

Mémoire de projet de fin d'étude

Présenté par

TALAGHZI Hamza

En vue de l'obtention du diplôme

Master

Ingénierie de Données et Développement Logiciel



Refonte d'un portail de gestion électronique des documents (Arkevia Refonte)



Soutenu le 30/09/21

Encadrant interne :

Pr. ZITI Soumia

Encadrant externe :

M. GHAICH Mourad

Membres du Jury :

Pr. ZITI, Soumia

PH, FSR-UM5

Président

Pr. EZZAHOUT, Abderrahmane

PA, FSR-UM5

Examinateur

Pr. KHARMOUM, Nassim

PA, CNRST

Examinateur

Avant-propos

Nom et prénom de l'étudiant :

TALAGHZI Hamza

Intitulé du travail :

Refonte d'un portail de gestion électronique des documents (Arkevia Refonte)

Établissement d'accueil :

- **Nom de l'entreprise** : CEGEDIM MAROC
- **Adresse** : Arribat Center Immeuble D et E 2ème étage Avenue Omar Ibn Khattab 10090 Agdal Rabat.
- **Site web** : <https://www.cegedim.fr>

Encadrant de l'établissement d'accueil :

M. GHAICH Mourad

Tuteur académique de la Faculté des Sciences de Rabat :

Mme. ZITI Soumia

Date de début et de fin du stage :

Du **15 Février 2021** au **15 Août 2021** inclus

Remerciement

Au terme de mon projet de fin d'études, je tiens à exprimer ma profonde gratitude et mes sincères remerciements pour tous ceux qui ont aidé de près ou de loin à la réalisation de ce travail.

Je tiens à remercier, Mme. Soumia ZITI mon encadrante interne à la faculté des sciences de Rabat, pour ses efforts et ses conseils, et pour le soutien qu'elle m'a prodigué tout au long de cette période de stage. Je la remercie plus particulièrement pour sa patience et sa gentillesse.

Je remercie vivement M. Mourad GHAICH mon encadrant interne à Cegedim SRH, qui m'a consacré son temps et son énergie et m'a fait bénéficier de son expertise et de ses précieux conseils.

Je tiens à remercier aussi M. Lotfi WAHBI chargé de projet R&D, de m'avoir accordé l'opportunité de bénéficier d'une expérience professionnelle enrichissante au sein d'une équipe dynamique.

À la même occasion, j'aimerais en profiter et remercier l'équipe de dév R&D pour le transfert de compétences qui m'a été très utile dans l'avancement de mon sujet ainsi que pour leur sens de la collaboration et du partage d'expérience.

Que les membres du jury trouvent ici l'expression de ma reconnaissance d'avoir accepté de juger ce travail.

Sommaire

Avant-propos	ii
Remerciement	iii
Sommaire	iv
Liste des figures	vii
Liste des tables	viii
Glossaire des acronymes	ix
Introduction générale	1
Contexte du projet	1
Objectifs et missions	1
Contenu du mémoire	1
1 Présentation de l'organisme d'accueil	3
Introduction	3
1.1 Groupe Cegedim	3
1.1.1 Présentation	3
1.1.2 Le Groupe Cegedim en quelques chiffres	4
1.1.3 Domaines d'activité	5
1.2 Cegedim SRH	6
1.2.1 Présentation	6
1.2.2 Domaines fonctionnels	7
1.2.3 Agences et centres de services	9
1.2.4 Organigramme SRH	9
Conclusion	10
2 Présentation du projet	11
Introduction	11
2.1 Étude préliminaire	11
2.1.1 Qu'est-ce que la GED?	11
2.1.2 Bénéfices d'une GED	11
2.2 Étude de l'existant	12
2.2.1 Présentation du projet Arkevia	12
2.2.2 Architecture globale d'Arkevia	13
2.2.2.1 Généralités	14

SOMMAIRE

2.2.2.2 Présentation métier	14
2.2.2.3 Vue d'ensemble	16
2.3 Problèmes et axes d'amélioration	17
2.3.1 Défis technologiques	17
2.3.2 Refactorisation et nettoyage du code	17
2.3.3 Mécanisme de Notification	18
2.4 Objectif de la refonte	18
Conclusion	18
3 Méthodologie de gestion de projets	19
Introduction	19
3.1 Étude préliminaire	19
3.1.1 Origine du génie logiciel	19
3.1.2 Comparaison des différentes méthodes de développement	20
3.1.2.1 Les approches traditionnelles	20
3.1.2.2 Les méthodes agiles	20
3.1.2.3 Synthèse	20
3.2 Conduite de projet	21
3.2.1 Modèle de livraison	21
3.2.2 Déroulement de projets - Méthode SCRUM	25
3.2.2.1 L'équipe Agile	25
3.2.2.2 Le Sprint	26
3.2.2.3 Les Rituels	26
3.2.3 Processus de développement	28
3.2.4 Gestion du workflow git	29
3.3 Planification et suivi du projet	31
3.3.1 Diagramme de Gantt	31
Conclusion	33
4 Conceptions et spécification des besoins	34
Introduction	34
4.1 Étude conceptuelle du système existant	34
4.1.1 Méthodologie de conception	34
4.1.2 Identification des acteurs	34
4.1.3 Diagramme de cas d'utilisation général	35
4.1.4 Raffinement des cas d'utilisation prioritaires	37
4.2 Analyse des besoins	46
4.2.1 Migration du socle technique	46
4.2.2 Refactorisation et nettoyage du code	48

SOMMAIRE

4.2.2.1	Refactorisation des modules applicatifs	48
4.2.2.2	Renoncer à recourir à la brique de contrôle	49
4.2.2.3	Retrait des dépendances inutilisables	49
4.2.3	Optimisation du mécanisme d'envoi des notifications	50
Conclusion	54
5	Réalisation	55
	Introduction	55
5.1	Environnements et outils	55
5.1.1	Composants logiciels	55
5.1.1.1	Front-end	55
5.1.1.2	Back-end	57
5.1.1.3	Système de gestion de base de données	58
5.1.1.4	Stockage et sécurité des données	58
5.1.2	Environnement de développement	59
5.1.2.1	Environnement matériel	59
5.1.2.2	Environnement logiciel et outils	59
5.2	Démonstration de la fiabilité du nouveau système d'envoi de notifications	62
5.2.1	Préparation de l'environnement et paramétrage du système . . .	65
5.2.2	Exécution, résultats et interprétations	67
5.2.3	Tests unitaires	70
5.2.4	Audit de code	71
5.2.5	Benchmark	74
Conclusion	74
Conclusion générale		76
A	ANNEXES	78
A.1	Présentation des principales interfaces d'Arkevia	78
A.2	Audit de code : checklist élaborée par Cegedim SRH	80
A.3	Outils utilisés pour la réalisation de ce rapport	82
Bibliographie		83

