2 Contexte du projet | 17 |
| 1.2.1 Cadre du projet | 17 |
| 1.2.2 Problématique | 17 |
| 1.2.3 Solution proposée | 17 |
| 1.3 Planification et conduite du projet | 20 |
| 1.3.1 Processus de développement | 20 |
| 1.3.2 Planning du projet | 21 |
| 1.4 Conclusion | 23 |
| 2 Analyse et spécification des besoins | 24 |
| 2.1 Spécification des besoins fonctionnels | 25 |
| 2.2 Analyse des besoins fonctionnels | 26 |
| 2.2.1 Identification des acteurs | 26 |
| 2.2.2 Diagrammes des cas d'utilisation | 28 |
| 2.2.3 Conclusion | 40 |

| | |
|---------------------------------------------------------------|-----------|
| 3 Étude Conceptuel et architectural | 41 |
| 3.1 Introduction | 42 |
| 3.2 Modèle de données | 42 |
| 3.2.1 Catalogue et versions | 42 |
| 3.2.2 Les produits | 44 |
| 3.2.3 Les catégories | 48 |
| 3.2.4 Les medias | 49 |
| 3.3 Le flux d'import | 50 |
| 3.4 Architecture Logiciel du projet | 54 |
| 3.5 Architecture physique du projet | 55 |
| 3.6 Conclusion | 56 |
| 4 Mise en oeuvre de la solution | 57 |
| 4.1 Introduction : | 58 |
| 4.2 Outils et technologies utilisées : | 58 |
| 4.2.1 La plateforme Java EE | 58 |
| 4.2.2 La plateforme SAP Hybris | 58 |
| 4.2.3 Outils de développement et de collaboration : | 60 |
| 4.3 Quelques interfaces de la plate-forme | 60 |
| 4.3.1 La page de Login | 60 |
| 4.3.2 La Vue Administration | 61 |
| 4.3.3 La Vue Catalogue | 63 |
| 4.3.4 La Vue Catégories | 63 |
| 4.3.5 La Vue Dashboard | 64 |
| 4.4 Conclusion | 65 |
| Conclusion générale | 66 |
| Webographie | 67 |

Table des figures

| | | |
|-----|----------------------------------------------------------------------|----|
| 1.1 | Logo de SQLI. | 13 |
| 1.2 | Quelque clients de SQLI groupé par secteur d'activité. | 13 |
| 1.3 | Organigramme de SQLI Maroc. | 15 |
| 1.4 | Schéma illustratif d'une solution PIM | 18 |
| 1.5 | Schéma illustratif de la solution proposé | 19 |
| 1.6 | Cycle de vie de la méthodologie scrum. | 20 |
| 1.7 | Diagramme de gantt du projet. | 22 |
| | | |
| 2.1 | Héirarchie des profiles humaines. | 26 |
| 2.2 | Les cas d'utilisation du profil Admin. | 28 |
| 2.3 | Les cas d'utilisation du profil Catalog Manager. | 34 |
| 2.4 | Les cas d'utilisation du profil Support Member. | 34 |
| 2.5 | Les cas d'utilisation du profil Contributor. | 36 |
| 2.6 | Les cas d'utilisation du profil Centrale Translator. | 36 |
| 2.7 | Les cas d'utilisation du profil Approver. | 37 |
| 2.8 | Les cas d'utilisation du profile Reader. | 38 |
| 2.9 | Les cas d'utilisation d'import des produits et des médias. | 39 |
| | | |
| 3.1 | Aperçu du diagramme de classes du catalogue et ses versions. | 42 |
| 3.2 | Le principe du catalogue des produits d'Hybris. | 43 |
| 3.3 | Récapitulatif du processus de Synchronisation. | 44 |
| 3.4 | Récapitulatif de l'abstraction du produit. | 45 |
| 3.5 | Diagramme de classes des produits. | 47 |
| 3.6 | Diagramme de classe des catégories. | 49 |

| | | |
|------|------------------------------------------------------------------------|----|
| 3.7 | Diagramme de classe des médias. | 50 |
| 3.8 | Processus d'import des produits. | 51 |
| 3.9 | Processus de splitting des fichiers xml. | 52 |
| 3.10 | Splitting d'un fichier multiple vers des fichier unitaire. | 52 |
| 3.11 | Architecture logiciel du projet. | 54 |
| 3.12 | Architecture physique du projet. | 55 |
| 4.1 | La page du login de la plateforme PIM. | 61 |
| 4.2 | La vue Administration de la plate-forme PIM. | 62 |
| 4.3 | La vue Administration avec un type de catégories sélectionnée. | 62 |
| 4.4 | La vue Catalogue de la plate-forme. | 63 |
| 4.5 | La vue Categories de la plateforme. | 64 |
| 4.6 | La vue Dashboard de la plate-forme. | 65 |

Liste des sigles et acronymes

| | |
|-------------|-------------------------------------------|
| DAM | <i>Digital Asset Management</i> |
| ERP | <i>Enterprise Resource Planning</i> |
| EAI | <i>Enterprise Application Integration</i> |
| HAC | <i>Hybris Administration Console</i> |
| MEP | <i>Mise En Production</i> |
| OAT | <i>Operational Acceptance Testing</i> |
| OOTB | <i>Out Of The Box</i> |
| PIM | <i>Product Information Management</i> |
| PCM | <i>Product Content Management</i> |
| TFS | <i>Team Foundation Server</i> |
| UAT | <i>User Acceptance Testing</i> |
| WFJ | <i>Watches and Fine Jewelry</i> |

Introduction

Parmi les principales activités de SQLI, figure l'offre d'une panoplie de services en e-commerce, allant de l'intégration jusqu'au développement des applications et solutions e-commerce qui répond aux besoins spécifiques de chaque client.

La Maîtrise d'Ouvrage dans notre cas est une maison de ventes des produits de luxe qui a commencé à investir dans le commerce électronique depuis 2010 pour vendre des produits de la haute couture commandés par ses clients, et ce afin de répondre à leurs besoins en termes de services en ligne à travers le maillage de trois divisions : la Mode (Fashion), les Parfums et Beauté (fragrance and beauty) et l'Horlogerie Joaillerie (Watches And Fine Jewelry).

Notre client adopte la stratégie é-commerce multicanal, ce qui signifie la commercialisation des produits via plusieurs canaux. Ces canaux peuvent inclure des points de vente physiques, des catalogues, des magasins de commerce électronique ou des application mobiles.

C'est ainsi que l'entreprise cliente s'est retrouvée avec les informations de ces produit stockées dans différents endroits, parfois dans différents services avec des interlocuteurs multiples. Or, arriver à maintenir une bonne qualité et fiabilité de ces informations entre ces différents endroits est une opération très fastidieuse et coûteuse vue les problèmes de synchronisation et le taux d'erreur élevé.

Notre client a sollicité les services de SQLI qui est reconnue par son expertise dans le domaine de l'e-commerce afin de lui concevoir une solution permettant la centralisation de la gestion et l'enrichissement des informations de ces produits afin de standardiser et unifier ces informations entre toutes les filiales et les sites qui sont alimentés par ces informations autour du monde.

Le présent rapport synthétise ainsi le déroulement de mon travail sur ce projet. Il est structuré en trois chapitres :

Le premier chapitre comporte une brève présentation de l'organisme d'accueil SQLI Rabat et du cadre général de ce projet. Il expose en effet, la problématique et met l'accent

sur la solution proposée. Et il aborde à la fin la méthodologie appliquée pour assurer le bon déroulement de mon travail avec une planification de ce dernier.

Le deuxième et le troisième chapitre présentent les phases d'analyse et de conception de mon projet. Dans ces chapitres je détaille donc les besoins fonctionnels et non fonctionnels et j'aborde l'analyse de ces besoins et puis l'étude technique, conceptuelle et architectural de la solution.

Dans le quatrième chapitre, il y a une présentation de la phase de réalisation du projet. Il présente les différents outils et les techniques utilisés ainsi que le résultat d'implémentation de la solution.

En fin je clôture par une conclusion générale qui présente une analyse du travail réalisé au sein de SQLI. Cette partie met en évidence non seulement un résumé du travail effectué, mais aussi des propositions et des diverses perspectives.

Chapitre 1

Contexte général du projet

“ Ce chapitre a pour but de situer le projet dans son environnement organisationnel et contextuel. Il présente dans sa première partie l’organisme d’accueil, tandis que sa deuxième partie décrit les objectifs ainsi que les exigences de base de la solution PIM (Product Information Management ou gestion des informations des produits) La dernière partie est réservée à la démarche et la conduite adoptée pour la réalisation du projet. ”

1.1 Présentation de l'organisme d'accueil

1.1.1 Carte d'identité



FIGURE 1.1 – Logo de SQLI.

SQLI dont le logo est présenté dans la figure 1.1 est une société française créée en 1990, pour accompagner les entreprises dans l'utilisation des nouvelles technologies [1].

Elle s'est spécialisée dans la réalisation des systèmes d'informations de la nouvelle génération. Elle est organisée en agences de proximité, afin de conserver le maximum de réactivité face aux besoins de ses clients. En prenant en compte, au sein même de son organisation, les préoccupations du tissu économique régional, SQLI offre une approche sur mesure aux enjeux spécifiques des entreprises.

SQLI compte plus de 1200 clients, grands comptes et PME, issus de tous les secteurs d'activité comme le montre la figure 1.2.

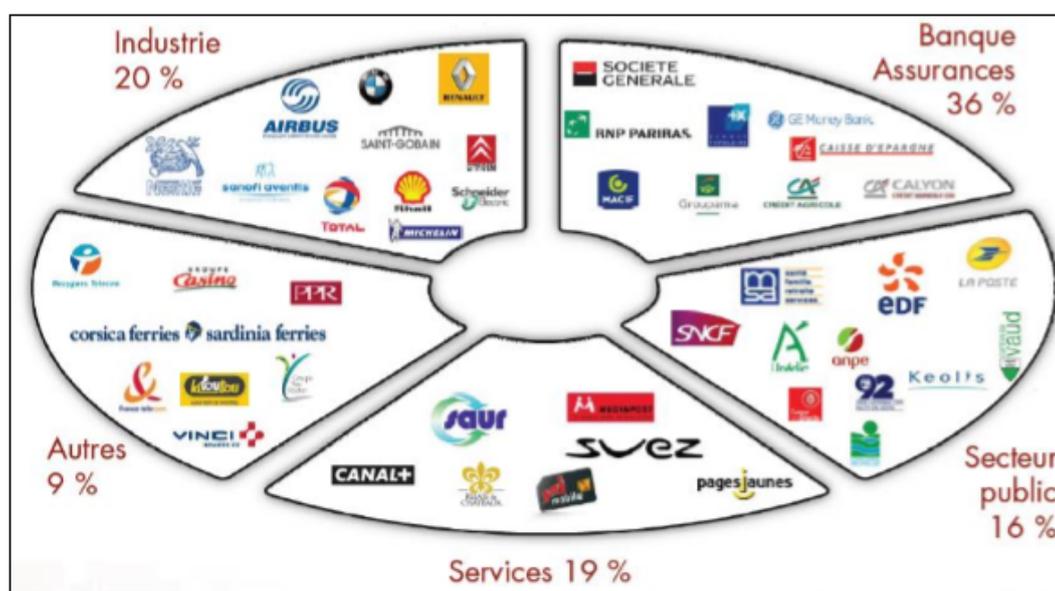


FIGURE 1.2 – Quelques clients de SQLI groupé par secteur d'activité.

SQLI comporte plus de 2000 collaborateurs répartis dans les 32 agences dont 11 en France (Paris, Lyon, Toulouse, Bordeaux, Rouen, Nantes ...), en Suisse (Lausanne et Genève), au Luxembourg, en Belgique (Bruxelles), aux Pays-Bas et au Maroc (Rabat et Oujda).

1.1.2 Métiers SQLI

SQLI s'est spécialisée dans les projets e-commerce, liés aux systèmes d'information intégrant l'utilisation des technologies internet. Pour aider les entreprises à tirer parti des technologies web, SQLI propose un accompagnement global sur tout le cycle du projet , à savoir :

- Des prestations de conseil pour aider les clients à faire les bons choix.
- La mise en œuvre concrète de ces choix par la réalisation et l'intégration.
- Un accompagnement dans le déploiement des projets et le transfert de compétences.

Le groupe SQLI fédère toutes les compétences indispensables au bon déroulement des projets de ses clients, du conseil à la réalisation en passant par l'ergonomie, le design, l'interface utilisateur et la formation.

Ainsi le métier du groupe SQLI est organisé en quatre pôles :

- Le pôle « Ingénierie et intégration ».
- Le pôle « Conception web : Studio SQLI ».
- Le pôle « Stratégie en Systèmes d'Information ».
- Le pôle « Formation et Transfert de compétences ».

1.1.3 Organigramme

Les différents centres de Rabat et Oujda étaient indépendants mais aujourd'hui avec la nouvelle organisation les deux sites ont été fusionnés en donnant naissance à une seule entité.

L'objectif de cette transformation est de gagner en synergie et donner une lecture plus simple de l'organisation.

La figure 1.3 montre le nouvel organigramme de SQLI Maroc :

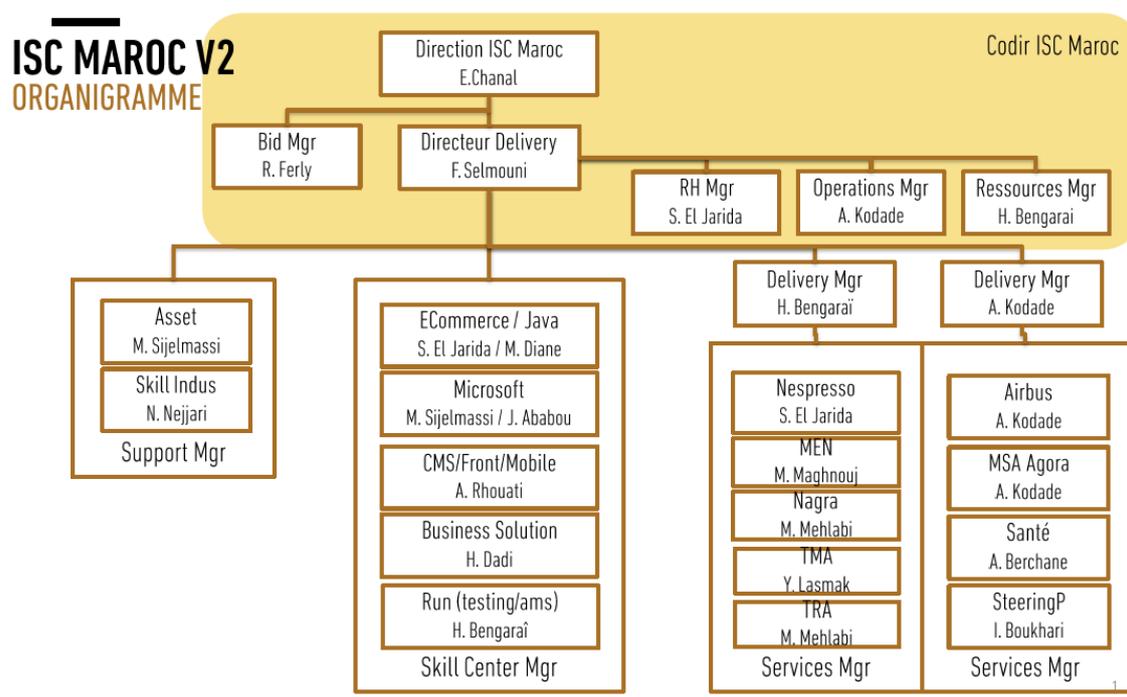


FIGURE 1.3 – Organigramme de SQLI Maroc.

Eric Chanal assure la Direction globale de l'ISC Maroc en s'appuyant sur **Fouad Selmouni** pour les livraisons et **Renaud Ferly** pour la détection des futurs projets et la bonne coordination des avant-ventes.

Les deux sites passent sous la responsabilité du directeur, qui a un focus Delivery global afin de s'assurer que les produits sont de haute qualité pour les clients avec des moyens de production adéquats.

Pour cela, il aura sous sa responsabilité :

Saïd el Jarida : Responsable RH sur les deux sites. Il veillera au bon déroulement des processus RH et proposera des actions pour améliorer les conditions de travail et la montée en expertise des collaborateurs.

Abderahmane Kodade : Operations Manager sur les deux sites. Son rôle est d'anticiper le résultat des activités pour les mois à venir afin de prendre au plus tôt les meilleures décisions et d'être prédictible dans les prévisions pour le groupe.

Hicham Bengarai : devient Ressource Manager sur les deux sites. Il sera informé de tous besoins de recrutement et il optimisera les affectations sur les projets pour utiliser au mieux les ressources sur les 2 sites.

1.1.4 Fondamentaux SQLI

L'excellence opérationnelle

C'est un engagement de résultat reposant sur CMMI qui est un modèle de référence, un ensemble structuré de bonnes pratiques, destiné à appréhender, évaluer et améliorer les activités des entreprises d'ingénierie [2].

1ère société de service informatique européenne à avoir intégré le modèle en 2002, SQLI a obtenu une certification CMMI de niveau 3 (Scampi A "Standard CMMI Appraisal Method for Process Improvement" qui est la méthode officielle d'évaluation du Software "Engineering Institute" concernant les modèles CMMI) en 2006, et déploie le niveau au sein de la plupart des agences.

Le savoir être

Le savoir être, c'est une sélection et une fidélisation des meilleurs collaborateurs via People- CMM. SQLI est la seule entreprise du marché à combiner depuis 2004 CMMI et People- CMM, une démarche de qualité 100% dédiée au management de ses Ressources Humaines et de ses compétences, tant sur le plan individuel que collectif.

La relation client

La relation client, c'est une écoute attentive pour comprendre les métiers, les enjeux, l'organisation cliente et son besoins. Cela se fait au travers d'un programme de qualité inédit, mis au point par le Groupe depuis 2006 : Business CMM.

L'innovation

L'innovation, c'est un laboratoire nommé 6mmx pour proposer des solutions toujours plus innovantes : performance fonctionnelle, productivité, rentabilité, et génératrices de valeur. Basé sur l'intelligence collective, ce tout nouveau programme « made in SQLI » est déployé à l'échelle du groupe, pour gérer l'innovation de façon originale et pragmatique.

1.2 Contexte du projet

1.2.1 Cadre du projet

Notre projet intitulé « Mise en œuvre et personnalisation d'une solution PIM à base de la plate-forme SAP Hybris » a été proposé dans le cadre de l'élaboration d'un stage de fin d'études au sien de la société SQLI pour répondre au besoin de l'un de ses clients qu'est une maison internationale de vente des produits de luxe.

1.2.2 Problématique

Notre client est un géant dans le monde de vente des produits de luxe. Il a commencé à investir dans le commerce électronique en 2010 pour vendre des produits de la haute couture commandés par ses clients, et ce afin de répondre à leurs besoins en matière de services en ligne.