Liste des figures

1.1	Logo Cegedim	3
1.2	Écosystème Cegedim	3
1.3	Groupe Cegedim en chiffres	4
1.4	Présence mondiale du groupe cegedim	4
1.5	Logo Cegedim SRH	7
1.6	Clients Cegedim SRH	7
1.7	Les différents sites SRH	9
1.8	Organigramme des Directions Cegedim SRH au 1er janvier 2019	10
2.1	Répartition des tâches dans l'écosystème Arkevia	16
3.1	Modèle de livraison	22
3.2	Modèle organisationnel visé pour les équipes de développement et de QA	23
3.3	Schéma illustrant l'interaction entre les différentes branches du flux de travail gitflow	31
3.4	Diagramme de Gantt	32
4.1	Diagramme de cas d'utilisation général	36
4.2	Architecture modulaire d'Arkevia	49
4.3	Diagramme de séquence modélisant le processus d'envoi de notifications	53
5.1	Liens entre les divers composants impliqués dans le traitement par lots	63
5.2	Exemple d'alerte envoyée au superviseur	69
5.3	Modèle générique de mail pour l'envoi de notifications	69
5.4	Mise en place d'un projet Sonar	72
5.5	Résultats récapitulatifs d'analyse SonarQube	73
A.1	Page d'authentification Arkevia	78
A.2	Page de gestion des documents	79
A.3	Page de gestion de compte Arkevia	80

Liste des tables

3.1	Responsabilités et missions des différents acteurs de l'équipe de dév	23
3.2	Responsabilités et missions des différents acteurs de l'équipe QA	24
3.3	Présentation des différentes branches définies sur gitflow	30
4.1	Description textuelle du CU « S'inscrire »	37
4.2	Description textuelle du CU « Se connecter »	38
4.3	Description textuelle du CU « Créer un nouveau répertoire »	38
4.4	Description textuelle du CU « Déposer un document »	39
4.5	Description textuelle du CU « Gérer un document »	40
4.6	Description textuelle du CU « Gérer l'affichage des documents »	40
4.7	Description textuelle du CU « Partager un répertoire »	42
4.8	Description textuelle du CU « Supprimer un partage »	43
4.9	Description textuelle du CU « Modifier son adresse email »	44
4.10	Description textuelle du CU « Modifier son mot de passe »	44
4.11	Description textuelle du CU « Consulter son historique de connexion »	45
4.12	Description textuelle du CU « Désigner un contact de confiance »	45
4.13	Description textuelle du CU « Désactiver son compte »	46
4.14	Exemple de déroulement du Cron	51
4.15	Description des paramètres de la fonction <code>getLastPushedFiles</code>	51
5.1	Technologies utilisées au niveau du front-end	55
5.2	Technologies utilisées pour les solutions back-end	57
5.3	Environnements et outils de développement et de collaboration	60
5.4	Exemple de fichiers CSV générés pour le test	66
5.5	Comparaison des délais d'exécution entre l'ancien et le nouveau système d'envoi de notifications	74
A.1	Extrait des règles et pratiques de développement logiciel instaurées par Cegedim SRH	80
A.2	Supports utilisés pour la réalisation de ce rapport	82

Glossaire des acronymes

Acronyme	Désignation
API	Application Programming I nterface
B2B	B usiness t o B usiness
BPO	B usiness P rocess O utsourcing
BU	B usiness U nit
CA	C hiffre d' a ffaires
CI/CD	C ontinuous I ntegration / C ontinuous D elivery
CSS	C ascading S tyle S heets
DSI	D irection des S ystèmes d' I nformation
EDI	E lectronic D ata I nterchange
GED	G estion É lectronique des D ocuments
GTA	G estion des T emps et A ctivités
JVM	J ava V irtual M achine
ORM	O bject- R eferential M apping
POJO	P laint O ld J ava O bject
QA	Q uality A surance
R&D	R esearch and D evelopment
RDP	R emote D esktop P rotocol

Suite à la page suivante

GLOSSAIRE DES ACRONYMES

RH	Ressources Humaines
SaaS	Software as a Service
SASS	Syntactically Awesome Style Sheets
SE	Système d'exploitation
SGBDR	Système de Gestion de Bases de Données Relationnelles
SIRH	Système d'Information Ressources Humaines
SRH	Service des Ressources Humaines
UML	Unified Modeling Language

Introduction générale

Contexte du projet

Face à la croissance de l'activité de Cegedim SRH, et dans l'optique de garantir un produit de haute qualité, cette dernière a décidé d'adopter une stratégie d'amélioration et d'évolution des systèmes existants.

L'un des sujets qualifiés au titre des priorités fut la refonte d'un portail de gestion électronique de documents appelé Arkevia, également connu sous le nom de coffre-fort électronique salarié, un outil de réception et de stockage sécurisé dans lequel le salarié peut stocker ses documents professionnels et personnels : bulletins de paie, ou tout autre document importé au format dématérialisé, pièces d'identités, diplômes, factures, ou autres documents personnels. Ce portail a subi plusieurs changements et évolutions au cours de ces dernières années, rendant l'application volumineuse et difficile à gérer et à comprendre. Malgré les efforts déployés pour maintenir un code modulaire et évolutif, cela n'a pas été suffisant pour réduire la complexité du produit ni à virer les pratiques et méthodes obsolètes ni à corriger les problèmes de performances qui ne cessaient d'augmenter de façon spectaculaire avec le nombre croissant d'utilisateurs.

Conscient de ces enjeux, le département R&D de Cegedim SRH de Rabat a opté pour s'engager dans la refonte de ce portail pour traiter les problèmes et les axes d'amélioration identifiés suite à l'audit architectural réalisé. C'est dans ce contexte que le département R&D m'a confié ce sujet dont la mission principale est la refonte du portail Arkevia.

Objectifs et missions

Le stage est axé sur le développement de la refonte de la structure du portail Arkevia afin de répondre aux besoins suivants :

- Analyse de l'existant et formalisation du besoin;
- Revue du cœur de l'application et migration vers de nouvelles technologies;
- Restructuration du projet et du mécanisme d'envoi de notifications;
- Réalisation de nouvelles fonctionnalités;
- Adaptation au processus de livraison;
- Réalisation de tests unitaires;
- Rédaction de rapports techniques sur l'avancement du sujet.