Ainsi pour réduire le "Time To Market" et acquérir le plus possible de ses clients, notre client a adopté la stratégie e-commerce multicanale pour vendre ces produits dans différents canaux, à savoir des points de vente autour du monde, des sites web e-commerce, des applications mobiles ou bien des Boutique Apps. Notre client s'est affronté donc à la situation à laquelle une énorme base de données des informations de ces produits étaient utilisés et réparties dans plusieurs endroits et parfois les informations relatives à un même produit se retrouvent différentes d'un endroit à l'autre.

Or les conséquences d'une information erronée à propos d'un produit peuvent être importantes, que ce soit pour des raisons juridiques, techniques (caractéristiques de l'offre), commerciales (disponibilité ou non des produits) ou marketing (insatisfaction des clients).

1.2.3 Solution proposée

Hypothèse de la solution

La solution proposée consiste à mettre en place une plate-forme PIM (Product Information Management) unique qui regroupe toutes les opérations de gestion, de contrôle et de maintenance des informations dans un seul endroit et qui permet en suite d'exporter ces informations vers les autres canaux qui sont alimentés par ces produits [3].

Une telle plate-forme permettra ainsi de :

- Garantir que les informations relatives à un produit sont les mêmes dans tous les canaux.

- Faciliter le contrôle et la classification des produits.
- Enrichir et traduire les informations des produits.

Introduction au PIM

Un PIM (Product Information Management) est un système qui sert comme référentiel unique et centralisé de toutes les données des produits d'une entreprise. D'un point de vue technique, un PIM permet d'extraire et harmoniser les données et de diffuser ces données vers ou à partir de différentes sources. D'un point de vue fonctionnel, un PIM permet de classer, d'enrichir et de traduire les informations de ces produits[4].

La figure 1.4 constitue un schéma général d'une solution PIM :

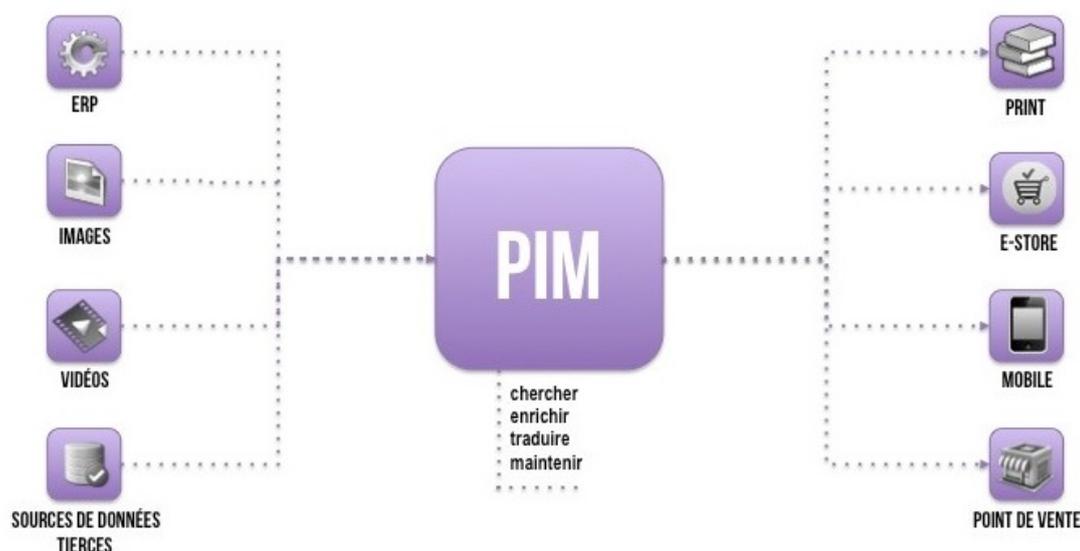


FIGURE 1.4 – Schéma illustratif d'une solution PIM

Ainsi on peut résumer le fonctionnement d'un tel système en trois étapes principales :

Extraire les données

- Recueillir des données à partir de sources existantes, ERP ou bases de données de fournisseurs.
- Nettoyer et choisir les données que vous souhaitez intégrer à vos catalogues.
- Définir les priorités entre les différentes sources de données.

Enrichir les données

- Enrichir les produits avec les informations marketing ou les données techniques.

- Classer les produits suivant une ou plusieurs arborescences.
- Traduire les données en plusieurs langues.
- Gérer les processus du cycle de vie des données (flux de travail et audits).
- Contrôler la qualité des données (exhaustivité, doublons, ...).

Distribuer les données

- Gérer plusieurs canaux de vente comme des sites web e-commerce, print (pour l'impression des produits Prêt-à-Porter comme les T-shirts, chapeaux, chaussures, ...), applications mobiles, réseaux sociaux, places de marché, etc...
- Sélectionner les listes de produits et l'arborescence que l'on souhaite distribuer à travers chaque canal.
- Envoyer des données vers des logiciels d'impression de catalogues comme Indesign et Easycatalog pour créer des catalogues imprimés.

Implémentation proposée

La solution PIM qu'on propose dans notre projet consiste à récupérer les produits avec un flux d'import à partir de l'ERP central du client ainsi que leurs médias à partir du DAM du client aussi, pour enfin gérer et enrichir les informations de ces produits et ensuite les distribuer avec un flux d'export vers un ensemble de canaux qui sont alimentés par ces produits (voir la figure 1.5).

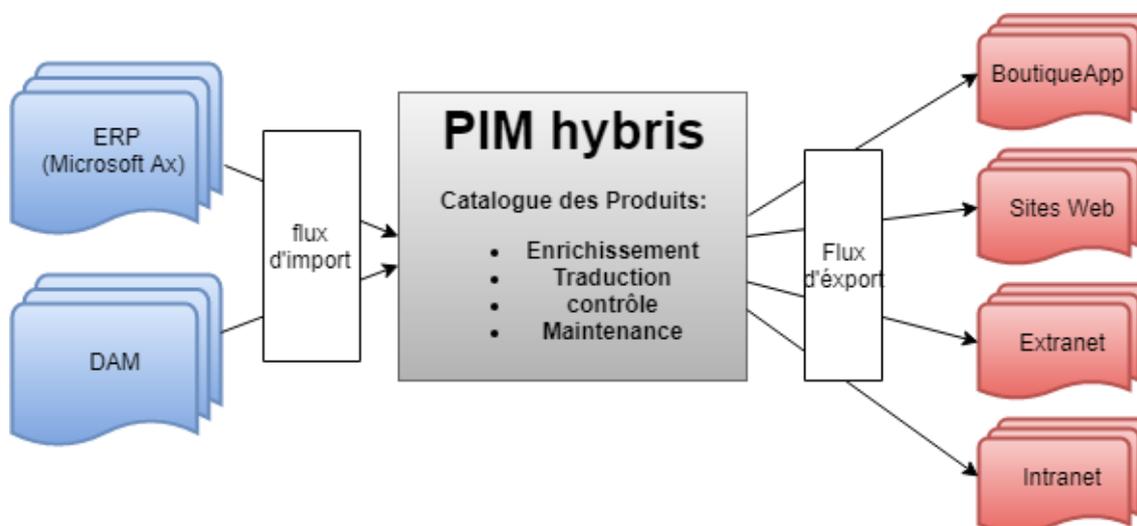


FIGURE 1.5 – Schéma illustratif de la solution proposé

Dans notre projet, nous proposons la mise en oeuvre d'une telle solution en se basant sur la plate-forme e-Commerce SAP Hybris. Ce choix se justifie par sa flexibilité ainsi que

les éléments OOTB (Out Of The Box ou prêt à utilisé) qu'il offre pour les grands solutions e-commerce.

Ainsi nos objectifs à travers se projet consiste à centraliser les accès aux données, et de fournir des interfaces automatisées de maintien et de consommation de données (imports, exports, enrichissement, traduction etc...).

1.3 Planification et conduite du projet

1.3.1 Processus de développement

La nature du projet et sa forte dépendance aux acteurs du domaine sont les raisons qui expliquent le fait d'être toujours à l'écoute du client et prêt à répondre à ses nouveaux besoins. C'est pour cela, l'équipe du projet a opté pour un cycle de développement agile et plus précisément SCRUM.

Le principe de la méthodologie SCRUM est de développer un logiciel de manière incrémentale en maintenant une liste totalement transparente des demandes d'évolutions ou de corrections à implémenter (backlog).

Avec des livraisons très fréquentes, toutes les 4 semaines en général, le client reçoit un logiciel à chaque itération. Plus nous avançons dans le projet, plus le logiciel est complet et possède de plus en plus de fonctionnalités.

Pour cela, la méthode s'appuie sur des développements itératifs à un rythme constant d'une durée de 2 à 4 semaines (2 semaines pour notre cas) comme le montre la figure 1.6 :

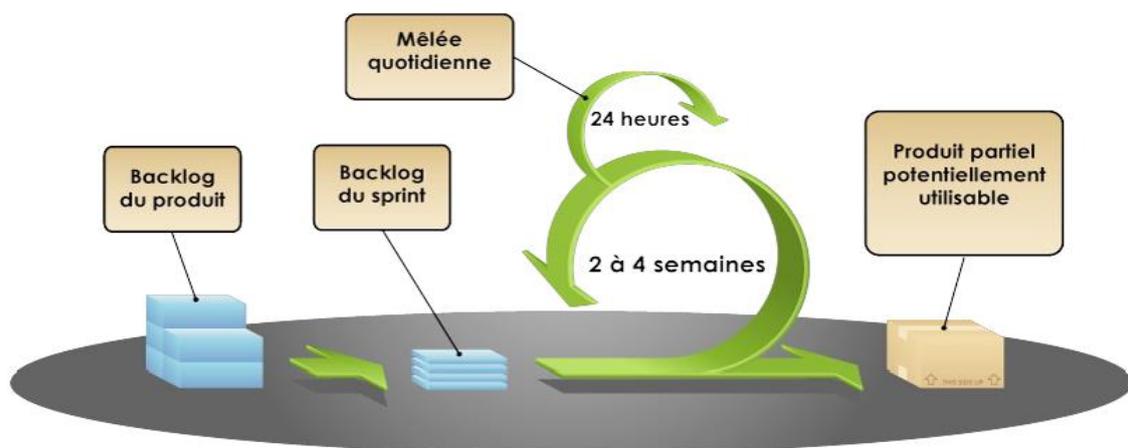


FIGURE 1.6 – Cycle de vie de la méthodologie scrum.

Le schéma illustre un exemple de planification en Scrum : les itérations (sprints) durent en pratique entre 2 et 4 semaines, et possède chacune un but. Le but de chaque sprint une liste d'items du backlog de produit ou de fonctionnalités à réaliser. Ces items sont décomposés par l'équipe en tâches élémentaires de quelques heures.

Comme nous pouvons le remarquer dans cette figure, pour mettre en place la méthode SCRUM, il faut tout d'abord définir les différentes fonctionnalités de notre application qui forment le backlog du produit. Ensuite, vient l'étape de la planification du sprint pour définir le plan détaillé d'une itération.

Durant un sprint, il y a toujours des réunions quotidiennes entre les différents collaborateurs du projet afin de présenter l'état d'avancement des différentes tâches en cours, les difficultés rencontrées ainsi que les tâches restantes à réaliser. Une fois le produit partiel est prêt, nous vérifions la conformité de ce qui a été fait durant le sprint et nous pouvons alors l'améliorer en procédant à l'étape de rétrospective.

La méthodologie SCRUM fait intervenir 3 rôles principaux qui sont :

- **Product Owner** : Dans la majorité des projets, le responsable produit (product owner) est le responsable de l'équipe projet client. C'est lui qui va définir et prioriser la liste des fonctionnalités du produit et choisir la date et le contenu de chaque sprint sur la base des valeurs (charges) qui lui sont communiquées par l'équipe.
- **ScrumMaster** : Véritable facilitateur sur le projet, il veille à ce que chacun puisse travailler au maximum de ses capacités en éliminant les obstacles et en protégeant l'équipe des perturbations extérieures.
- **Équipe de développement** : elle regroupe l'ensemble des rôles habituellement nécessaires à un projet, à savoir le concepteur, le développeur, le testeur, etc. L'équipe s'organise elle-même et elle reste inchangée pendant toute la durée d'un sprint.

1.3.2 Planning du projet

La planification du projet est une phase importante d'avant-projet. Elle consiste à prévoir le déroulement de ce dernier tout au long des phases constituant le cycle de développement.

Le diagramme de Gantt dans la figure 1.7 illustre le déroulement du stage dans le temps :

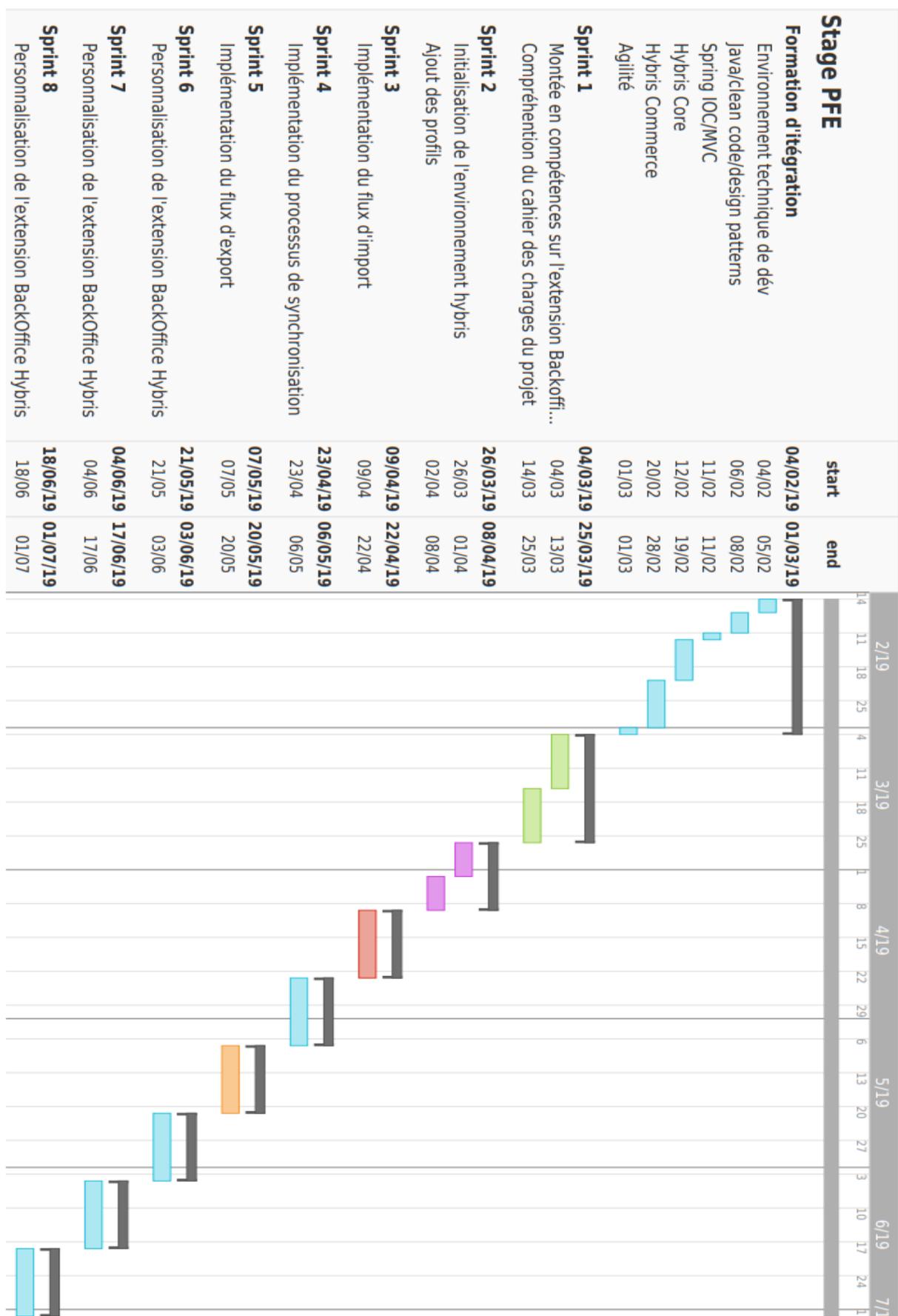


FIGURE 1.7 – Diagramme de gantt du projet.

L'entreprise SQLI, comme la plupart des SSII (Société de Services et d'Ingénierie Informatique), est divisée en équipes projets, chaque équipe travaille sur un projet de manière quasi indépendante, le chef de projet assure les fonctions RH quotidiennes et le Scrum Master s'occupe de tout le cycle de vie logiciel.

Notre stage a commencé par une phase d'intégration qui s'est étalée du lundi 4 Février 2019 au vendredi 01 Mars 2019. Cette phase comportait principalement des formations sur différents aspects et technologies et fût animée par des collaborateurs de SQLI Rabat.

Les formations programmées consistaient tout d'abord en quelques rappels concernant les bases du langage JAVA et des principes de développement mais aussi des technologies utilisées dans le projet (le système de contrôle des versions GIT, l'outil Atlassian JIRA pour le suivi des tickets, la plate-forme e-commerce SAP Hybris etc ..).

Le plan d'intégration devait surtout faire une montée en compétence des collaborateurs sur des points critiques du projet, surtout en ce qui concerne les principes du clean code et aussi une introduction au principe de l'agilité adoptée dans les projets menés par SQLI.

Après la formation d'intégration, nous avons rejoint les équipes de développement pour enfin travailler sur les projets clients.

Au début, nous avons été menés à une phase de montée en compétence dans l'extension Backoffice de la plate-forme SAP Hybris, puisque il constitue l'extension principale utilisé dans le projet. Nous avons passé ensuite à la première itération qui concerne l'étude fonctionnelle ainsi que la capture et l'analyse des besoins fonctionnels, et ensuite le reste des itérations ou sprints de l'implémentation du projet.

Les sprint étaient d'une durée de deux semaines pour chacune dans lesquels il y avait les processus de développement et de test de chaque fonctionnalité ajouté et ainsi dans chaque deux à trois mois il y avait un processus de MEP (Mise en production) de la solution.

1.4 Conclusion

Dans ce chapitre, nous avons présenté en premier lieu l'organisme SQLI. Dans un second lieu, nous avons déterminé le cadre du projet. Ensuite nous avons spécifié la problématique du projet. Et en dernier lieu, nous avons présenté la méthode de travail à adopter et illustré le déroulement du travail avec le planning suivi tout au long du projet. Dans le chapitre qui suit, nous présentons l'analyse fonctionnelle et non fonctionnelle de notre projet ainsi que la conception détaillée, phase dans laquelle, on a défini les spécifications dans le but d'élaborer l'architecture globale du projet.

Chapitre 2

Analyse et spécification des besoins

“ Dans ce chapitre, j’aborde les phases d’analyse et de spécification des besoins du projet, dans le but d’avoir une vision globale claire du comportement du projet ainsi que les attentes des utilisateurs. ”

2.1 Spécification des besoins fonctionnels

La capture des besoins fonctionnels est une étape importante du projet. Cette étape produit le dossier des spécifications fonctionnelles, et durant laquelle les fonctionnalités attendues sont formalisées ainsi que toutes les règles de gestion régissant. Les besoins fonctionnels détectés se résument dans les points suivants :

Gestion des profils :

Le projet doit comporter plusieurs profils qui permettent d'attribuer aux utilisateurs du PIM un accès en lecture et/ou écriture en fonction des entités, perspectives (vues), actions et langues pour garantir la sécurité des données des produits.

Flux d'import :

Le socle d'import doit permettre d'importer des données depuis des systèmes tiers vers le PIM. Il permet, par exemple, d'importer des produits, des données référentielles, des médias ou encore des prix de produits et cela d'une manière automatisée.

Gestion du catalogue :

Le projet doit permettre la gestion du catalogue des produits afin de segmenter les données et pour distinguer les données en cours d'enrichissement des données à publier.

Gestion des catégories :

La plate-forme doit permettre de regrouper les produits dans des catégories et des systèmes de classification. Ces catégories peuvent être structurées en arborescences.

Gestion des produits :

La plateforme doit permettre une gestion avancée des informations des produits avec des vues intuitives et ergonomique avec des fonctionnalités de recherche et filtrage puissantes.

Flux d'export :

La solution doit permettre un export personnalisé avec différents formats et suivants les différents canaux à alimenter.

2.2 Analyse des besoins fonctionnels

L'analyse des besoins est une étape très importante dans le processus de l'étude et le développement des systèmes d'informations. Cette partie identifie l'ensemble des acteurs qui interagissent avec le système et définit l'ensemble des cas d'utilisation de ce dernier en se basant sur les diagrammes UML.

2.2.1 Identification des acteurs

Un acteur est une personne, un matériel ou un logiciel qui interagit avec le système. L'analyse du présent projet commence par une identification des acteurs agissant sur les différentes parties du système. Les acteurs présentés dans la figure 2.1 sont des employés du clients en plus du serveur EAI Microsoft BizTalk (Entreprise Application Integration) qui est la solution adoptée par le client pour la communication et le partage des informations entre ces systèmes et départements (ERP, PIM, sites web, etc ...).

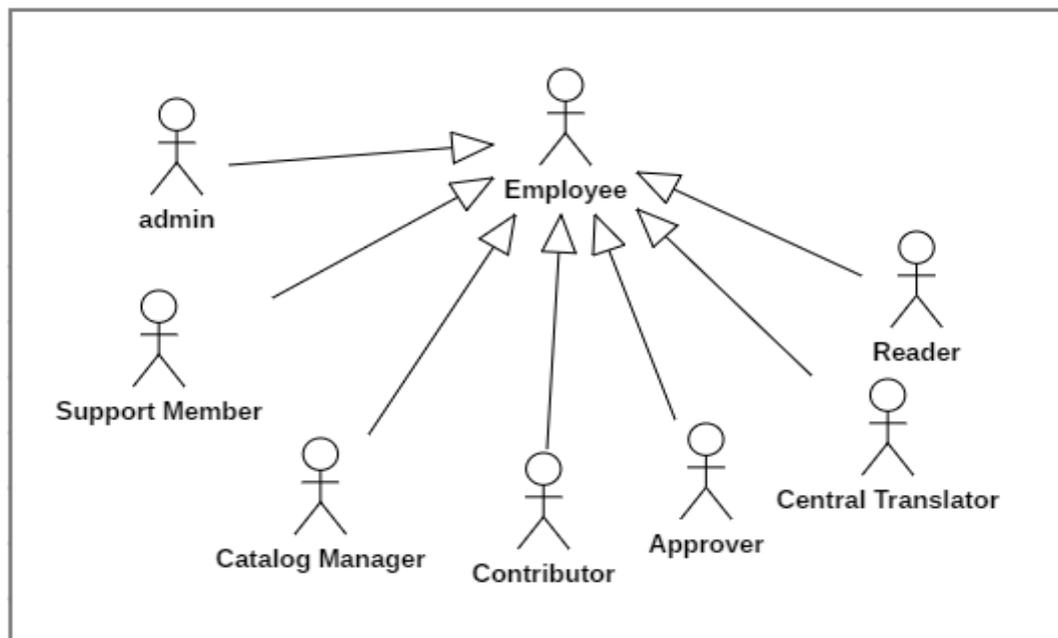


FIGURE 2.1 – Héirarchie des profils humaines.

Le tableau 2.1 récapitule les acteurs en interaction avec le système en spécifiant le rôle de chacun avant de définir plus précisément leurs interactions avec le système en utilisant des diagrammes de cas d'utilisation.

| Acteur | Fonction |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| L'admin | Administrateur technique qui a accès à toutes les fonctionnalités du PIM. C'est un profil du groupe "AdminGroup Hybris". |
| Le Catalog Manager | Gère le catalogue (produits, catégories, classification...). Il peut, par exemple : Créer une nouvelle catégorie, une nouvelle classification. Par contre, il ne peut pas créer un nouveau produit. |
| Le Contributor | Profil contributeur qui enrichit les produits du catalogue. Il ne gère pas la validation des fiches produits après enrichissement. Donc Il n'a pas le droit de changer le statut d'approbation du produit. |
| L'Approver | Profil contributeur et approbateur qui approuve le bon enrichissement du produit. Il valide la fiche produit et passe son statut à "Approuvée". |
| Le Central Translator | S'occupe de toutes les traductions nécessaires du catalogue (produit attributs, classification, catégorie. . .) |
| Le Reader | Un profil lecteur qui a accès uniquement au catalogue en lecture seule. |
| Le Support Member | Gère le support du PIM, C'est un profile qui peut créer des nouveaux comptes et des nouvelles catégories de classification. |
| L'EAI Microsoft BizTalk | Lance l'import automatisé des produits ainsi que des médias. |

TABLE 2.1 – Acteurs en interaction avec le système

2.2.2 Diagrammes des cas d'utilisation

Cette partie présente l'ensemble des cas d'utilisation de la plate-forme PIM en se basant sur les diagrammes des cas d'utilisation du système et des tableaux qui décrivent les scénarios nominaux. Les cas d'utilisation sont présentés dans ce qui suit suivant les acteurs du système.

Le profil Admin :

Les présents cas d'utilisation concernent l'acteur Administrateur (à part celui de l'authentification). Une fois authentifié, l'administrateur a la possibilité de gérer les utilisateurs et leurs droits, gérer le catalogue, les catégories, les produits, lancer l'import ou l'export des produits et d'accéder à l'espace HAC (Hybris Administration Console) pour effectuer des tâches techniques sur la plat-forme. Tous les cas d'utilisation passent obligatoirement par une authentification (voir la figure 2.2) :

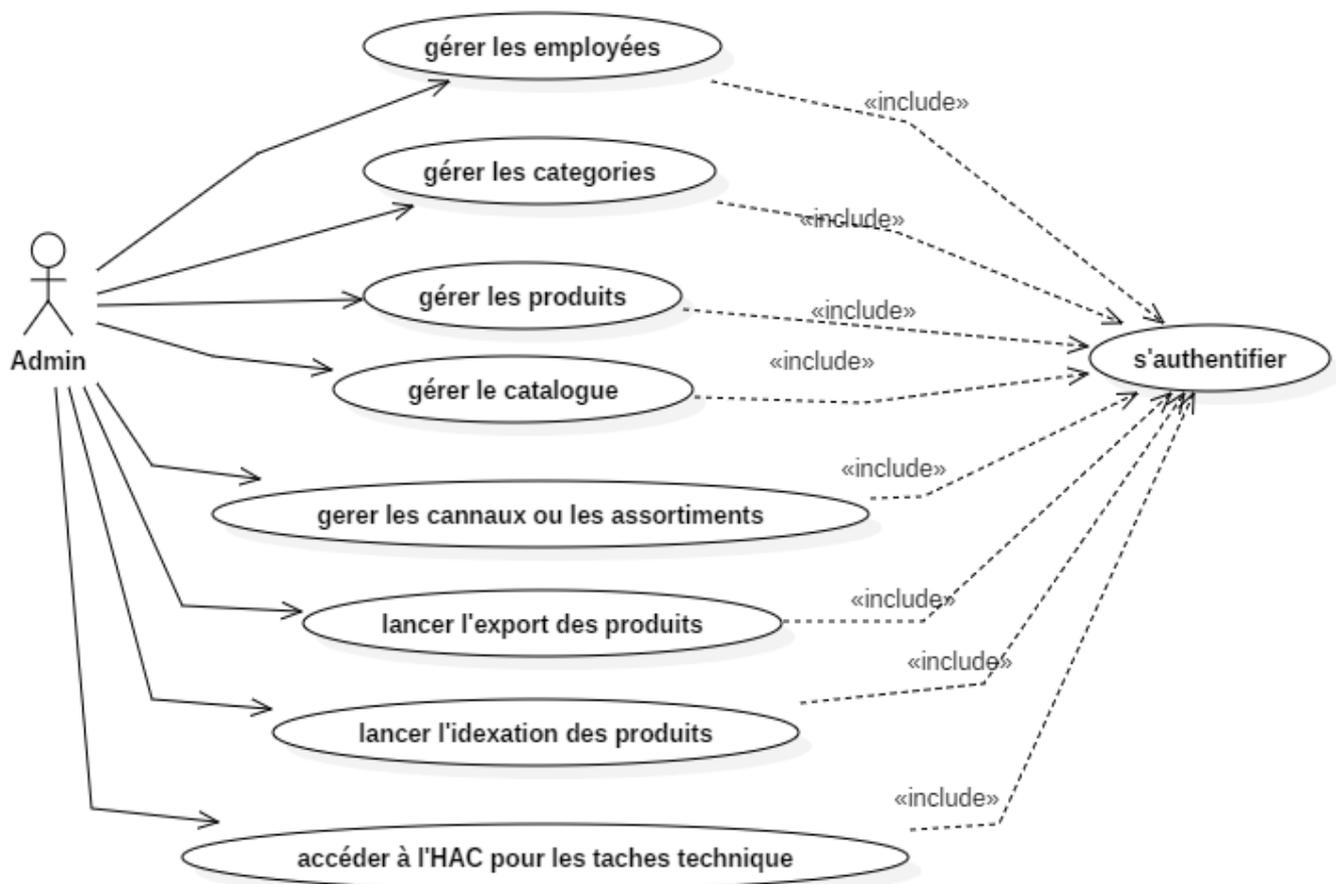


FIGURE 2.2 – Les cas d'utilisation du profil Admin.

- **Cas d'utilisation S'authentifier** : permet aux utilisateurs de se connecter au système avec leurs logins et mots de passe afin de sécuriser la plateforme.
 - **Objectif** : Cette fonctionnalité permet aux différents acteurs de se connecter.
 - **Acteur** : Tous les acteurs
 - **Pré-condition** : L'acteur se connecte au système.
 - **Scénario nominal** :
 1. L'acteur saisit son login et son mot de passe.
 2. Le système vérifie les informations saisies.
 3. Le système trouve que les informations saisies sont valides.
 4. Le système vérifie le rôle de l'acteur.
 5. Le système connecte l'acteur à son espace.
 - **Scénario d'erreur** :
 1. L'acteur saisit son login et son mot de passe.
 2. Le système vérifie les informations saisies.
 3. Le système trouve que les informations saisies sont invalides.
 4. Le système demande à l'acteur de vérifier les informations saisies.

- **Cas d'utilisation Gérer les employés** : L'administrateur prend en charge la gestion des utilisateurs de la plat-forme :
 - **Objectif** : Cette fonctionnalité permet à l'administrateur de gérer les utilisateurs du PIM qui sont des employés du client.
 - **Acteur** : Administrateur
 - **Pré-condition** : L'acteur se connecte au système.
 - **Scénarios nominaux** :
 1. *Ajouter un utilisateur.*
 - ★ L'acteur saisit les informations de l'utilisateur.
 - ★ Le système confirme à l'administrateur l'enregistrement de l'utilisateur.
 - ★ Le système affiche la liste des utilisateurs contenant l'utilisateur ajouté.
 2. *Modifier un utilisateur.*
 - ★ L'acteur affiche la liste des utilisateurs.
 - ★ L'acteur sélectionne l'utilisateur à modifier.
 - ★ Le système affiche les informations de l'utilisateur sélectionné
 - ★ L'acteur modifie les champs concernés.
 - ★ L'acteur valide ses modifications.
 - ★ Le système confirme à l'acteur la mise à jour des informations.
 - ★ Le système affiche la liste des utilisateurs contenant l'utilisateur modifié.
 3. *Supprimer un utilisateur* :

- ★ L'acteur affiche la liste des utilisateurs.
- ★ L'acteur sélectionne l'utilisateur à supprimer.
- ★ Le système alerte l'acteur sur son action.
- ★ L'acteur valide son action.
- ★ Le système confirme à l'acteur la suppression de l'utilisateur.

4. *Chercher des utilisateurs :*

- ★ L'acteur affiche la liste des utilisateurs.
- ★ L'acteur clique sur le volet de recherche.
- ★ L'acteur choisit les critères de recherche.
- ★ L'acteur lance la recherche.
- ★ Le système affiche les résultats de recherche.

- **Cas d'utilisation Gérer les catégories :** L'administrateur peut gérer les catégories qui sert a classifier les produits selon leurs critères et typologies :

- **Objectif :** Cette fonctionnalité permet à l'administrateur de gérer les catégories des produits disponibles dans le PIM.
- **Acteur :** Administrateur
- **Pré-condition :** L'acteur se connecte au système.
- **Scénarios nominaux :**

1. *Ajouter une catégorie.*

- ★ L'acteur saisit les informations de la catégorie.
- ★ Le système confirme à l'administrateur l'enregistrement de la catégorie.
- ★ Le système affiche la liste des catégories contenant la catégorie ajoutée.

2. *Modifier une catégorie.*

- ★ L'acteur affiche la liste des catégories.
- ★ L'acteur sélectionne la catégorie à modifier.
- ★ Le système affiche les informations de la catégorie sélectionnée
- ★ L'acteur modifie les champs concernés.
- ★ L'acteur valide ses modifications.
- ★ Le système confirme à l'acteur la mise à jour des informations.

3. *Supprimer une catégorie :*

- ★ L'acteur affiche la liste des catégories.
- ★ L'acteur sélectionne la catégorie à supprimer.
- ★ Le système alerte l'acteur sur son action.
- ★ L'acteur valide son action.
- ★ Le système confirme à l'acteur la suppression de la catégorie.

4. *Affecter/désaffecter des produits à une catégorie :*

- ★ L'acteur affiche la liste des catégories.
- ★ L'acteur sélectionne la catégorie souhaitée.

- ★ Le système affiche les informations de la catégorie sélectionnée
 - ★ L'acteur ouvre l'onglet des produits inclut dans la catégorie.
 - ★ L'acteur ajoute/supprime des produits à/de la catégorie.
 - ★ L'acteur valide ses modifications.
 - ★ Le système confirme à l'acteur la mise à jour des informations.
5. *Chercher une catégorie :*
- ★ L'acteur affiche la liste des catégories.
 - ★ L'acteur sélectionne ouvre le voler de recherche.
 - ★ L'utilisateur choisit les critères de recherche.
 - ★ L'acteur lance la recherche.
 - ★ Le système affiche la liste des catégories conformes à la recherche.
- **Cas d'utilisation Gérer les produits :** Cette fonctionnalité permet à l'administrateur de lister, modifier ou chercher les produits disponible dans le PIM. l'ajout et la suppression des produits ne sont pas incluses.
- **Objectif :** L'administrateur peut gérer les produits disponibles dans le PIM.
 - **Acteur :** Administrateur
 - **Pré-condition :** L'acteur se connecte au système.
 - **Scénarios nominaux :**
 1. *Modifier un produit.*
 - ★ L'acteur affiche la liste des produits.
 - ★ L'acteur sélectionne le produit à modifier.
 - ★ Le système affiche les informations du produit sélectionné
 - ★ L'acteur modifie les champs concernés.
 - ★ L'acteur valide ses modifications.
 - ★ Le système confirme à l'acteur la mise à jour des informations.
 2. *Chercher un produit :*
 - ★ L'acteur affiche la liste des produits.
 - ★ L'acteur ouvre le voler de recherche.
 - ★ L'acteur choisit les critères de recherche.
 - ★ L'acteur lance la recherche.
 - ★ Le système affiche la liste des produits conformes à la recherche.
 3. *Affecter/désaffecter un produit à une ou plusieurs catégories :*
 - ★ L'acteur affiche la liste des produits.
 - ★ L'acteur sélectionne le produit souhaité.
 - ★ Le système affiche les informations du produit sélectionné
 - ★ L'acteur ouvre l'onglet de catégorisation.
 - ★ L'acteur ajoute/supprime des catégories au/du produit.
 - ★ L'acteur valide ses modifications.

- ★ Le système confirme à l'acteur la mise à jour des informations.
- 4. *Établir des liens entre des produits* :
 - ★ L'acteur affiche la liste des produits.
 - ★ L'acteur sélectionne le produit souhaité.
 - ★ Le système affiche les informations du produit sélectionné
 - ★ L'acteur ouvre l'onglet des liens.
 - ★ L'acteur ajoute/supprime des liens au/du produit vers d'autres produits.
 - ★ L'acteur valide ses modifications.
 - ★ Le système confirme à l'acteur la mise à jour des informations.
- **Cas d'utilisation Gérer le catalogue** : Le catalogue constitue une sorte de conteneur global des produits et catégories disponibles dans le PIM. Il est divisé en deux version Staged (intermédiaire) et Online (validée ou publiée) pour distinguer les données en cours d'enrichissement des données publiées (utilisables pour être exportées). Le présent cas d'utilisation permet la synchronisation entre ces deux versions et l'indexation des objets de ce catalogue.
 - **Objectif** : L'administrateur peut gérer le catalogue des produits du PIM :
 - **Acteur** : Administrateur
 - **Pré-condition** : L'acteur se connecte au système.
 - **Scénarios nominaux** :
 1. *Lancer l'indexation du catalogue.*
 - ★ L'acteur affiche la liste des jobs disponibles.
 - ★ L'acteur choisit le job d'indexation.
 - ★ L'acteur lance le job d'indexation.
 - ★ Le système affiche l'état d'avancement du job.
 2. *Lancer la synchronisation des version du catalogue* :
 - ★ L'acteur affiche la liste des jobs disponibles.
 - ★ L'acteur sélectionne le job de synchronisation.
 - ★ L'acteur lance le job.
 - ★ L'acteur lance la recherche.
 - ★ Le système affiche l'état d'avancement du job.
- **Cas d'utilisation Gérer les canaux ou les assortiments** : Le canaux ou les assortiments constituent une sorte de catégories permettant de regrouper les produits par les endroits ou les canaux vers lesquels ils sont exportés.
 - **Objectif** : L'administrateur peut gérer les canaux ou les assortiments :
 - **Acteur** : Administrateur
 - **Pré-condition** : L'acteur se connecte au système.
 - **Scénarios nominaux** : les scénarios de ce cas sont les mêmes que pour le cas d'utilisation de gestion des catégories.