Contenu du mémoire

Ce travail a abouti à la rédaction du présent mémoire, il est composé de cinq chapitres :

- **Le premier chapitre** est consacré à la présentation de l'organisation d'accueil, de ses domaines d'activité et de sa hiérarchie.
- **Le deuxième chapitre** constitue une étude des aspects du système existant, de son architecture globale et de son fonctionnement, permettant de révéler de manière synthétique le contexte général du projet, d'identifier les problèmes et de proposer des solutions et des axes d'amélioration.
- **Le troisième chapitre** couvrira les méthodes et pratiques employées pour la conduite de projets et l'organisation des tâches au sein de l'organisme d'accueil.
- **Le quatrième chapitre** est consacré à la conception détaillée du portail Arkevia (le portail existant), ainsi qu'à l'élaboration des besoins, des contraintes et des solutions requises.
- **Le cinquième chapitre** sera consacré aux travaux réalisés pendant la période du stage, une première partie sera consacrée à la présentation des différents outils et environnements de travail par lesquels la mise en œuvre de la refonte du système ARKEVIA a été réalisé, puis dans une seconde partie, à une démonstration d'autres travaux confiés à ma charge, à savoir le système d'envoi de notifications.

Chapitre 1

Présentation de l'organisme d'accueil

Introduction

Ce chapitre a pour vocation de présenter l'organisme d'accueil, ses activités et sa structure interne.

1.1 Groupe Cegedim

1.1.1 Présentation

Fondée en 1969, Cegedim (**C**entre de **G**estion, de **D**ocumentation, d'**I**nformatique et de **M**arketing) est une entreprise de technologies et de services spécialisée dans la gestion des flux numériques de l'écosystème santé et B2B, ainsi que dans la conception de logiciels métiers destinés aux professionnels de santé et de l'assurance.



Figure 1.1. Logo Cegedim

Ses offres s'adressent notamment aux professionnels de santé, industries de santé, laboratoires pharmaceutiques, et compagnies d'assurance. Le Groupe est également présent dans les métiers de la gestion des ressources humaines et de la dématérialisation pour tous types d'industries.

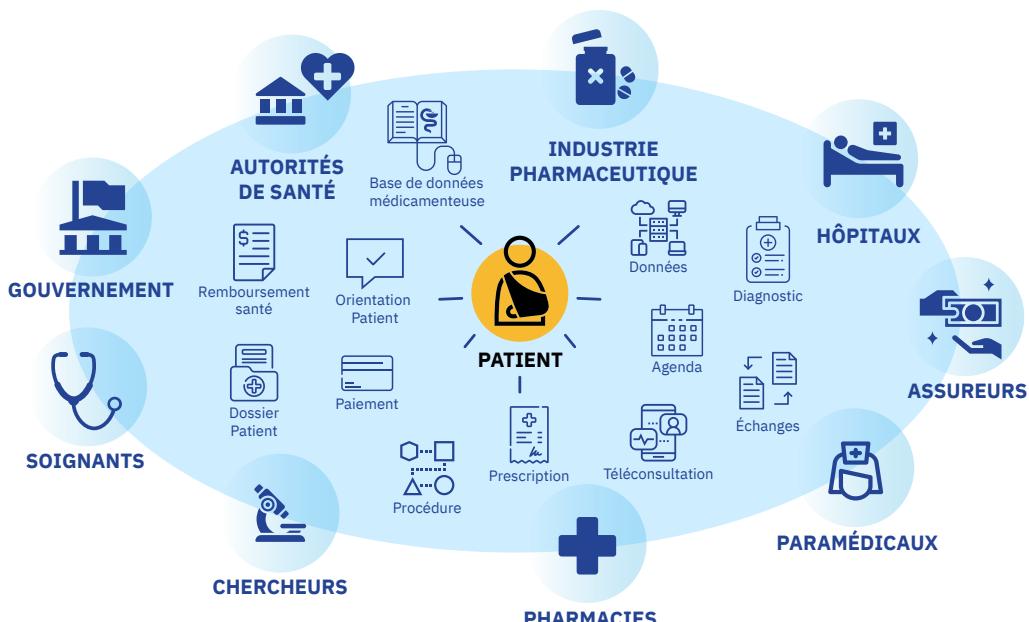


Figure 1.2. Écosystème Cegedim

1 PRÉSENTATION DE L'ORGANISME D'ACCUEIL

1.1.2 Le Groupe Cegedim en quelques chiffres

Cegedim compte près de 5 300 collaborateurs dans plus de 10 pays (voir figure 1.4) et a réalisé un chiffre d'affaires de 496.9 millions d'euros en 2020 [3]. Cegedim SA est cotée en bourse à Paris (EURONEXT : CGM).



Figure 1.3. Groupe Cegedim en chiffres

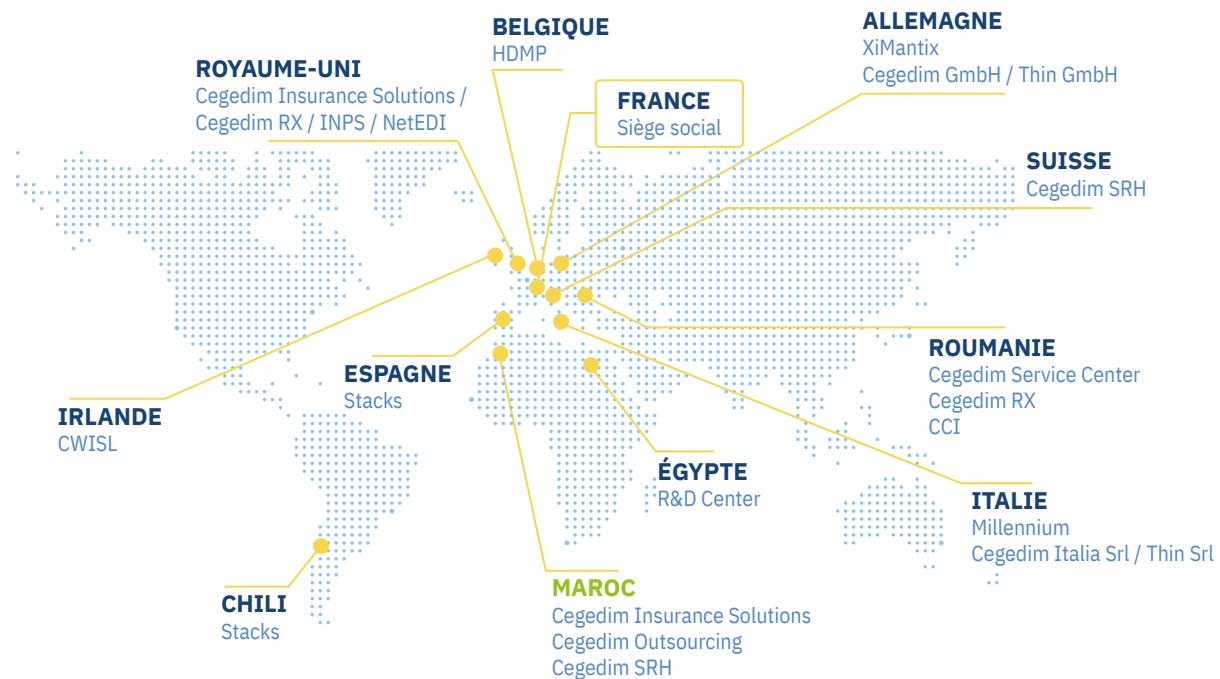


Figure 1.4. Présence mondiale du groupe cegedim

1. Réalisé à partir d'une enquête par questionnaire en ligne. Le Palmarès a été réalisé sur la base des données transmises par chaque entreprise participante, et complétées dans certains cas par des sources extérieures. Certaines données, de nature confidentielle, sont traitées uniquement de façon agrégée. Lire le classement complet sur le site [Truffle 100](#) (publié le 8 juillet 2021).