- **Cas d'utilisation lancer l'export des produits :**

- *Objectif* : L'administrateur peut exporter des produits avec différents formats.
- *Acteur* : Administrateur
- *Pré-condition* : L'acteur se connecte au système.
- *Scénarios nominaux* :

1. *Exporter tous les produits.*

- ★ L'acteur affiche la liste des produits disponibles.
- ★ L'acteur choisit le format d'export.
- ★ le système affiche la pop-in de l'export et demande à l'utilisateur de renseigner les champs et les langues à exporter.
- ★ l'acteurs renseigne les champs et les langues et lance l'export.
- ★ le système exporte les produits.

2. *Exporter partiellement les produits*

- ★ L'acteur affiche la liste des produits disponibles.
- ★ L'acteur ouvre le volet de recherche et choisit les critères souhaités.
- ★ L'acteur lance la recherche.
- ★ le système affiche les résultats de recherche.
- ★ L'acteur choisit le format d'export.
- ★ le système affiche la pop-in de l'export et demande à l'utilisateur de renseigner les champs et les langues à exporter.
- ★ l'acteurs renseigne les champs et les langues et lance l'export.
- ★ le système exporte les produits correspondants au critères de recherche.

- **Cas d'utilisation se connecter à l'HAC (Hybris Administration Console) :**

Ce cas d'utilisation permet à l'administrateur d'effectuer des tâches purement techniques comme l'interrogation de la base de données avec des requêtes SQL ou avec le langage Flexible Search de Hybris et d'exécuter des scripts Groovy en runtime ainsi que d'importer des données avec les Impex de Hybris et bien plus d'autres fonctionnalités[5].

Le profil Catalog Manager :

Les présents cas d'utilisation concernent l'acteur Catalog Manager. Après avoir authentifié le Catalog Manager peut de gérer les catégories et les produits ainsi que les assortiments et les canaux (voir la figure 2.3).

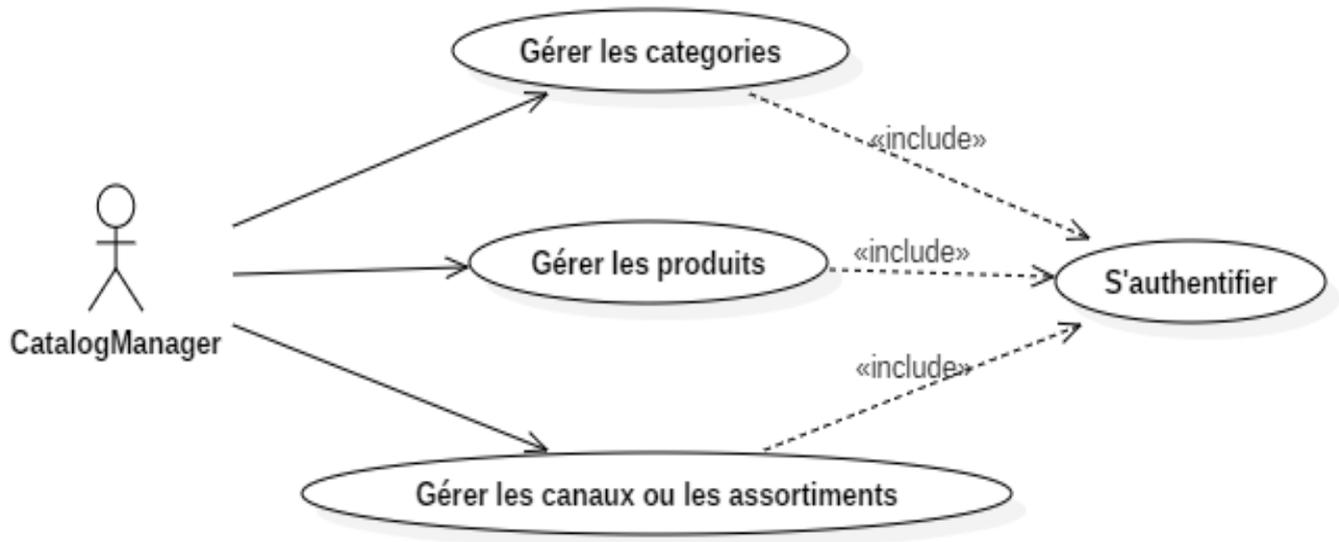


FIGURE 2.3 – Les cas d'utilisation du profil Catalog Manager.

Les trois cas d'utilisation sont semblables à celles décrites pour le profil Admin.

Le profil Support Member :

Les présents cas d'utilisation concernent l'acteur Support Member. Un profil Support qui gère le support du PIM, il peut donc créer des nouveaux comptes, nouvelles catégories. Il a le droit de naviguer en lecture seule dans les produits et il ne peut rien modifier ou supprimer. Tous les cas d'utilisation passent obligatoirement par une authentification (voir la figure 2.4).

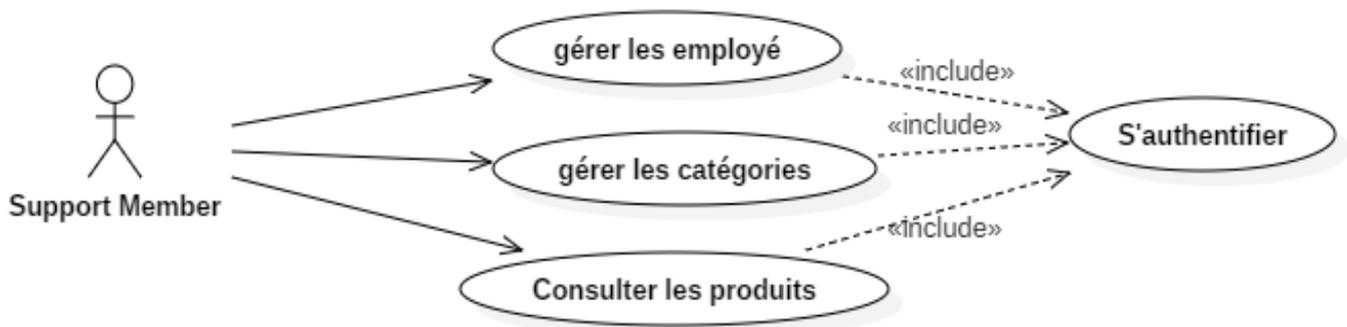


FIGURE 2.4 – Les cas d'utilisation du profil Support Member.

- **Cas d'utilisation Consulter des produits** : Cette fonctionnalité permet au Support Member de lister ou chercher des produits disponibles dans le PIM pour les consulter.
 - **Objectif** : Le Support Member peut consulter les produits disponibles dans le PIM.
 - **Acteur** : Support Member
 - **Pré-condition** : L'acteur se connecte au système.
 - **Scénarios nominaux** :
 1. *Consulter un produit.*
 - ★ L'acteur affiche la liste des produits.
 - ★ L'acteur sélectionne le produit à modifier.
 - ★ Le système affiche les informations du produit sélectionné
 - ★ L'acteur modifie les champs concernés.
 - ★ L'acteur valide ses modifications.
 - ★ Le système confirme à l'acteur la mise à jour des informations.
 2. *Chercher un produit* :
 - ★ L'acteur affiche la liste des produits.
 - ★ L'acteur ouvre le volet de recherche.
 - ★ L'acteur choisit les critères de recherche.
 - ★ L'acteur lance la recherche.
 - ★ Le système affiche la liste des produits conformes à la recherche.
- **Cas d'utilisation gérer les utilisateurs** : Cette fonctionnalité est la même décrite pour le profil Admin.
- **Cas d'utilisation gérer les catégories** : Cette fonctionnalité est la même décrite pour le profil Admin.

Le profil Contributor :

Les présents cas d'utilisation concernent l'acteur Contributor. Un profil qui a comme mission l'enrichissement des informations des produits et catégories. Tous les cas d'utilisation passent obligatoirement par une authentification.

Les quatre cas d'utilisation sont similaires à celles décrites pour le profil Admin (voir la figure 2.5).

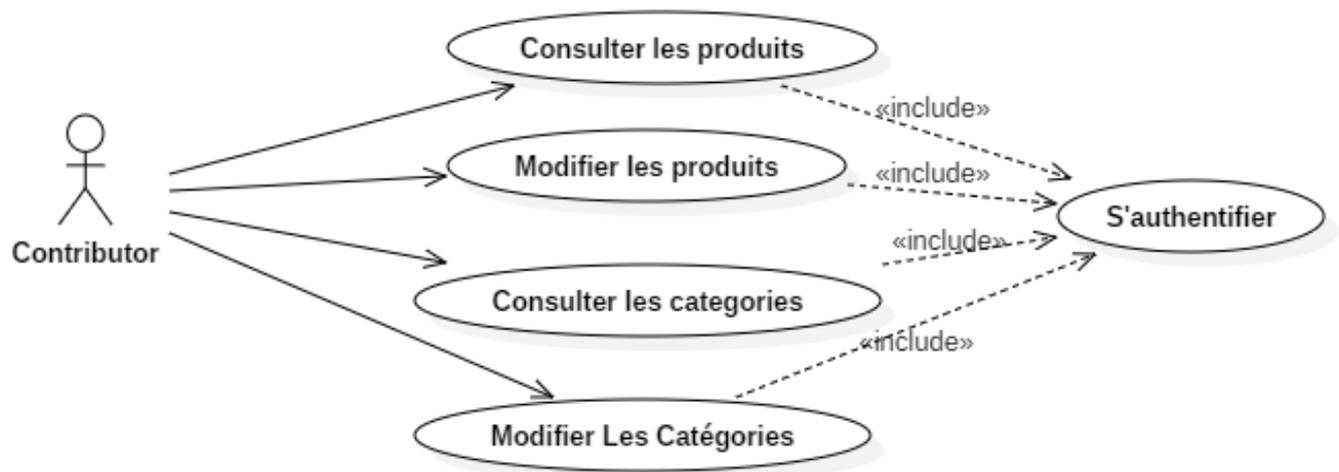


FIGURE 2.5 – Les cas d'utilisation du profil Contributor.

Le profil Central Translator :

Les présents cas d'utilisation concernent l'acteur Central Translator qui a pour mission la traduction des champs localisés (présentés dans plusieurs langues) des produits et des catégories (voir la figure 2.6).

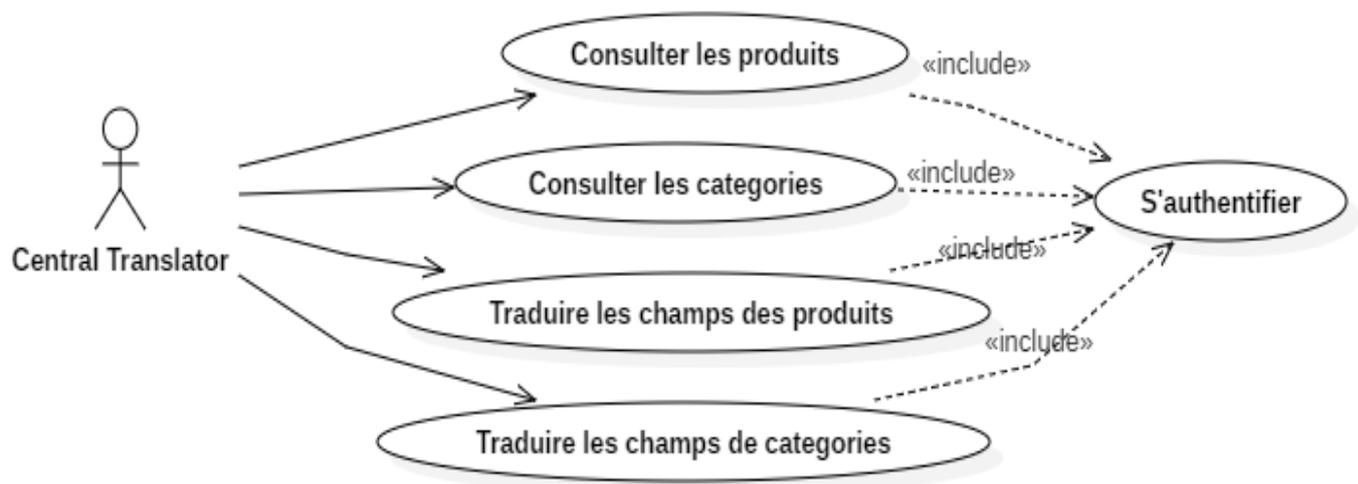


FIGURE 2.6 – Les cas d'utilisation du profil Centrale Translator.

- **Cas d'utilisation Consulter des produits** : Cette fonctionnalité est la même décrite pour le profil Support Member.
- **Cas d'utilisation Consulter les catégories** : Cette fonctionnalité suit la même

procédure que pour les produits.

- **Cas d'utilisation traduire les champs des produits :**

- **Objectif** : Le Central Translator peut traduire les informations des produits disponibles dans le PIM.
- **Acteur** : Central Translator.
- **Pré-condition** : L'acteur se connecte au système.
- **Scénario nominal** :
 1. L'acteur affiche les informations du produit.
 2. L'acteur remplit les champs localisés.
 3. L'acteur valide ses modifications.
 4. Le système confirme à l'acteur la mise à jour des informations.

- **Cas d'utilisation traduire les champs des produits :**

- **Objectif** : Le Central Translator peut traduire les informations des catégories disponibles dans le PIM.
- **Acteur** : Central Translator.
- **Pré-condition** : L'acteur se connecte au système.
- **Scénario nominal** :
 1. L'acteur affiche les informations de la catégorie.
 2. L'acteur remplit les champs localisées.
 3. L'acteur valide ses modifications.
 4. Le système confirme à l'acteur la mise à jour des informations.

Le profil Approver :

Les présents cas d'utilisation concernent l'acteur Approver qui a pour mission l'approbation des modification importé sur les produits du PIM (voir la figure 2.7).

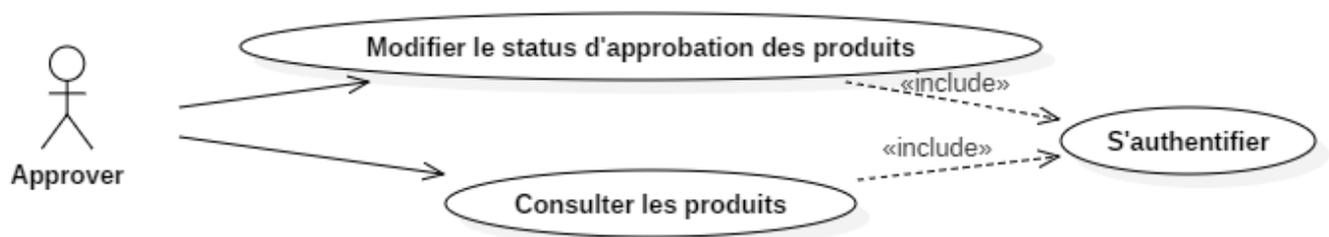


FIGURE 2.7 – Les cas d'utilisation du profil Approver.

- **Cas d'utilisation Consulter des produits** : Cette fonctionnalité est la même décrite précédemment pour le profil Support Member.
- **Cas d'utilisation changer l'état d'approbation des produits** :
 - *Objectif* : L'Approver peut modifier l'état d'approbation des produits disponibles dans le PIM.
 - *Acteur* : Approver.
 - *Pré-condition* : L'acteur se connecte au système.
 - *Scénario nominal* :
 1. L'acteur affiche les informations du produit.
 2. L'acteur change l'état d'approbation du produit (Initial, In Progress ou Approved).
 3. L'acteur valide ses modifications.
 4. Le système confirme à l'acteur la mise à jour de l'état d'approbation.

Le profile Reader :

Les présents cas d'utilisation concernent l'acteur Reader qu'a que le droit en lecture seuls sur tous les objets du PIM (voir la figure 2.8).

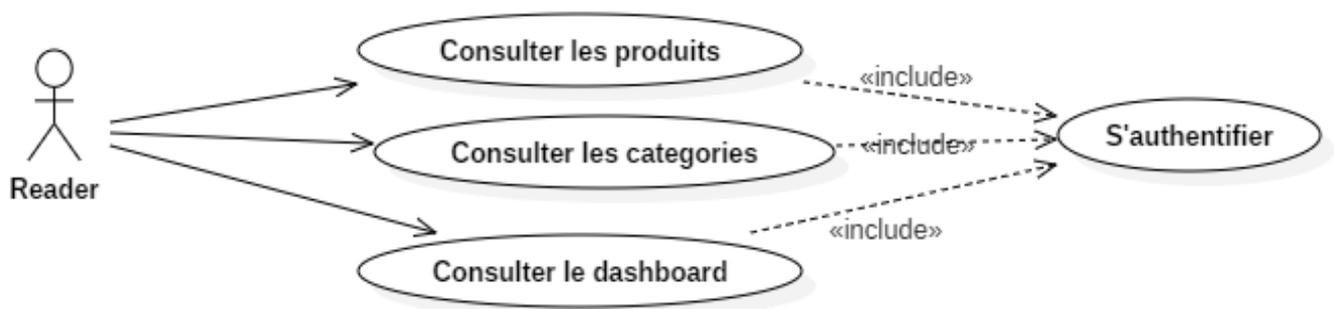


FIGURE 2.8 – Les cas d'utilisation du profile Reader.

- **Cas d'utilisation Consulter des produits** : Cette fonctionnalité est la même décrite précédemment pour le profile Support Member.
- **Cas d'utilisation Consulter les catégories** : Cette fonctionnalité suit la même procédure que pour les produits.
- **Cas d'utilisation Consulter le dashboard** :
 - *Objectif* : Le Reader peut voir les reportings de la plat-forme.

- *Acteur* : Reader.
- *Pré-condition* : L'acteur se connecte au système.
- *Scénario nominal* :
 1. L'acteur sélectionne la vue dashboard.
 2. le systeme affiche les graphes et statistiques du PIM.

Les cas d'utilisation d'import des produits et des médias :

Les présents cas d'utilisation sont effectués par un système tiers qui est le EAI (Entreprise Application Integration) Microsoft BizTalk responsable du transport des produits de l'ERP du client vers le PIM (voir la figure 2.9).