1 PRÉSENTATION DE L'ORGANISME D'ACCUEIL

1.1.3 Domaines d'activité

Alliant maîtrise technologique des données, du numérique et des réseaux, les activités du groupe Cegedim se déclinent autour de **quatre divisions** opérationnelles. Ce découpage vise à améliorer la compréhension des activités en mettant en évidence les différents métiers exercés pour lesquels le lecteur disposera aisément de comparables connus sur le marché.

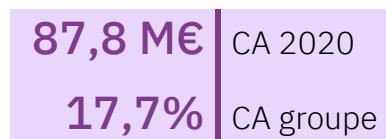
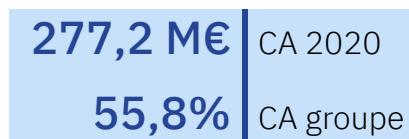


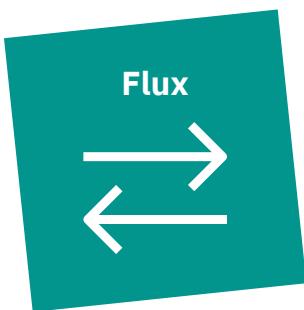
regroupe l'ensemble des offres logiciels du Groupe sous toutes leurs formes (licence, SaaS, services internet) ainsi que l'hébergement (agrément HDS) et l'infogérance. Cegedim cible l'assurance santé et prévoyance (France et Royaume-Uni), les professions paramédicales : kinésithérapeutes, infirmiers, orthophonistes, orthoptistes, podologues, sages-femmes... (France), les directions des ressources humaines (France), les pharmacies indépendantes, groupements ou chaînes de pharmacies (France, Roumanie et Royaume-Uni), les médecins et centres de santé (France, Royaume-Uni, Belgique, Espagne, Italie).



regroupe les activités :

- Données pour les autorités de santé, les professionnels de santé, les chercheurs, l'industrie de santé et ses partenaires en France, Italie, Allemagne, Espagne, Roumanie et Royaume-Uni;
- Communication en pharmacie et parapharmacie d'enseigne en France sous format papier et numérique;
- Marketing digital auprès des médecins;
- Distribution de produits de santé.





regroupe les activités de gestion du tiers payant (France), de dématérialisation de processus et factures, d'archivage à valeur probante et d'EDI (France, Royaume-Uni, Allemagne). Pour cette activité Cegedim dispose de centres de services en France, Roumanie et Maroc.

regroupe les activités de Business Process Outsourcing en France pour le compte des assureurs complémentaires santé (entre autre gestion des remboursements) et institutions de prévoyance, et départements RH. Pour cette activité Cegedim dispose de centres de services en France et en Roumanie.

79,4 M€ CA 2020
16,0% CA groupe

48,9 M€ CA 2020
9,8% CA groupe

Note : Cette ventilation par division présentée ci-dessus est la typologie privilégiée dans les communiqués de presse et les présentations financières de Cegedim. Il existe encore une cinquième division (non présentée ci-dessus), il s'agit de la division **Corporate et autres** et regroupe à la fois des activités inhérentes au statut de tête de Groupe coté, et des activités de support aux divisions du Groupe. Cette division a réalisé en 2020 un chiffre d'affaires de 3,7 millions d'euros (0,7% du chiffre d'affaires consolidé du groupe).

1.2 Cegedim SRH

1.2.1 Présentation

Cegedim SRH, filiale du Groupe Cegedim, est l'un des leaders français des solutions et services de gestion de la paie et des ressources humaines. Acteur majeur du Cloud RH et des services externalisés, Cegedim SRH dispose d'une expertise de plus de 25 ans dans ce domaine accumulée et reflétée dans sa plateforme **TEAMS^{RH}** qui offre une large couverture fonctionnelle de solutions dédiées aux Ressources Humaines.

1 PRÉSENTATION DE L'ORGANISME D'ACCUEIL



Figure 1.5. Logo Cegedim SRH

La société compte parmi ses clients des entreprises nationales et internationales, de tous secteurs d'activité, issues des grands comptes et du mid-market.



Figure 1.6. Clients Cegedim SRH

1.2.2 Domaines fonctionnels

Capitalisant sur l'apport des nouvelles technologies, Cegedim SRH s'appuie sur un investissement continu et une stratégie de différenciation pour réinventer les outils de gestion RH à travers son offre TEAMS^{RH}, un SIRH complet et modulaire de solutions et services en mode externalisé pour répondre aux besoins d'agilité, de fiabilité et de performance de ses clients. La plateforme TEAMS^{RH} est constituée d'un ensemble de modules tels que :

- paie et gestion administrative,
- portail RH collaboratif,
- gestion des temps et des activités,
- pilotage social,
- gestion des RH (Entretiens et Formations),
- processus RH dématérialisés intégrés dans la solution (signature électronique et coffres-forts numériques).

1 PRÉSENTATION DE L'ORGANISME D'ACCUEIL

Pour aider ses clients à améliorer la performance de leurs activités, Cegedim SRH propose des services de gestion de la paie et des RH selon quatre niveaux de prestations, en fonction du degré de responsabilité de chaque acteur :

SaaS +

abonnement aux services hébergés de TEAMS^{RH} incluant la maintenance corrective et les mises à jour légales et conventionnelles de l'application.

Processing

externalisation partielle avec pilotage de la relation client, le suivi du traitement de la paie, des opérations d'exploitation, de production et d'édition.

BPO on demand

service d'externalisation de la paie évolutif et sur-mesure.

BPO

externalisation complète avec prise en charge de l'ensemble des opérations de traitement de la paie (accréditation ISAE 3402).

1 PRÉSENTATION DE L'ORGANISME D'ACCUEIL

1.2.3 Agences et centres de services

Cegedim SRH dispose de sites de proximité à Paris, Lyon, Lille, Nantes et Toulouse, associés à des centres de développement nearshore à Montargis, Vichy et Genève, et offshore à Bucarest et Rabat (voir figure 1.7).

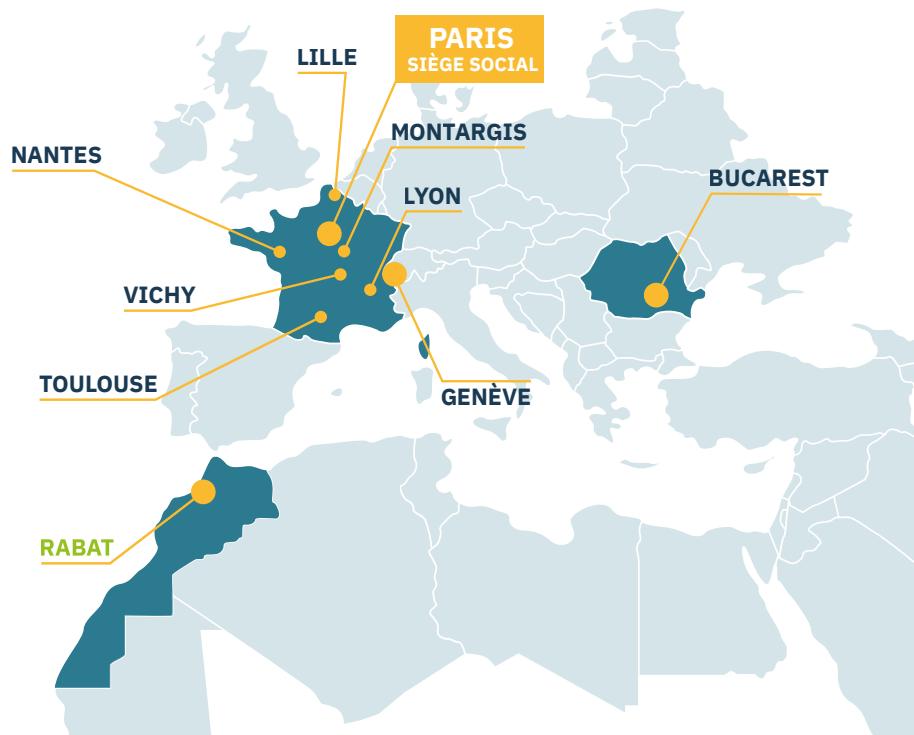


Figure 1.7. Les différents sites SRH

1.2.4 Organigramme SRH

Cegedim SRH dispose d'un organigramme qui montre sa structure générale ainsi que l'organisation de ses directions.

1 PRÉSENTATION DE L'ORGANISME D'ACCUEIL

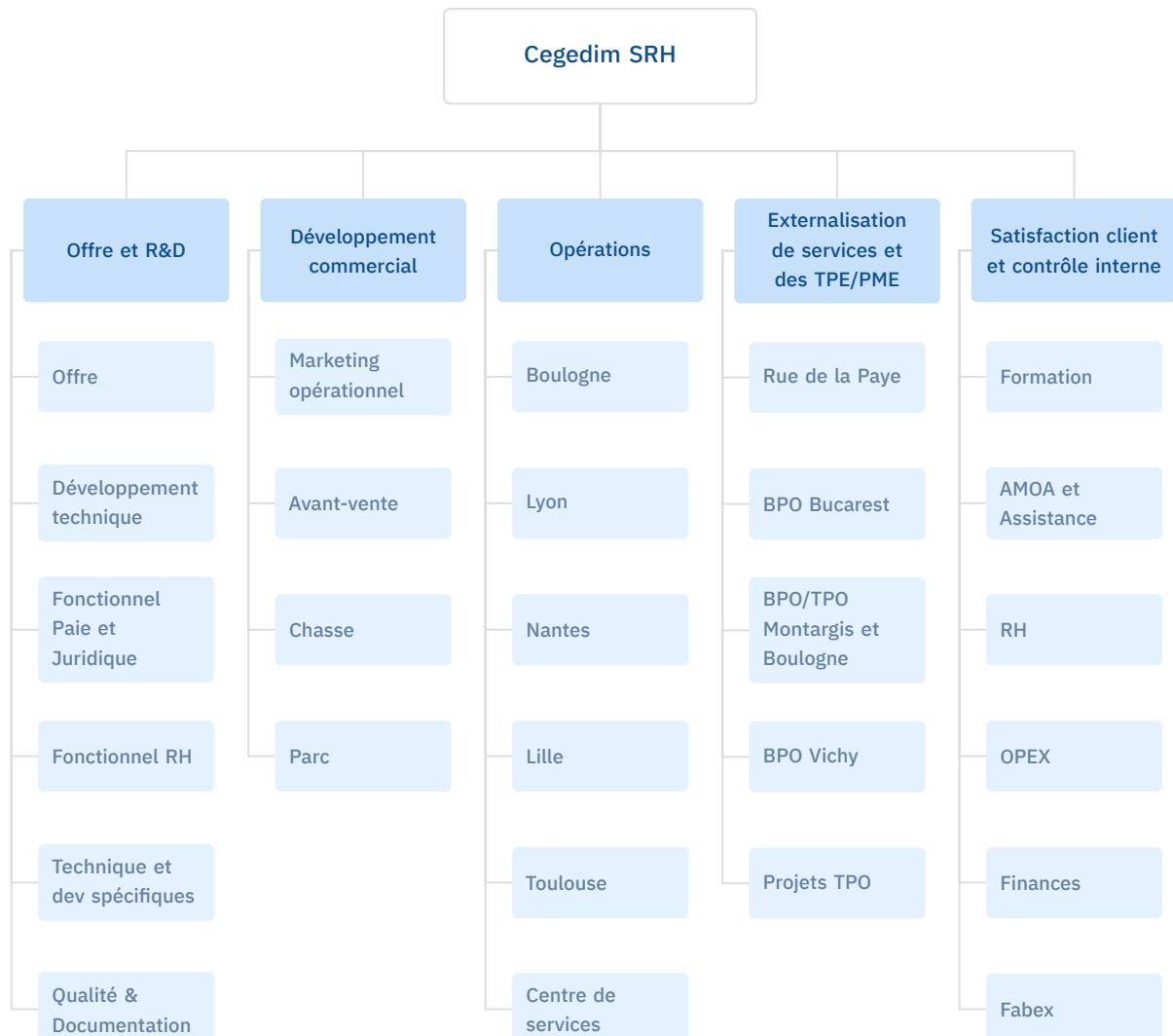


Figure 1.8. Organigramme des Directions Cegedim SRH au 1er janvier 2019

Conclusion

La présentation de l'organisation d'accueil est une étape préliminaire pour introduire l'environnement de notre projet. Dans le chapitre suivant, nous expliquerons en détail l'application sur laquelle portera le travail de refonte.