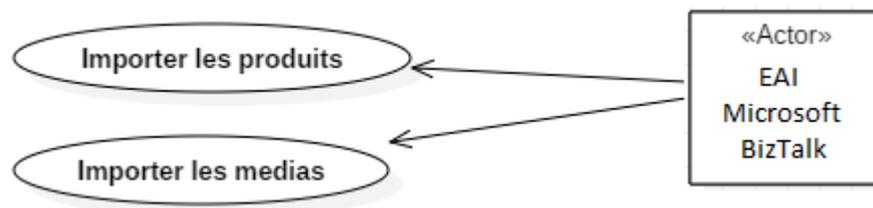


FIGURE 2.9 – Les cas d'utilisation d'import des produits et des médias.

• Cas d'utilisation importer les produits :

- *Objectif* : Le EAI Biztalk peut importer les produits.
- *Acteur* : EAI Biztalk.
- *Pré-condition* : les fichiers XML des produits à importer respectent une règle de nommage prédéfini.
- *Scénario nominal* :
 1. L'acteur positionne les fichiers XML des produits dans le dossier d'import.
 2. Le système importe automatiquement les produits.

• Cas d'utilisation importer les médias :

- *Objectif* : Le EAI Biztalk peut importer les médias.
- *Acteur* : EAI Biztalk.
- *Pré-condition* : les médias à importer respectent une règle de nommage prédéfini.
- *Scénario nominal* :
 1. L'acteur positionnes les fichiers des médias dans le dossier d'import.
 2. Le système importe automatiquement les médias.

2.2.3 Conclusion

Le but de ce chapitre était de définir et d'analyser l'ensemble des besoins fonctionnels de notre solution. Cette étape est primordiale dans le développement d'un projet informatique puisque elle nous permet de définir le périmètre fonctionnel du projet, et de garantir la couverture de l'ensemble des fonctionnalités recensées.

Chapitre 3

Étude Conceptuel et architectural

“ Dans ce chapitre nous allons nous concentrer sur l’aspect conceptuel et architectural de la solution à mettre en place. Cette étude conceptuelle est basée sur l’élaboration et l’analyse des différents schémas et diagrammes de classes. ”

3.1 Introduction

L'organisation de l'entreprise cliente repose sur le maillage de trois divisions : la Mode (Fashion), les Parfums Beauté (fragrance and beauty) et l'Horlogerie Joaillerie (Watches And Fine Jewelry). La plate-forme Hybris est potentiellement utilisée pour les 3 divisions.

Pour ce qui concerne la solution PIM, elle est principalement implémentée pour les deux divisions de Mode et d'Horlogerie Joaillerie. Pour cela, une partie de la modélisation de cette solution est partagée entre les deux solutions, et ce pour remédier à la répétition et mettre en valeur les éléments communs entre les deux divisions.

Le présent projet porte sur la solution PIM de la division Horlogerie Joaillerie. Pour des raisons de lisibilité, on préfixera dans le reste de ce chapitre les entités et notions propre à cette division avec le préfixe WFJ (Watches and Fine Jewelry) et pour les entités et notions communes entre les deux divisions avec le préfixe PIM.

3.2 Modèle de données

3.2.1 Catalogue et versions

Hybris est une solution logicielle centrée sur le principe de catalogue de produits afin de gérer les données des produits, connu aussi sous le nom de PCM (Product Content Management). Ce catalogue est composé de plusieurs versions et regroupe l'ensemble des produits avec leurs données relatives (voir la figure 3.1).



FIGURE 3.1 – Aperçu du diagramme de classes du catalogue et ses versions.

Notre solution PIM s'appuie sur ce principe. Par la suite, on utilise les deux versions de catalogue **Staged** et **Online** pour distinguer les données en cours d'enrichissement des données publiées (utilisables pour être exportées).

La réplication des données d'une version du catalogue à l'autre est effectué à partir d'un traitement de **Synchronisation** natif à hybris, qui est à son tour possible à travers le principe du **Catalog Aware** d'Hybris.

Ce processus est un job d'Hybris qui sélectionne les produits et les autres éléments du système qui sont **Catalog Aware** de la version intermédiaire ou **Staged**, vérifie la

présence de certaines règles (indiquant que l'élément est prêt à être mis en ligne ou publié) et crée (ou la met à jour s'il est déjà créé) une copie de l'élément avec la version En ligne ou **Online**.

Le principe du **Catalog Aware** sert à indiquer les produits ou les entités devraient avoir deux copies dans le système, afin que les utilisateurs professionnels puissent apporter les modifications appropriées avant leur publication. Ces éléments sont appelés **Catalog Aware** et entrent dans le processus de synchronisation. Or les autres éléments du système qui ne participent pas ou ne devraient pas participer au processus de synchronisation sont appelés **Catalog Unaware**.

La figure 3.2 illustre le principe du catalogue et des versions de catalogue d'Hybris, implémenté dans la plate-forme PIM sous le nom de **WFJProductCatalog** :

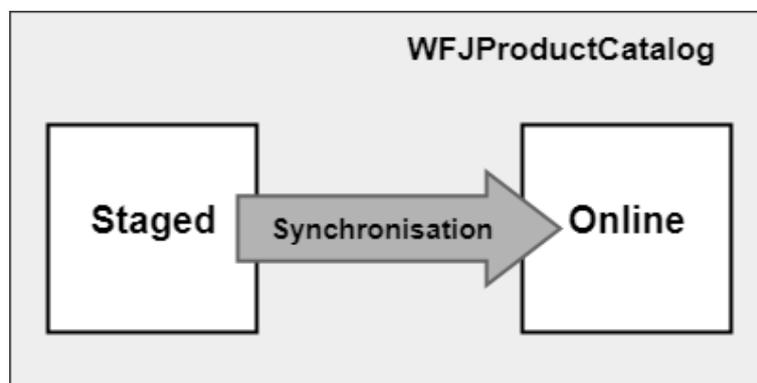


FIGURE 3.2 – Le principe du catalogue des produits d'Hybris.

On suit donc les règles de gestion suivantes pour adapter ce système aux besoins de notre solution PIM :

- L'import des données se fait sur la version Staged.
- L'enrichissement des données se fait sur la version Staged.
- Lorsqu'un produit est correctement enrichi, il est manuellement passé à l'état "approved".
- L'export des données se fait à partir de la version Online.
- Le système de synchronisation recopie les données de la version Staged vers la version Online pour tous les produits qui sont à l'état "approved".
- Si un produit change d'état d'approbation sur la version Staged
 - Il n'est plus synchronisé sur la version Online.
 - Mais il reste présent sur la version Online (avec les données de la dernière synchronisation).

On met en place un traitement (cronjob) de synchronisation, basé sur le fonctionnement natif hybris, permettant de répliquer les données du catalogue (produits avec leurs attributs, catégories, etc.) de la version **Staged** vers la version **Online**. Les entités qui sont synchronisées sont :

- Produits (et toutes les déclinaisons)
- Catégories (et toutes les déclinaisons)
- Médias (et toutes les données associées)

Les processus d'import, d'enrichissement et d'export se résument dans la figure 3.3 :

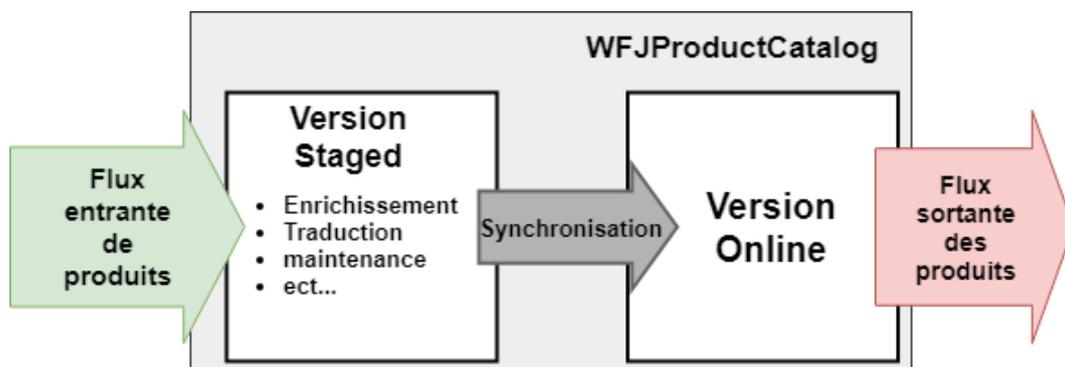


FIGURE 3.3 – Récapitulatif du processus de Synchronisation.

3.2.2 Les produits

Tous les produits de la division Horlogerie Joaillerie ou WFJ (Watches And Fine Jewelry) sont gérés à partir de l'ERP central, nommé CERP et basé sur la solution technique Microsoft AX. Indépendamment de leurs typologies, les produits WFJ sont déclinés suivant 2 à 3 niveaux de variante : **Référence** (entité produit de base qui contient la plupart des attributs de base du produit) déclinées en **SKU** (variation du produit de base selon un certain attribut comme la taille ou la couleur) et éventuellement **Matricule** (pour les pièces numérotées).

Le schéma de la figure 3.4 illustre cette hiérarchie des produits :

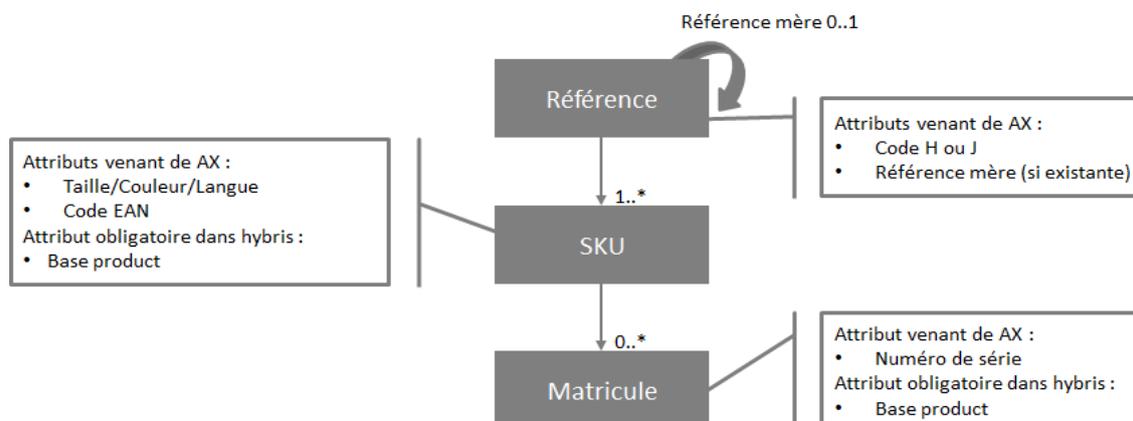


FIGURE 3.4 – Récapitulatif de l'abstraction du produit.

Les Références ou les produits de base :

La **Référence** est équivalente au **Product** dans le modèle de données natif d'Hybris. On définit donc la référence dans notre solution avec l'entité **WfjProduct**, en héritant de ce modèle.

Chaque référence est unique et possède un code (unique) défini dans l'ERP est utilisé telquel dans le PIM.

- Dans AX, l'horlogerie et la joaillerie ont chacune une base de données distincte même si certaines données référentielles sont communes aux deux.
- Les références de joaillerie ont un code commençant par un **J**.
- Les références d'horlogerie ont un code commençant par un **H**.
- Une référence peut avoir une référence mère ; nommée masterProduct coté PIM.
- Les références sont identifiables par leur **code (H ou J)**, il s'agit de leur clé d'unicité.

Pour définir les variations d'une certaine **Référence** on a introduit la notion de **Grille de Tailles**, modélisée par l'entité **WfjErpSizeGrid**. Cette entité est liée à la référence (WfjProduct) et porte les différents valeurs possibles de la **Taille** modélisée par l'entité **WfjErpSize** et représentant le paramètre de variation de cette référence.

Cette notion de grille de taille est détournée pour représenter non seulement les variations de vraies tailles (S, M, X, XL, ...) mais aussi des variations de couleurs ou de langues (pour les catalogues imprimés).

Les SKUs ou les variants du produit selon la taille :

Le **SKU** est la variante de produit basée sur l'attribut de taille en fonction de la grille de la taille renseignée sur la **Référence**. Les **SKUs** sont présentés dans le PIM par l'entité

WfjSkuVariant.

- Un SKU est lié à une et une seule Référence via l'attribut *"baseProduct"* et une référence peut avoir plusieurs SKU.
- la notion de taille est légèrement détournée dans AX pour certains types de produits. En effet, certaines grilles "de taille" sont déclarées pour gérer des déclinaison de couleur ou de vraies tailles.
- Les SKU sont identifiables par leur **code** de référence de rattachement et leur **taille** (ou leur position dans leur grille de taille), les deux forment la clé d'unicité.

Les Matricules ou les pièces numérotées :

Les **Matricules** sont les produits réels dans le cas de pièce numérotées. Chaque matricule représente une version unique d'un SKU et possède donc un numéro de série unique (généralisé par AX). Les Matricules sont présentés dans le PIM par l'entité **WfjSerialNumberVariant**.

- Un matricule est lié à un et un seul SKU via l'attribut *"baseProduct"*. Un SKU peut avoir plusieurs matricules.
- Les matricules sont identifiables par (autrement dit, la clé d'unicité est le triplet) :
 - Leur **code** de Référence de rattachement.
 - Leur **taille** (ou leur position dans leur grille de taille).
 - Leur **numéro de série**.

Le diagramme de classes de la figure 3.5 présente les différentes classes utilisés pour représenter le modèle du produit. Il met en évidence avec trois couleurs les classes qui sont natives à Hybris (en bleu) et les classes qui sont communes avec les autres divisions où est implémenté le PIM (en orange et préfixées par PIM) et celles conçues spécifiquement pour les produits de la division WFJ (en rose et préfixées par WFJ).

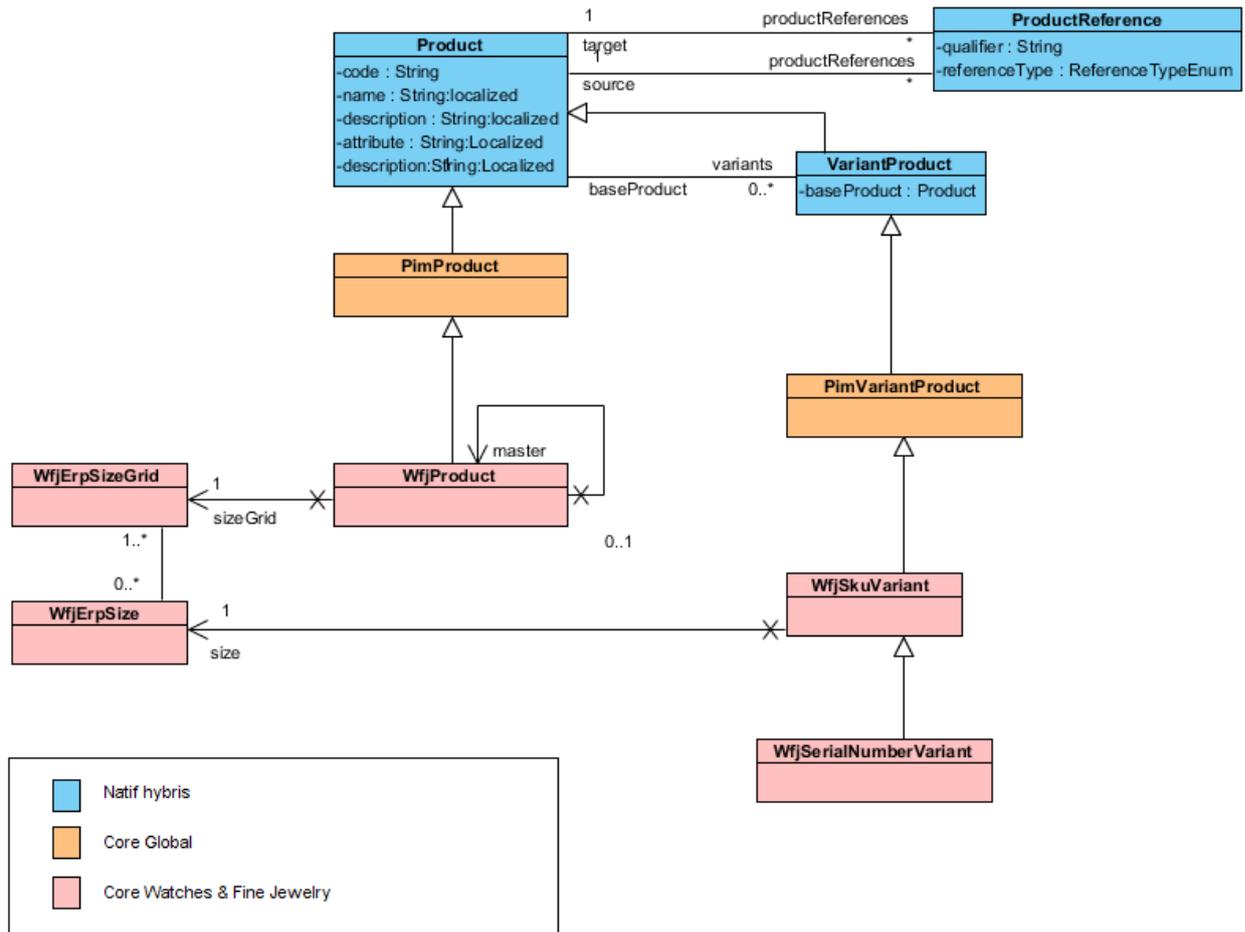


FIGURE 3.5 – Diagramme de classes des produits.

Comme le montre la figure 3.5, les différentes entités présentées dans ce diagramme sont comme suit :

- **WfjProduct**, hérite de **PimProduct** qui hérite à son tour, de L'entité **Product** d'Hybris est constitue la **Référence** présenté précédement.
- **WfjProduct** peut avoir un master product du même type permettant de mettre en oeuvre des systèmes d'héritage des données, c'est à dire de déduire certaines informations d'une référence à partir de sa référence master.
- **ProductReference** permet d'établir un lien entre de produit en spécifiant le type de ce lien et les deux produits source et target.
- **WfjProduct** porte la notion de "*SizeGrid*" (grille de déclinaisons de taille).
- **WfjErpSizeGrid** Les grilles de tailles proviennent d'AX. Le concept de grille de taille est étendu (détourné) pour pouvoir gérer des tailles mais aussi des langues ou

des couleurs.

- **WfjSkuVariant**, représente le **SKU** introduit précédemment et hérite de **PimVariantProduct** qui, à son tour, hérite du **VariantProduct** du modèle natif d'Hybris, représentant des variations du produit en question comme par exemple au niveau de la taille (S, M, L, XL...) ou la couleur.
- **WfjErpSize**, La taille indique l'information liée à la grille des tailles de la référence.
 - Si la référence est liée à la grille des tailles "Taille", alors l'id sera une taille.
 - Si la référence est liée à la grille des tailles "Couleur", alors l'id sera une couleur et la taille du produit sera unique.
- **WfjSerialNumber**, représente le **Matricule** introduit précédemment et hérite de **WfjSkuVariant**, il représente les produits réels dans le cas de pièce numérotées. Chaque matricule représente une version unique d'un SKU et possède donc un numéro de série unique (généralisé par AX).