Chapitre 2

Présentation du projet

Introduction

Dans ce présent chapitre, nous nous articulerons sur l'étude et l'analyse du projet. Pour ce faire, nous débuterons par une étude préalable dans laquelle nous exposerons les fonctionnalités de la GED et ses avantages compte tenu de son implication dans le sujet, puis nous entamerons une étude du système existant, pour ensuite identifier les problèmes, les points à améliorer et les solutions envisagées.

2.1 Étude préliminaire

Avant d'entamer la discussion sur le sujet du stage, nous allons introduire quelques concepts et terminologie de la GED qui nous aideront à mieux comprendre le sujet par la suite.

2.1.1 Qu'est-ce que la GED?

Définition : La GED ou Gestion Électronique de Documents est ensemble de logiciels concourant à réaliser les diverses étapes de la chaîne de traitement d'un document : acquisition, restitution, diffusion [2].

La GED est donc un système informatique qui vise à gérer et organiser les documents numériques (souvent issus de l'entreprise). En outre, la GED permet la numérisation des documents papier, ainsi que la dématérialisation des processus métier associés.

La gestion électronique des documents utilise des outils et des fonctionnalités pour gérer toutes les étapes du cycle de vie d'un document numérique :

- Création ou acquisition du document (ex : numérisation) ;
- Stockage, indexation et organisation du document électronique ;
- Gestion de la sécurité des données du document électronique ;
- Recherche, consultation du document électronique et échange des informations ;
- Archivage électronique du document durant son délai de conservation.

2.1.2 Bénéfices d'une GED

D'un point de vue stratégique, la mise en place d'une solution de GED rend possible une rationalisation des flux d'informations et donc un gain de temps. Elle permet notamment de :

- Éviter la duplication des processus et ainsi réduire les coûts de traitement.

2 PRÉSENTATION DU PROJET

- Éviter la perte de documents.
- Trouver facilement et rapidement la bonne version d'un document.
- Uniformiser les pratiques documentaires.
- Partager des données avec un certain nombre de personnes autorisées.

D'un point de vue plus opérationnel et technique, la GED garantit :

- La pérennité des documents et de leur support.
- L'interopérabilité : les documents peuvent être accessibles sur différentes plates-formes et pour des usages divers.
- La sécurité : l'accès aux données est sécurisé, la solution s'adapte aux différents profils utilisateurs.
- La traçabilité : retrouver les actions effectuées (dépôts, restitutions, demandes de copies, etc.)

2.2 Étude de l'existant

2.2.1 Présentation du projet Arkevia

ARKEVIA est un service innovant et pratique de coffre-fort numérique développé par Cegedim SRH permettant aux salariés des entreprises adhérentes de recevoir et de conserver dans un espace sécurisé les bulletins de salaire et les communications institutionnelles en version électronique dans des conditions optimales de confidentialité et de sécurité. Arkevia permet également aux salariés de classer et de conserver leurs documents personnels importants tels que les pièces d'identité, les diplômes, les factures, etc.

Le coffre-fort Arkevia constitue un espace de stockage sécurisé pour les documents qui y sont déposés (EDI, EDI signé, PDF signé).

Il permet de garantir :

- **Intégrité** : L'intégrité des documents, au moyen d'une fonction de signature électronique.
- **Confidentialité** : La confidentialité des documents, au moyen d'une fonction de chiffrement de données.
- **Traçabilité** : La traçabilité des actions effectuées (dépôts, restitutions, demandes de copies, etc.)
- **Vocation probante** : Les solutions de traçabilité constituent de vraies preuves juridiques basées sur leur intégrité, leur exhaustivité et leur opposabilité en cas de litige.

2 PRÉSENTATION DU PROJET

Bénéfices pour le salarié :

- Accès illimité (24 h/7 j) depuis n'importe quel endroit grâce à une connexion Internet.
- Tous les bulletins de salaire déposés dans le coffre-fort ont la même valeur juridique que leur équivalent papier, même en cas de départ de l'entreprise.
- Espace personnel sécurisé de 1 Go dédié pour archiver les bulletins de paie et documents RH en version électronique pour une durée de conservation peut aller jusqu'à 50 ans des bulletins de salaire.
- Protection des documents importants contre le vol et la perte.
- Accès au coffre-fort sécurisé et garanti.

Bénéfices pour l'entreprise :

- Optimisation des processus RH d'édition et de distribution documentaire.
- Réduction des coûts de distribution.
- Image moderne et innovante de la DRH.
- Enrichissement de l'expérience du salarié.

2.2.2 Architecture globale d'Arkevia

Terminologie nécessaire à la compréhension du sujet :

Terminologie :

- **SignArchive** : application de la BU Cegedim e-business, hostée au datacenter de Boulogne et exploitée par l'équipe interne, ayant pour but le stockage et la signature de documents numériques dans des services à valeur légalement probante.
- **Arkevia** : application Web, décrite dans ce document, présentant une interface web publique consultable par un navigateur, dont le but est l'exploration et la manipulation de documents dans les coffres SignArchive.
- **Arkevia-SRH** : module d'extension pour Arkevia permettant la mise à disposition dans Arkevia de données (traductions, formulaires d'abonnement, conditions d'utilisation) issues du système d'information de SRH. Ce module permet la prise en charge des abonnements des utilisateurs d'Arkevia au système de dématérialisation des bulletins de paie émis par SRH.
- **SignArchive-Batcher** : module de traitement générique pour les applications souhaitant émettre des documents dans SignArchive, à des fins d'exploration par un utilisateur Arkevia. Le système d'information de SRH utilise SignArchive-Batcher pour placer les bulletins dans les coffres.

2.2.2.1 Généralités

Le développement du produit SignArchive offre à la BU Cegedim e-business, ses clients et partenaires, l'occasion de disposer d'un produit capable à la fois :

- de stocker des fichiers (de façon légalement probante si besoin) ;
- de les organiser par des métadonnées ;
- de fédérer des droits d'accès dynamiques autour de ces fichiers via une API standard (purement HTTP).

Au-delà de l'aspect juridique de ce logiciel d'archivage (exploité par les solutions de dématérialisation du groupe), le groupe dispose également d'un moteur de stockage, gestion, recherche et contrôle d'accès qui peut être mis à disposition de toute problématique orientée partage ou publication de document.

C'est à une de ces fonctions que s'attache le projet Arkevia, dont le périmètre fonctionnel est triple :

1. Fournir un site web portant une offre de coffre-fort numérique à destination du (grand) public et exposé via le produit SignArchive. Ce site web s'appelle Arkevia.
2. Offrir à des tiers la capacité de devenir « producteur de documents » pour le portail Arkevia, c'est-à-dire, définir les modalités techniques selon lesquelles un système d'information (interne ou externe) peut s'accrocher au portail Arkevia et à SignArchive, de façon à ce que les clients d'Arkevia puissent visualiser des documents produits par des tiers.
3. Implémenter un tel « tiers producteur » pour les besoins de SRH, permettant la dématérialisation des bulletins de paie émis par l'entreprise.

Le deuxième point définit clairement l'objectif d'extensibilité du système, ce qui détermine la plupart des choix de mise en œuvre effectués pour parvenir à la solution globale.

2.2.2.2 Présentation métier

La réalisation de la finalité du produit, à cause de son besoin de généréricité, est confiée à trois livrables distincts.

Note : La présence de ces trois produits est nécessaire au fonctionnement global de la solution Arkevia+SRH, mais Arkevia peut/aurait pu/pourrait tout à fait exister sans le module SRH, et réciproquement, le module SRH n'a pas besoin d'Arkevia pour pousser les fichiers dans les coffres SignArchive. D'autre part, le module SRH a été développé par Cegedim e-business à la demande de SRH, mais le produit est conçu de sorte qu'Arkevia puisse être implémenté par un producteur qui souhaite prendre en charge lui-même cette implémentation.

Ainsi, les fonctionnalités sont réparties comme suit :

Arkevia

Le site web Arkevia propose à des navigateurs web en provenance d'internet, la consultation de coffres-forts SignArchive dans une interface de navigation proche d'un explorateur. Son rôle premier est donc celui de « proxy » SignArchive (se connecter pour le compte de l'utilisateur sur SignArchive), et de couche de présentation. D'autre part, Arkevia permet à ses utilisateurs de « s'abonner » à des « producteurs » de documents. La gestion de ces abonnements (présentation du formulaire d'abonnement spécifique à un producteur, messages de notification du producteur, ...) nécessite qu'Arkevia puisse communiquer avec le producteur pour échanger images, documents, données, etc. Cette communication doit respecter des principes techniques établis dans le cahier des charges d'implémentation d'Arkevia.

Arkevia-SRH

Ce livrable constitue le module d'abonnement du producteur SRH. Son rôle est :

1. De gérer la liste des personnes autorisées à s'abonner au producteur SRH; et d'en rendre compte à SRH.
2. De fournir à Arkevia les éléments nécessaires à la présentation et la prise en compte des données d'abonnement (champs de formulaire à présenter pour s'inscrire, libellés traduits des messages affichés par **Arkevia** pour le compte de SRH, etc.).

Il est donc constitué à la fois d'une chaîne de traitement chargée de la prise en compte des données d'inscription publiées par SRH, et de web services de publication de données.

SignArchive-Batcher

Ce livrable est une chaîne de traitement publant, pour le compte de SRH, les bulletins de salaire à dématérialiser dans **SignArchive**, selon le format admissible pour **Arkevia**.

2.2.2.3 Vue d'ensemble

Au global, on a une répartition des tâches sous la forme schématique suivante (voir figure 2.1) :

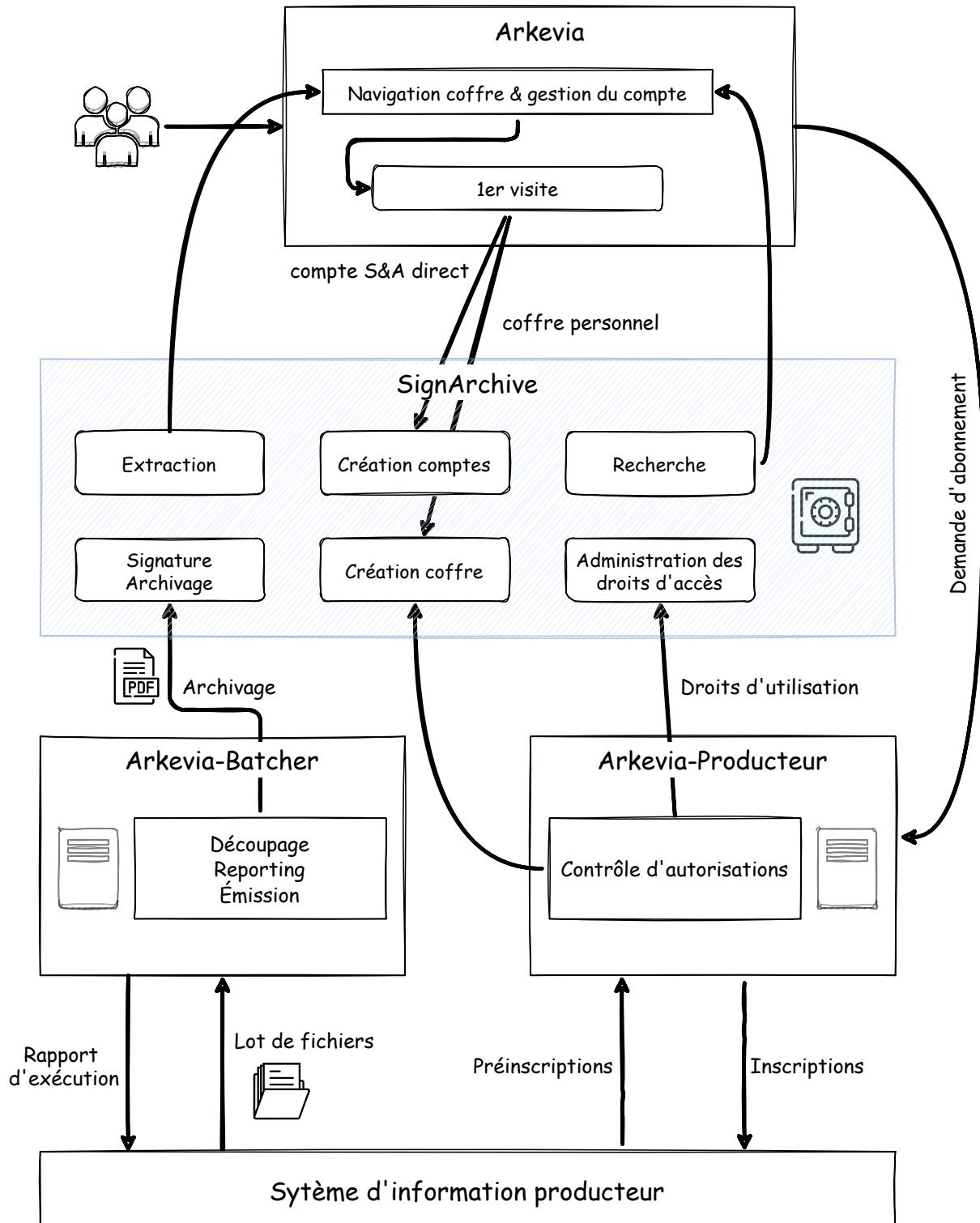


Figure 2.1. Répartition des tâches dans l'écosystème Arkevia

1. **Arkevia** se connecte à **SignArchive** pour en afficher le contenu, indépendamment du fait que SRH publie ou non des documents pour cet utilisateur. Cependant, si l'utilisateur souhaite s'abonner au service de production SRH, alors, Arkevia pousse la demande via un web service au module approprié, **Arkevia-SRH**.
2. **Arkevia-Producteur (Arkevia-SRH)** peut prendre en compte ou non la demande, selon les autorisations qui lui ont été publiées par SRH directement, via le fichier des préinscriptions. Chaque inscription ou résiliation est remontée à SRH via le même canal que le fichier des préinscriptions.
3. En conséquence de quoi, lors de la fabrication des bulletins de paie, le module de gestion interne à SRH peut constituer une archive des bulletins à archiver conforme aux utilisateurs Arkevia inscrits. Ce fichier est poussé au module **SignArchive-Batcher** qui pousse les fichiers dans **SignArchive**.