On suit ainsi les règles de gestion suivantes pour les références et les grilles de taille :

- On affecte toujours une et une seule grille de taille à une référence (**WfjProduct**) et une référence doit toujours posséder une grille de taille (**WfjErpSizeGrid**).
- La grille de taille unique "*UNI*" permet de gérer les cas où il n'y a pas de variation de la référence.
- Les grilles de tailles sont importées dans le PIM à partir du flux AX.

3.2.3 Les catégories

La plate-forme PIM permet de regrouper les produits dans des catégories. Ces catégories peuvent être structurées en arborescences et on identifie trois grandes typologies de catégories (voir la figure 3.6) :

- **ERP Category** : catégories permettant de représenter l'arborescence de l'ERP. Ces catégories et leurs liaisons sont répliquées à partir de flux de données en provenance de l'ERP et ne sont pas susceptibles d'être modifiés dans le PIM.
- **Communication Category** : catégories permettant de constituer l'arborescence principale de communication, autrement dit, de créer une autre arborescence à partir de celle de l'ERP pour le modifier et l'évoluer au besoin.
- **Tags Category** : catégories moins structurées, permettant d'organiser les produits de manière plus libre et avec d'éventuelles redondances.

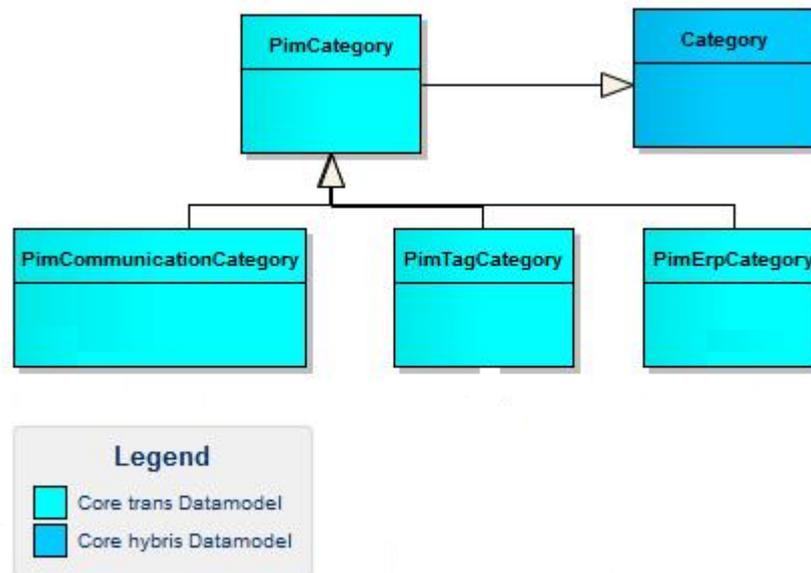


FIGURE 3.6 – Diagramme de classe des catégories.

3.2.4 Les medias

Le DAM (Digital Asset Management) permet de gérer des Assets; c'est à dire des ressources numériques (principalement des images, scan PDF et vidéos) visant à être présentées sur des canaux digitaux ou autres supports papiers [6].

- Les Assets du **DAM** sont matérialisées à partir de l'entité native d'Hybris : **MediaContainer**.
- Tout ou partie des **métadonnées** d'un Asset sont stockées dans des attributs du **MediaContainer** correspondant.
- Les formats de déclinaison des Assets du DAM sont configurés en tant que **MediaFormat** hybris (entité native de formats des médias).
- Les différentes déclinaisons, dans différents formats, d'un même Asset du DAM sont chacune matérialisées par un **Media** dans hybris. Chaque Media concerne un format donné.
- Tous les Medias sont regroupés au sein du **MediaContainer** représentant l'Asset.

Le diagramme de classes de la figure 3.7 représente ces différents classes et leurs relations :

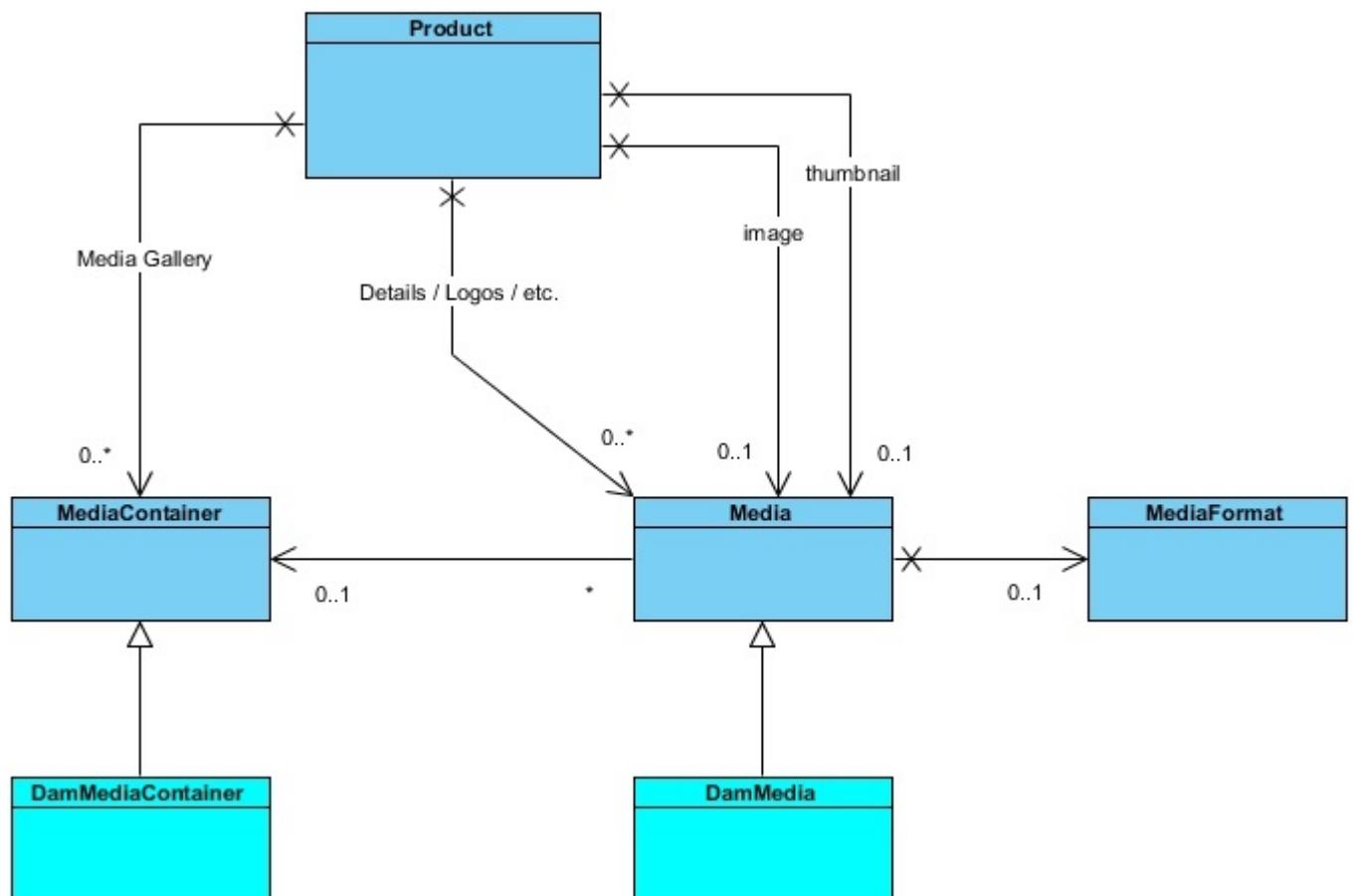


FIGURE 3.7 – Diagramme de classe des médias.

3.3 Le flux d'import

Le socle d'import permet d'importer des données depuis des systèmes tiers qui sont l'ERP centrale et le DAM du client vers le PIM. Il permet, par exemple, d'importer des produits ou des données référentielles ou encore des prix de produits.

Le socle d'import consomme des fichiers XML de données et plus précisément des fichiers au format CBL. En règle générale, les fichiers à importer dans le PIM sont mis à disposition par l'EAI Biztalk (qui prend donc en charge le transport des fichiers importés).

Le socle d'import fonctionne suivant un principe dit de "hotFolder" ; c'est à dire que les fichiers déposés dans les dossiers d'entrée (Splitting ou In) sont automatiquement consommés et traités dès lors qu'ils respectent une règle de nommage prédéfinie.

Le processus d'import comporte 3 principaux traitements comme le montre la figure 3.8 :

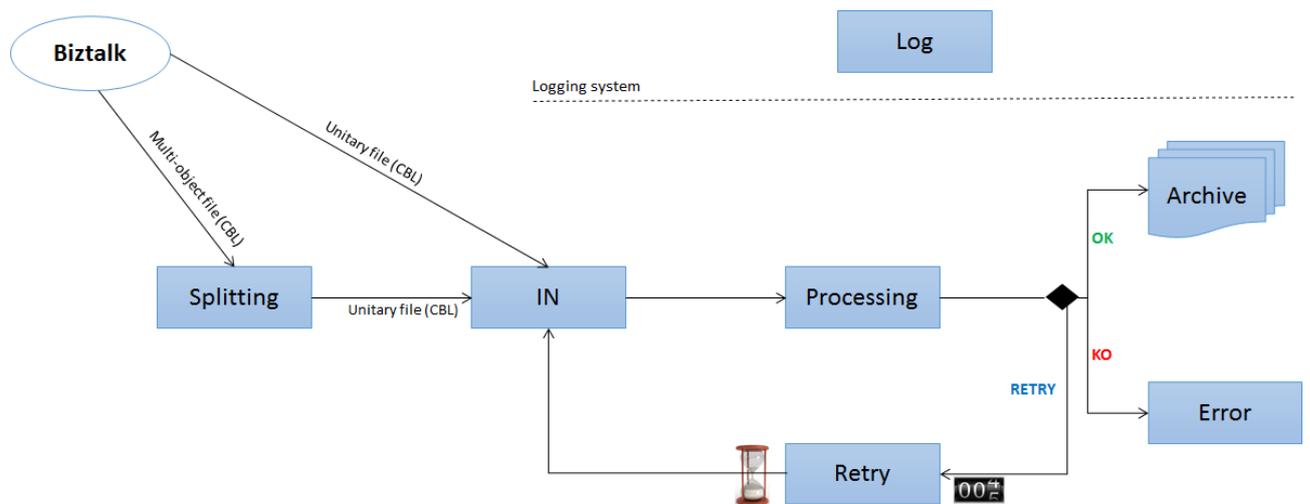


FIGURE 3.8 – Processus d'import des produits.

- **Splitting** : gère le découpage d'un fichier XML au format CBL¹ de collection d'objets en de multiples fichiers unitaires
 - L'opération de découpage n'est pas systématique. Pour la plupart des flux de données, Biztalk livre des fichiers unitaires qu'il dépose directement dans le dossier d'import.
- **Import** : étape d'injection des données d'un fichier unitaire vers un objet dans le PIM.
- **Retry** : traitement de relances multiples de l'import de fichiers unitaires ayant rencontré une erreur lors de l'import.

Une operation de **logging** ou journalisation et mise en place tout au long du processus d'import pour garder la traçabilité des différentes étapes et donc d'identifier les sources des éventuelles erreurs.

Traitement "splitting" :

Le Traitement de **splitting** est illustré dans la figure 3.9 :

1. Common Business Library : un format des fichiers XML utilisé dans l'e-commerce[7]

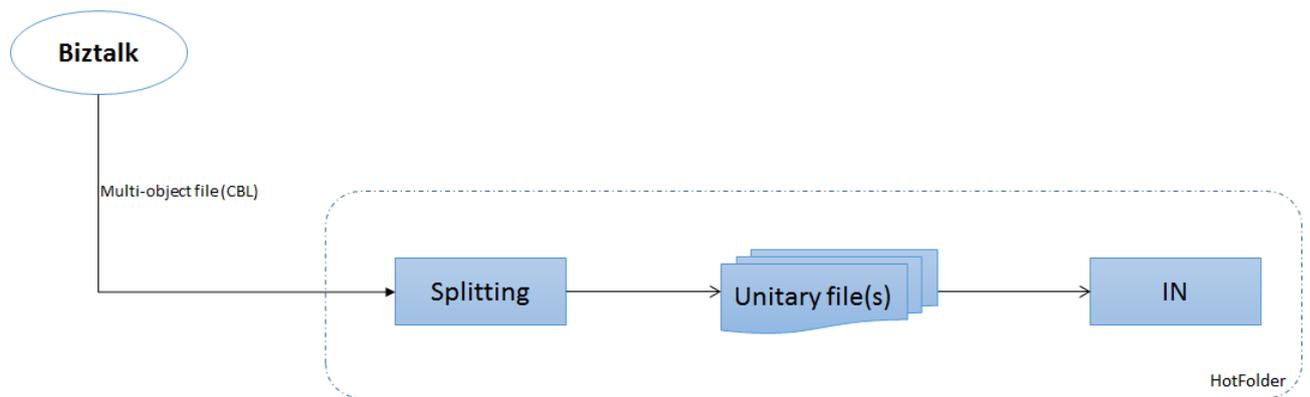


FIGURE 3.9 – Processus de splitting des fichiers xml.

Le traitement permet de découper des fichiers CBL contenant plusieurs objets de données (produits) en de multiples fichiers unitaires (contenant respectivement les données d'un seul produit) voir la figure 3.10.

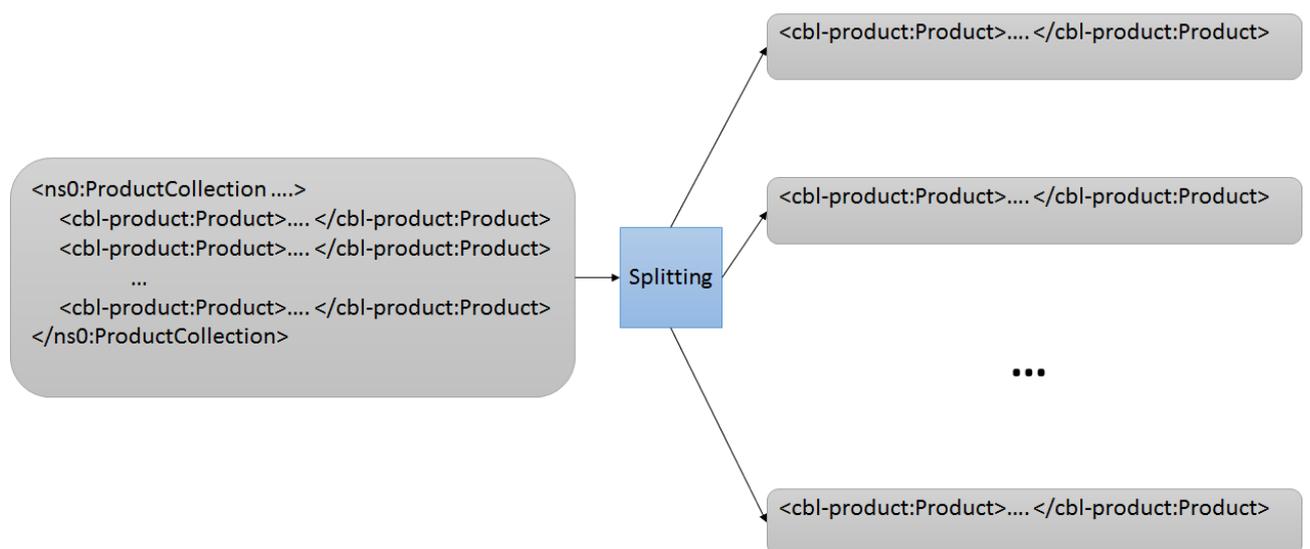


FIGURE 3.10 – Splitting d'un fichier multiple vers des fichiers unitaires.

- Le dossier d'entrée du traitement de découpage est le dossier "splitting".
- Le traitement est en mode "**hotFolder**".
- Le traitement lit le contenu du fichier initial (à découper) et génère à la volée (au fût et à mesure) les fichiers unitaires.
- Chaque fichier généré est automatiquement déposé dans le dossier d'import.
- Les fichiers unitaires suivent une règle de nommage particulière et intégrant l'identifiant de l'objet qu'ils représentent

Traitement "Import" :

Le traitement d'import d'un fichier de données commence par déplacer le fichier à importer dans un dossier de travail : *Processing*. Certains contrôles (dépendant de l'entité de donnée importée) sont effectués lors de l'import.

En fonction du bon déroulement de l'import et des contrôles, le fichier est déplacé dans l'un des dossiers suivants :

- **Archive** : si l'import est effectué avec succès
- **Error** : si l'import à échoué
- **Retry** : si l'on souhaite refaire de nouvelles tentatives d'import du même fichier unitaire.

Traitement "Retry" :

Le traitement de Retry permet de relancer l'import d'un fichier qui aurait rencontrée un erreur lors d'un import.

Le traitement de Retry s'appuie sur 2 paramètres :

- Le délai d'attente avant un Retry
- Le nombre de tentatives de Retry

Le principe du traitement est :

- Après un import "*normal*" avec erreur, le fichier importé est placé dans le dossier **Retry**.
- Après le délai d'attente du retry, le fichier est de nouveau déplacé vers le dossier d'import (**In**); et le système comptabilise la nouvelle tentative d'import.
- L'opération est répétées tant que le fichier soulève une erreur (configurée pour le retry).
- Si le nombre de tentative est atteint, après un nouvel import avec erreur, le fichier est finalement déplacé dans le dossier d'erreur (**error**)

3.4 Architecture Logiciel du projet

Notre solution est basée sur la plate-forme SAP Hybris et suit donc son architecture logicielle présentée dans la figure 3.11 :

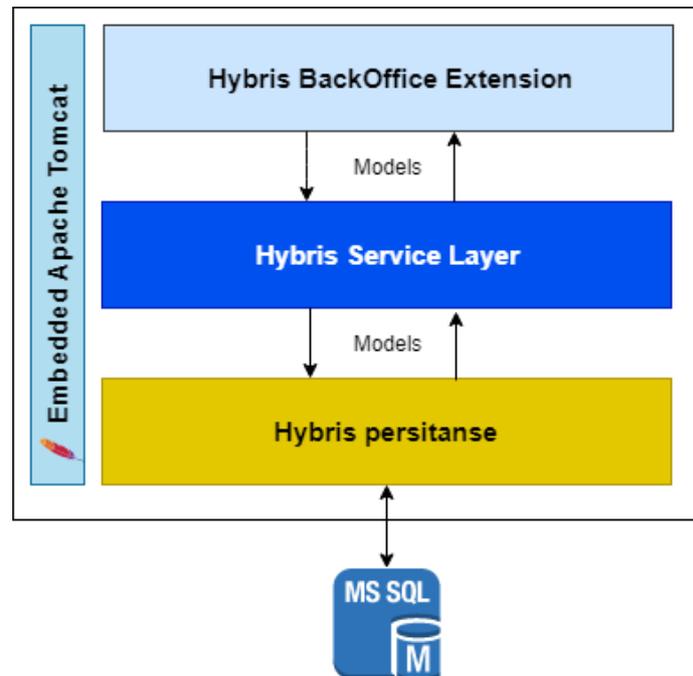


FIGURE 3.11 – Architecture logiciel du projet.

- la plate-forme Hybris inclut un serveur **Apache Tomcat** constituant son serveur Http.
- **l'extension Backoffice d'hybris** : c'est une extension Hybris qui permet aux utilisateurs métier d'avoir accès aux différentes fonctionnalités d'administration du contenu que se soit le catalogue, les catégories ou les produits ainsi que les différentes entités et types du système.
 - l'extension backoffice est basée sur le framework web **ZK**.
 - cette extension permet aux développeurs de créer ou bien de personnaliser les composants Hybris selon leurs besoins.
- **Le service Layer** de Hybris constituant la couche métier où il est implémenté la logique du business, il est constitué à son tour d'un ensemble de couches de services et communique avec le backoffice et la couche de persistance à travers les **Models** (les entités de la logique métier).
- **La couche persistance** d'hybris est l'intermédiaire entre la base de données **MS SQL Server** et le Service Layer.

3.5 Architecture physique du projet

Cette partie présente l'architecture physique du projet, avec les différents serveurs utilisés. L'architecture de notre projet est composé de deux parties : interne et externe.

La partie interne est l'architecture implémentée au niveau de SQLI Rabat, elle est essentiellement composée d'un seul serveur d'intégration INT (Intégration) en plus du serveur SonarQube² (un logiciel libre permettant de mesurer la qualité du code source en continu [8]) pour l'analyse du qualité de code.

La partie externe est celle implémentés dans l'environnement client et composée de trois serveurs : UAT pour le User Acceptance Testing, OAT pour l'Operational Acceptance Testing et PROD le serveur de production.

La figure 3.12 illustre cette répartition :

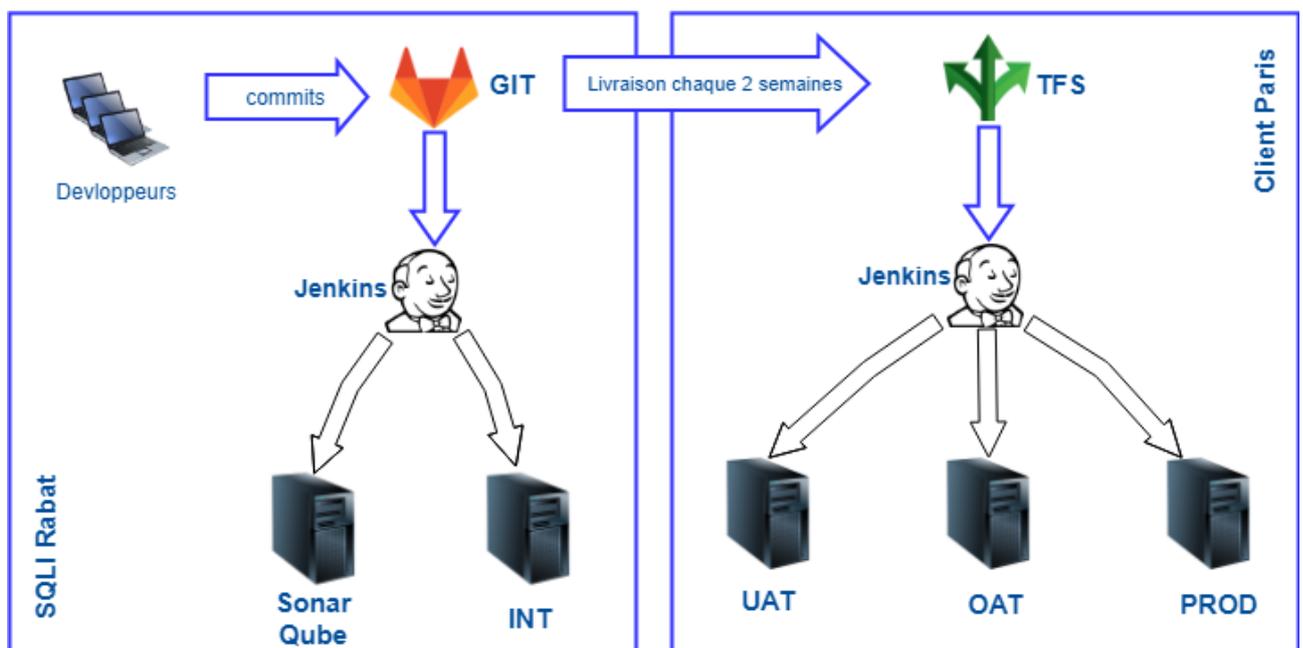


FIGURE 3.12 – Architecture physique du projet.

Les équipes de développement font des commits sur le serveur GIT (Gitlab), en suite un serveur d'intégration continue Jenkins (PIC ou Plate-forme d'Intégration Continue) récupère les versions produites depuis le GIT pour faire un build automatique avec ANT (un outil d'automatisation des opérations de build ou compilation des applications JAVA [9]) et lance une analyse Sonar depuis le serveur SonarQube sur le code afin de garantir sa qualité.

2. <https://www.sonarqube.org/>

Après avoir fait les analyses Sonar le serveur vérifie que le nombre d'erreurs trouvé ne dépasse pas les limites prédéfinies selon le degré de gravité des erreurs et les quotas définis dans le serveur SonarQube et notifie les développeurs des résultats d'analyse, puis il fait le déploiement dans le serveur d'intégration (INT).

Ce processus d'intégration continue se répète tous au long du sprint qui dure deux semaines et se termine par la livraison d'une nouvelle version au client dans son système de contrôle des versions TFS (Team Foundation Server).

Le client dispose de trois serveurs dont deux serveurs de test et un serveur de production :

- Le serveur UAT (User Acceptance Testing) sert à tester les livrables et s'assurer qu'ils correspondent aux attentes de l'utilisateur final.
- Le serveur OAT (Operational Acceptance testing) est un serveur qui permet de déterminer si les livrables sont opérationnelles et prêts à intégrer l'environnement de production.
- Après ces différents processus de test et dans chaque trois mois une opération de mise en production (MEP) s'effectue dans le serveur PROD pour qu'il soit disponible en ligne à l'utilisateur final.

3.6 Conclusion

Ce chapitre présente une vue conceptuelle de la solution à mettre en place. Il expose les différents diagrammes UML pour mieux comprendre les fonctionnalités offertes et pour mieux représenter la communication entre les différents objets du projet. Le chapitre suivant, présente la partie mise en œuvre de l'application.

Chapitre 4

Mise en oeuvre de la solution

“ Dans ce chapitre, on va présenter les outils utilisés pour la mise en oeuvre de la solution PIM ainsi qu’un aperçu des différentes vues et interfaces de cette solution. ”

4.1 Introduction :

Une des étapes de la vie d'un projet, aussi importante que la conception, est l'implémentation.

Cette étape constitue la phase d'achèvement et d'aboutissement du projet. Pour accomplir cette tâche avec succès il faut savoir utiliser les outils adéquats et nécessaires. Ce choix d'outils peut influencer la qualité du produit obtenu et donc nécessite une attention particulière et doit se baser sur les besoins du projet et le résultat escomptés.

Ce chapitre présente l'environnement technique du travail ainsi que le choix pris en matière d'environnement logiciel.

4.2 Outils et technologies utilisées :

4.2.1 La plateforme Java EE

Java EE ou Java Enterprise Edition (récemment renommée Jakarta EE) est une norme proposée par la société Oracle, portée par un consortium de sociétés internationales, visant à définir un standard de développement d'applications d'entreprises multi-niveaux[10].

Cette plate-forme est fortement orientée serveur et dédiée au développement et à l'exécution d'applications distribuées. Elle permet la simplification du processus de développement. En effet, JEE simplifie le contrôle et la gestion des ressources système en fournissant des méthodes permettant de gérer les transactions et la mise en commun des ressources.

4.2.2 La plateforme SAP Hybris

Hybris est une solution e-commerce créée en Allemagne, en 1997. Dès le début, la solution a ciblé son développement sur le « *e-commerce* », le « *PIM étendu* » et le « *multi canal* »[11].

Alors que certaines solutions sont clairement bâties pour faire la partie Front (produire du HTML pour faire des sites web), Hybris pense que le coeur du sujet, c'est pas le front, c'est le catalogue. L'idée, est de se dire qu'entre l'ERP, le système d'information de l'entreprise, et les sites Front, il y a besoin d'un middleware[11].

Ce référentiel sera souvent alimenté par un système de plus bas niveau (ERP ou équivalent). A partir de ce référentiel, l'autre axe clé de Hybris, c'est le multi canal. En effet, Hybris permet de publier de manière cohérente, vers les différents canaux de vente : Web, mobile, magasin, etc. . .

La plateforme SAP Hybris repose sur des standards ouverts, qui conjugue l'efficacité et l'extensibilité afin d'offrir des capacités infinies d'innovation et de devenir la plate-forme de commerce la plus performante du marché. Ainsi la plateforme repose sur plusieurs outils et frameworks open source et standardisés parmi lesquels on trouve :

- **Le Framework Spring**¹ : Un framework libre fournit un modèle complet de programmation et de configuration pour les applications d'entreprise modernes basées sur JEE. Le framework s'appuie principalement sur les principes de l'IOC (Inversion of Control) et de MVC (Model View Controller) [12].
- **Apache Tomcat**² : C'est un conteneur web libre de servlets et JSP Java EE. Issu du projet Jakarta. Il incorpore également un serveur HTTP[13].
- **Apache Solr**³ : C'est le moteur de recherche et d'indexation de référence dans le monde du libre. Basé sur la librairie Lucene, il dispose de nombreuses fonctionnalités avancées comme la recherche par facettes, la suggestion orthographique et la recherche par similarité [14].
- **ZK Framework**⁴ : C'est un framework open source web 2.0, proposant une interaction utilisateur riche. ZK permet tout autant de définir rapidement des interfaces graphiques via une syntaxe XML ou un éditeur Wysiwyg qui permet de manipuler directement les objets en Java [15].
- **Apache Ant**⁵ : Ant est un logiciel créé par la fondation Apache qui vise à automatiser les opérations répétitives du développement de logiciel telles que la compilation, la génération de documents ou l'archivage au format JAR, à l'instar des logiciels Make. Ant est écrit en Java et son nom est un acronyme pour « Another Neat Tool » [9].

La plateforme SAP Hybris dispose par défaut des espaces d'administration suivants :

- **Le Framework Backoffice** : C'est la nouvelle version de l'ancien HMC (Hybris Management Console), une interface backend du système qui permet de gérer les utilisateurs, les marchés et leurs catalogues de produits et les différents types du système, il permet les opérations de création, mise à jour, suppression et de recherche à travers des millions d'articles[5].
- **HAC (Hybris Administration Console)** : C'est une interface backend du système, il permet d'effectuer des opérations d'administrations comme la supervision des performances système, la modification à chaud des paramètres généraux, l'administration du cache système, de session, d'accès au logs etc[5].

1. <https://spring.io/>
2. <http://tomcat.apache.org/>
3. <https://lucene.apache.org/solr/>
4. <https://www.zkoss.org/>
5. <https://ant.apache.org/>

4.2.3 Outils de développement et de collaboration :

L'IDE IntelliJ IDEA :

Également appelé « IntelliJ », « IDEA » ou « IDJ » est un environnement de développement intégré de technologie Java destiné au développement de logiciels informatique. Il est développé par JetBrains⁶ (anciennement « IntelliJ ») et disponible en deux versions, l'une communautaire, open source, sous licence Apache 2 et l'autre propriétaire, protégée par une licence commerciale [16].

Le système de contrôle des versions GIT :

C'est un logiciel libre de gestion de versions, créé par Linus Torvalds (le créateur du noyau Linux), c'est un outil bas niveau, qui se veut simple et très performant, dont la principale tâche est de gérer l'évolution du contenu d'une arborescence. Il fonctionne en mode distribué avec un serveur distant[17].

Atlassian JIRA :

C'est un système de suivi de bugs, un système de gestion des incidents, et un système de gestion de projets développé par Atlassian. Jira⁷ n'est pas un acronyme mais une troncation par aphérèse de Gojira[18].

4.3 Quelques interfaces de la plate-forme

Les différentes interfaces qu'on présentera par la suite sont basé sur des composants ou bien des widgets d'Hybris qu'ils ont été personnalisé et évolués selon le besoin de la plate-forme.

4.3.1 La page de Login

La figure 4.1 représente la page de login des utilisateurs. Cette page est composée des éléments suivants :

1. Logo
2. Formulaire d'authentification
3. Liste des langues disponibles

6. <https://www.jetbrains.com/idea/>

7. <https://www.atlassian.com/software/jira>

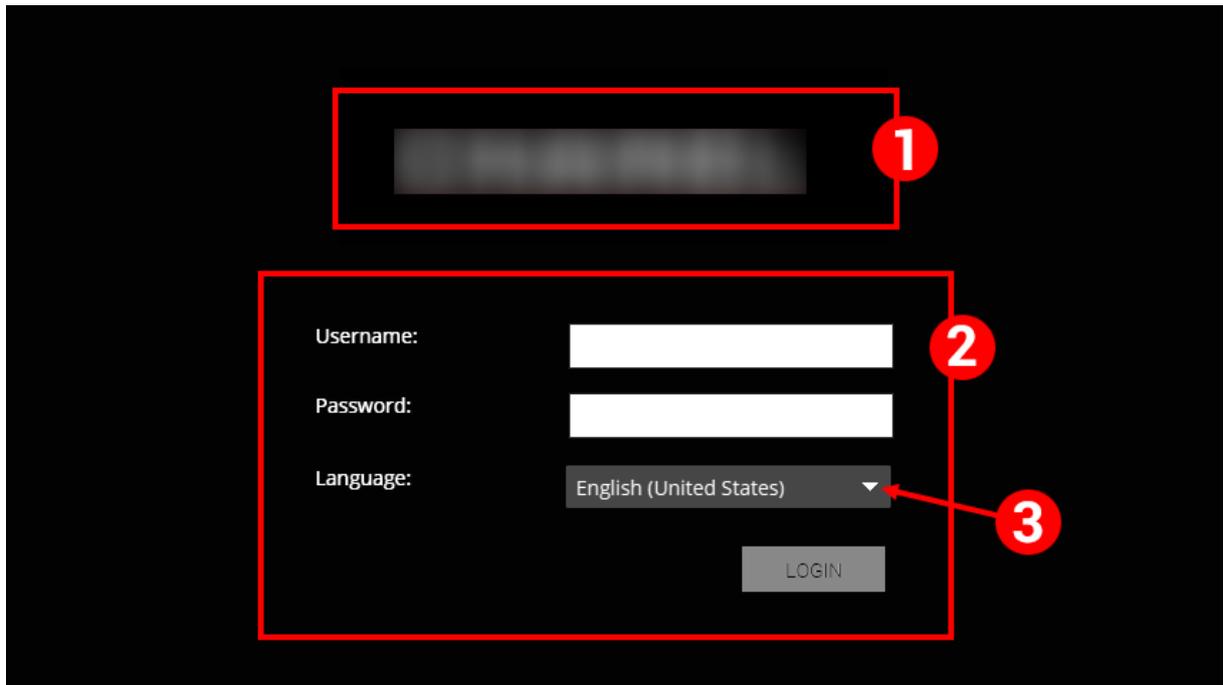


FIGURE 4.1 – La page du login de la plateforme PIM.

4.3.2 La Vue Administration

La vue Administration regroupe toutes les opérations de gestion des entités du PIM qu'elle soient natives à hybris comme les cronjob, les employés, les catalogues etc... ou celles spécifiquement créés pour le projet.

Cette vue est présentée dans les deux figures 4.2 4.3 est composé des éléments suivants :

1. Le menu d'administration qui regroupe les élément administrable dans le site sous forme d'arbres.
2. Le composant de recherche dans les éléments/types du menu d'administration.
3. Le composant de recherche dans la liste des objets de l'élément/type sélectionné.
4. La liste des objets disponibles pour l'élément/le type sélectionné.
5. Le composant d'édition des informations de l'objet sélectionne dans la liste 4.
6. Les icônes de déconnexion, langues, profiles et d'autres natives a Hybris.

Dans la figure 4.2 le type **Référence** est sélectionné alors que dans la figure 4.3 le type de catégorie **Communication Fonctionnality** est sélectionnée :

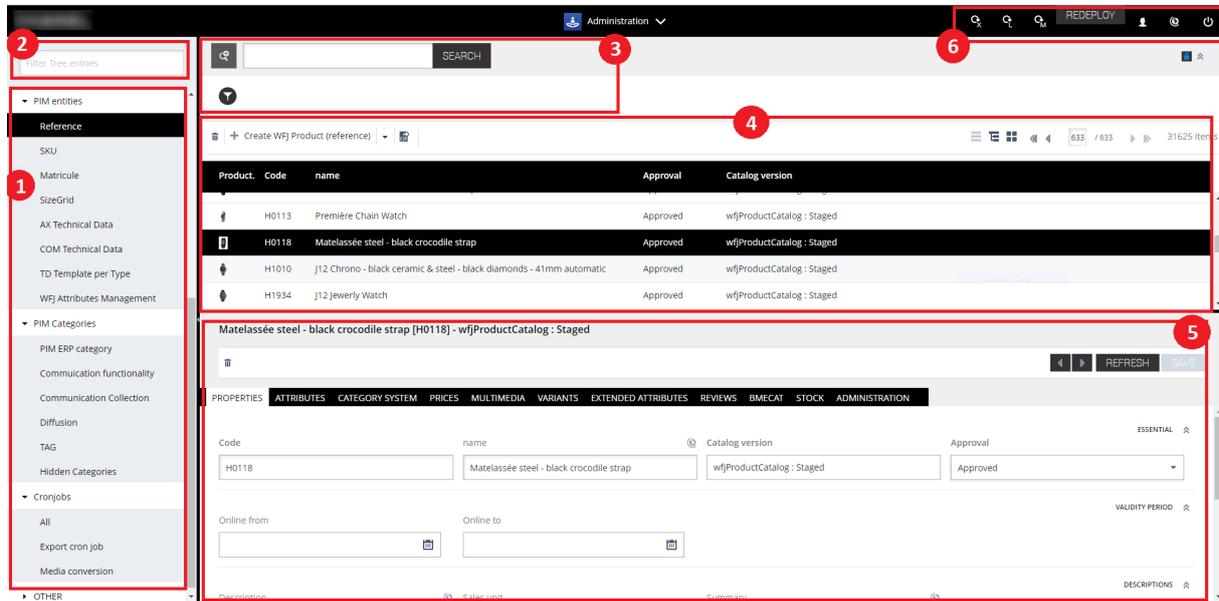


FIGURE 4.2 – La vue Administration de la plate-forme PIM.

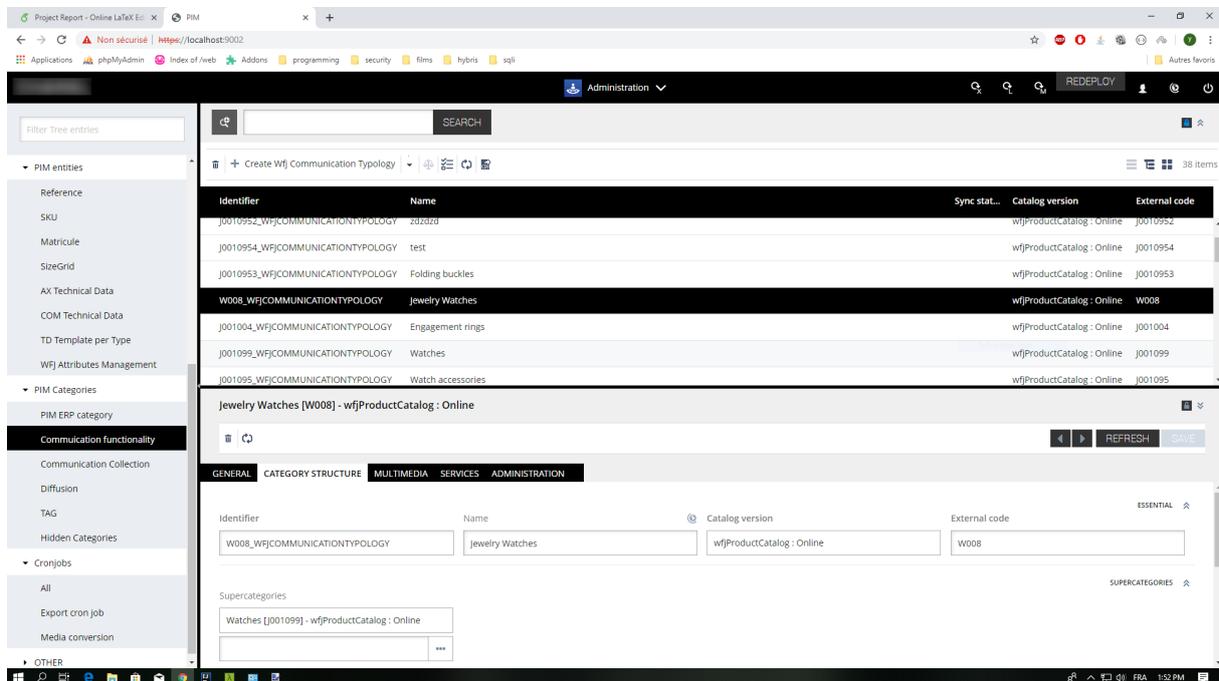


FIGURE 4.3 – La vue Administration avec un type de catégories sélectionnée.

4.3.3 La Vue Catalogue

La vue catalogue présenté dans la figure 4.4 est une espace de gestion avancé des produits du catalogue, elle est principalement composée des éléments suivants :

1. Les composants de filtrage selon les catégories et les filtres prédéfinis.
2. Le composant de recherche des produits.
3. La liste des produits recherchés.
4. Le nom de la vue "Wfj Catalog".
5. L'image et la présentation du produit sélectionné.
6. Le composant d'édition avancé des information du produit sélectionné.

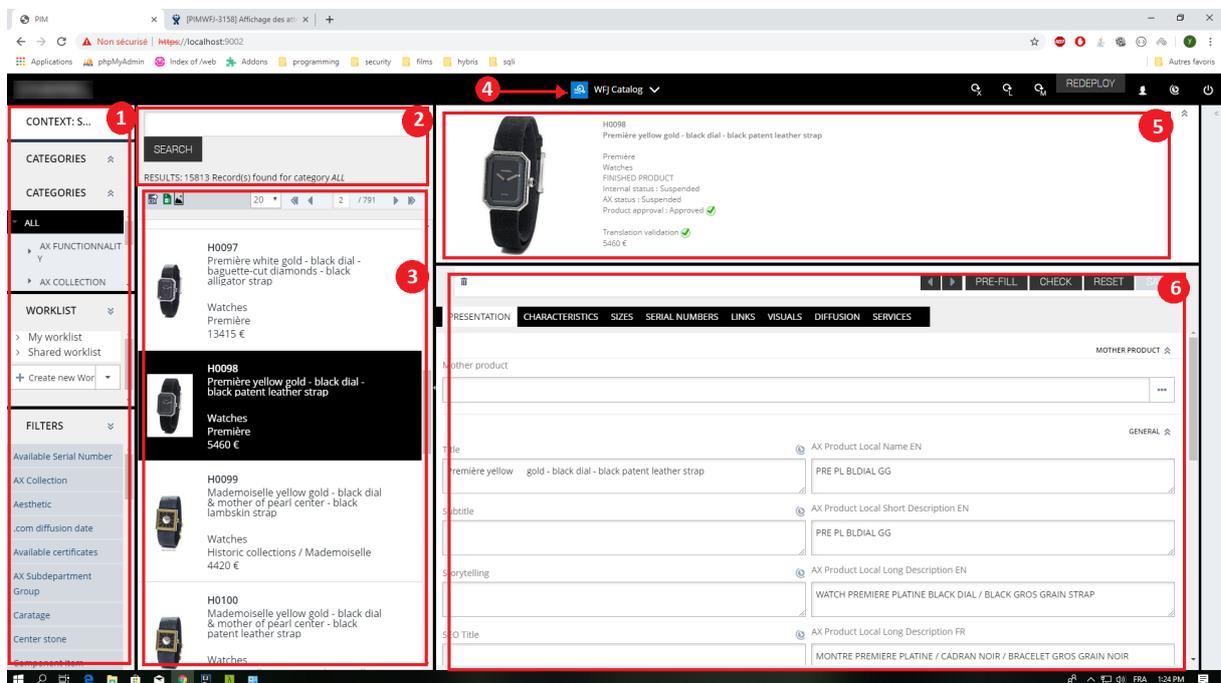


FIGURE 4.4 – La vue Catalogue de la plate-forme.

4.3.4 La Vue Catégories

La vue catégories La vue categories présenté dans la figure 4.5 est une perspective qui permet de :

- Gérer les catégories existantes au sein du PIM.
- Gérer des produits sur une catégorie sélectionnée, ie :

- Rechercher des produits.
- Associer des produits à une catégorie.

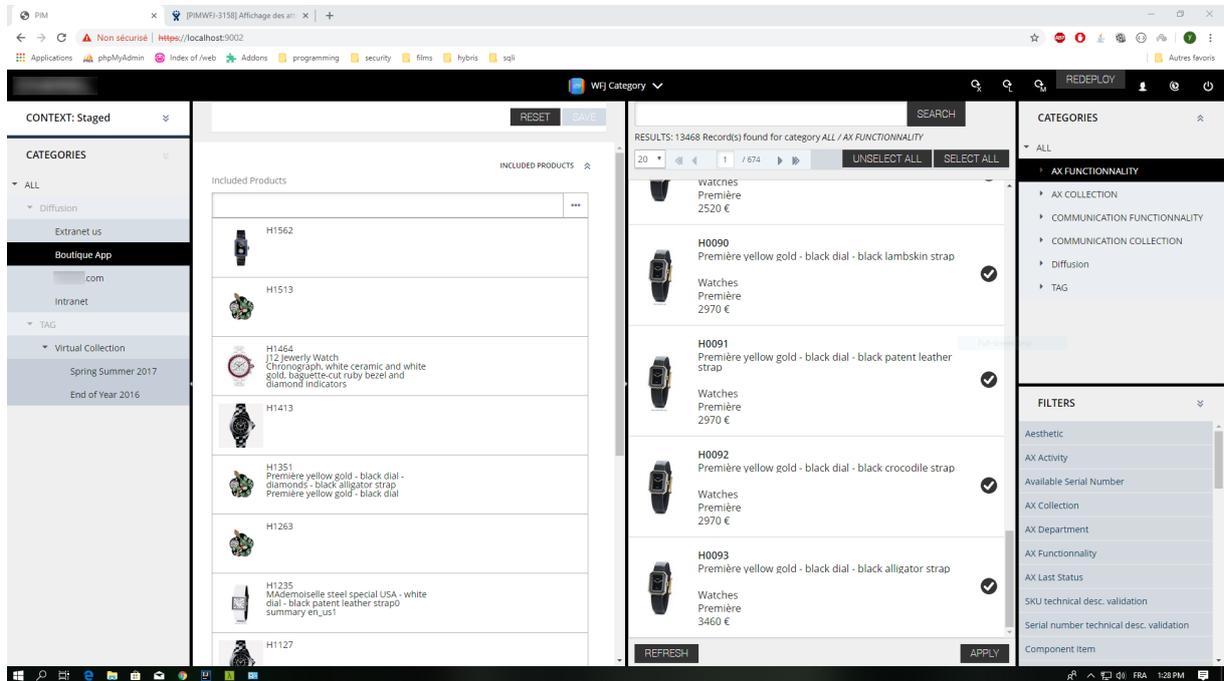


FIGURE 4.5 – La vue Categories de la plateforme.

4.3.5 La Vue Dashboard

Le dashboard WFJ permet d'afficher des indicateurs de suivi (voir la figure 4.6) :

- Un dashboard par statut (Approval Status) qui permet de voir le statut des produits par catégories (collection de communication).
- Un dashboard qui indique le nombre de produits créés sur les 12 derniers mois.
- Un dashboard pour les traductions par langue qui indique les produits qui ne sont pas traduits.

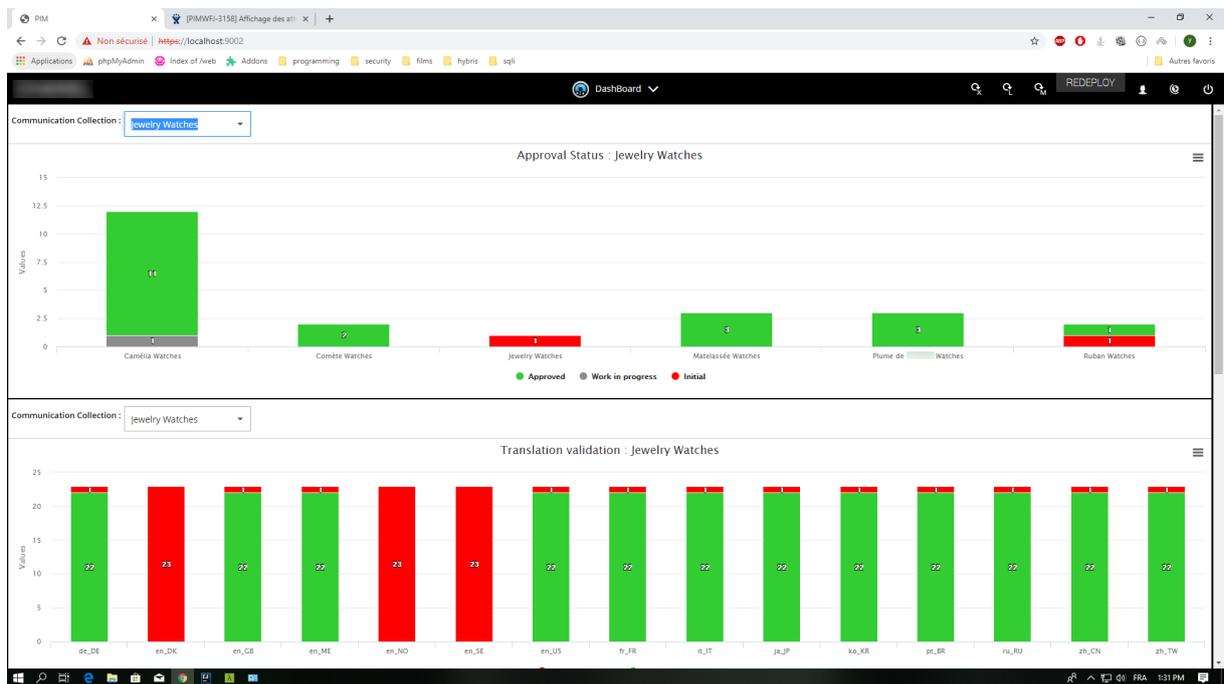


FIGURE 4.6 – La vue Dashboard de la plate-forme.

4.4 Conclusion

Dans ce chapitre, on a présenté la réalisation qu'on a effectué, l'environnement de développement, les outils et technologies utilisés dans le projet ainsi qu'une description détaillée des différentes interfaces utilisateur de la plate-forme.

Conclusion générale

Mon projet de fin d'études au sein de SQLI consistait en la mise en œuvre d'une solution PIM en faveur d'une maison de produits de luxe à base de la plateforme e-commerce SAP Hybris. La solution avait pour objectif de permettre au client de gérer les informations relatives à ses produits d'une façon centralisée et ce pour des raisons d'enrichissement et de diffusion vers les différents canaux comme les sites e-commerce, les points de vente, les applications mobiles, etc.

La première itération du projet consistait en la définition du périmètre du projet. Les itérations qui suivaient ont été consacrées à la réalisation de plusieurs modules métiers de base comme le flux d'import et la gestion des profils. Lors de chaque itération, une étude fonctionnelle a été effectuée suivie d'une conception détaillée permettant par la suite d'enchaîner la phase de réalisation et de développement des modules en question.

Le présent projet m'a apporté de la valeur ajoutée sur plusieurs niveaux. D'une part, j'ai appris à mieux gérer mon temps, mieux communiquer et collaborer avec les autres membres de l'équipe de développement. D'autre part, il m'a permis d'acquérir des compétences extrêmement importantes telle que le maintien et l'optimisation de mon code, l'intégration des bonnes pratiques, l'autonomie ainsi qu'une grande capacité de modélisation.

En terme de perspectives, le projet dans sa version actuelle n'est pas encore achevé, on est actuellement en cours de personnalisation de l'extension backoffice pour faciliter encore plus l'accès et la gestion des informations des produits. Les prochaines itérations porteront toujours sur ce sujet en plus de l'évolution du processus d'export des produits pour qu'il soit beaucoup plus personnalisable par l'utilisateur.

Webographie

- [1] Wikipédia, “Sqli — wikipédia, l’encyclopédie libre.” <http://fr.wikipedia.org/w/index.php?title=SQLI&oldid=158242840>, 2019. [Consulté le 7 avril 2019].
- [2] Wikipédia, “Capability maturity model integration — wikipédia, l’encyclopédie libre.” http://fr.wikipedia.org/w/index.php?title=Capability_Maturity_Model_Integration&oldid=154504497, 2018. [Consulté le 7 avril 2019].
- [3] Wikipédia, “Product information management — wikipédia, l’encyclopédie libre.” http://fr.wikipedia.org/w/index.php?title=Product_information_management&oldid=154797272, 2018. [Consulté le 13 juin 2019].
- [4] Naolis, “C’est quoi un pim?.” <http://naolis.com/portail/solutions/akeneo/cest-quoi-un-logiciel-pim>, 2019. [Consulté le 13 juin 2019].
- [5] SAP, “Administering sap commerce.” <https://help.sap.com/viewer/5c9ea0c629214e42b727bf08800d8dfa/6.7.0.0/en-US>, 2019. [Consulté le 13 juin 2019].
- [6] Wikipédia, “Gestion des ressources numériques — wikipédia, l’encyclopédie libre.” http://fr.wikipedia.org/w/index.php?title=Gestion_des_ressources_num%C3%A9riques&oldid=156929136, 2019. [Consulté le le 16 juin 2019].
- [7] C. Pages, “Xml common business library (xcbl).” <http://xml.coverpages.org/cbl.html>, 2001. [Consulté le le 16 juin 2019].
- [8] Wikipédia, “Sonarqube — wikipédia, l’encyclopédie libre.” <http://fr.wikipedia.org/w/index.php?title=SonarQube&oldid=159160337>, 2019. [Consulté le le 16 juin 2019].
- [9] Wikipédia, “Apache ant — wikipédia, l’encyclopédie libre.” http://fr.wikipedia.org/w/index.php?title=Apache_Ant&oldid=154452007, 2018. [Consulté le le 16 juin 2019].
- [10] Wikipédia, “Jakarta ee — wikipédia, l’encyclopédie libre.” http://fr.wikipedia.org/w/index.php?title=Jakarta_EE&oldid=159136465, 2019. [Consulté le le 9 mai 2019].
- [11] F. Ziserman, “Hybris, solution pour bâtir un vrai système d’information e-commerce.” <http://www.ziserman.com/blog/2011/06/22/>

- [hybris-solution-pour-batir-un-vrai-systeme-dinformation-e-commerce/](#), 2019. [Consulté le le 8 juin 2019].
- [12] Wikipédia, “Spring (framework) — wikipédia, l’encyclopédie libre.” [http://fr.wikipedia.org/w/index.php?title=Spring_\(framework\)&oldid=159856465](http://fr.wikipedia.org/w/index.php?title=Spring_(framework)&oldid=159856465), 2019. [Consulté le le 16 juin 2019].
- [13] Wikipédia, “Apache tomcat — wikipédia, l’encyclopédie libre.” http://fr.wikipedia.org/w/index.php?title=Apache_Tomcat&oldid=159469213, 2019. [Consulté le le 8 juin 2019].
- [14] Wikipédia, “Apache solr — wikipédia, l’encyclopédie libre.” http://fr.wikipedia.org/w/index.php?title=Apache_Solr&oldid=159141307, 2019. [Consulté le le 16 juin 2019].
- [15] Wikipédia, “Zk (informatique) — wikipédia, l’encyclopédie libre.” [http://fr.wikipedia.org/w/index.php?title=ZK_\(informatique\)&oldid=155242227](http://fr.wikipedia.org/w/index.php?title=ZK_(informatique)&oldid=155242227), 2018. [Consulté le le 16 juin 2019].
- [16] Wikipédia, “Intellij idea — wikipédia, l’encyclopédie libre.” http://fr.wikipedia.org/w/index.php?title=IntelliJ_IDEA&oldid=158814565, 2019. [Consulté le le 16 juin 2019].
- [17] Wikipédia, “Git — wikipédia, l’encyclopédie libre.” <http://fr.wikipedia.org/w/index.php?title=Git&oldid=160009701>, 2019. [Consulté le le 8 juin 2019].
- [18] Wikipédia, “Jira — wikipédia, l’encyclopédie libre.” <http://fr.wikipedia.org/w/index.php?title=Jira&oldid=156573214>, 2019. [Consulté le le 8 juin 2019].