2.3 Problèmes et axes d'amélioration

2.3.1 Défis technologiques

Le backend de l'application Arkevia est développé principalement avec la technologie Spring 3.x. Ce framework a été publié pour la première fois en 2004 et a subi de nombreuses révisions depuis lors :

- Spring 2.0 a fourni des espaces de noms XML et le support d'AspectJ ;
- Spring 2.5 a adopté la configuration basée sur les annotations ;
- Spring 3.0 a introduit une solide base Java 5+ à travers le socle du framework et des fonctionnalités telles que le modèle @Configuration basé sur Java.

Aujourd'hui, le projet Spring et le développement Java ont tous deux évolué de manière significative, incluant à la fois des correctifs de sécurité et de nouvelles fonctionnalités et améliorations, faisant de la migration une tâche assez importante.

À cet égard, il a été convenu de réaliser une étude comparative sur l'impact d'une telle migration du framework Spring ; le résultat de cette étude fera l'objet d'un travail de développement sur la migration du socle technique d'Arkevia.

2.3.2 Refactorisation et nettoyage du code

Comme mentionné dans l'introduction, le portail Arkevia a subi plusieurs changements et évolutions au cours des dernières années, rendant le code trop compliqué, redondant et imprégné de mauvaises odeurs (code smells), d'où la nécessité de recourir à un processus de refactoring et d'amélioration du code tout en préservant les fonctionnalités existantes. Le but est de transformer un code inefficace et compliqué en un code plus efficace, de préférence plus simple et plus clair.

2.3.3 Mécanisme de Notification

Le mécanisme d'envoi de notifications est responsable de notifier un utilisateur par mail, lorsqu'il y a un nouveau dépôt de document (Bulletin de paie, note de frais, etc.) sur son coffre-fort par la société à laquelle il appartient.

Le module de gestion de l'envoi des notifications est intégré à l'application mère Arkevia. Le processus d'envoi est planifié pour être déclenché toutes les deux heures, et à chaque exécution, l'application Arkevia sollicite le service chargé d'interroger l'API SignArchive pour récupérer la liste des documents déposés et celle des utilisateurs associés afin de préluder à l'envoi des notifications. Le mécanisme actuel n'est pas capable à traiter la volumétrie de documents en production. En effet, ce dernier souffre de plusieurs failles et notamment quand le nombre de documents déposés dans SignArchive est élevé, le système se met à défaillir et à dysfonctionner, ce qui entraîne que certains utilisateurs ne reçoivent plus les notifications sur la présence des documents déposés dans leur coffre-fort ou les reçoivent plusieurs fois pour un même document, etc.

2.4 Objectif de la refonte

Après avoir introduit l'application existante et repéré les différents points d'amélioration et les problèmes engendrés au sein de cette dernière, nous procéderons à une refonte globale de l'application dont l'objectif sera de revoir, corriger et améliorer ces aspects recensés afin notamment de rendre l'application plus performante, facile à maintenir et à faire évoluer.

Conclusion

Dans ce chapitre, nous avons décrit de manière didactique l'application qui fera l'objet de la refonte, et ce, afin d'exposer le contexte global de notre projet en dégageant les problématiques et les différents axes d'amélioration. Enfin, nous avons clôturé ce chapitre en exprimant l'objectif de la refonte auquel s'articule le sujet de l'étude.

Chapitre 3

Méthodologie de gestion de projets

Introduction

Dans ce chapitre, nous aborderons les aspects de la gestion de projet, de la composition des équipes, des outils et des méthodes de développement de logiciels adoptés par Cegedim SRH.

3.1 Étude préliminaire

3.1.1 Origine du génie logiciel

Définition : Le génie logiciel est la science de l'ingénieur qui s'intéresse aux procédés scientifiques de construction et d'entretien des logiciels, et bien évidemment à la « matière » même de cette construction : d'abord les programmes eux-mêmes, les fichiers et bases de données, les scripts de paramétrage nécessaires à l'exécution du programme, puis tout ce qui gravite autour (spécification de besoins et exigences des futurs utilisateurs, conception, tests, documentation pour les mainteneurs, le support technique et les usagers)[6].

À la fin des années 50, l'informatique est devenue de plus en plus populaire et s'est étendue à d'autres disciplines. Ceci en raison de l'impulsion des grands projets spatiaux et militaires, grands consommateurs de logiciels et imposant des exigences de qualité et de sûreté de fonctionnement et du fait que les informaticiens du génie logiciel étaient plus enclins et plus à même de développer des outils logiciels d'aide à la conduite des activités.

En octobre 1968, l'OTAN a organisé une première conférence, à Partenkirchen en Allemagne, sur l'industrialisation de l'élaboration du logiciel. C'est à cette occasion qu'est forgée l'expression « Software Engineering » pour donner une tournure délibérément industrielle au propos[7].

À la fin des années 60, les grands systèmes commerciaux ont démontré qu'il était difficile d'adapter à grande échelle les principes qui avaient été adéquats jusque là. Les grands projets dépassaient les budgets et les délais, ce fût alors « the software crisis » la crise du logiciel[1].

La conférence de l'OTAN a été immédiatement suivie par de nombreuses autres

3 MÉTHODOLOGIE DE GESTION DE PROJETS

conférences portant sur des thèmes liés au génie logiciel, tels que la fiabilité des logiciels, le test du logiciel, la spécification du logiciel, la maintenance du logiciel, etc.

En 1975 s'est tenue la première conférence véritablement dédiée à l'ensemble du génie logiciel, accompagnée d'une exposition de premiers outils, notamment pour l'aide au test. Depuis, l'IEEE tient une rencontre annuelle internationale. Cette première conférence a correspondu à la création, au sein de l'IEEE, d'un comité technique dédié au génie logiciel, animateur d'une série de conférences et ateliers récurrents et éditeur de deux revues dédiées au génie logiciel[7].

3.1.2 Comparaison des différentes méthodes de développement

3.1.2.1 Les approches traditionnelles

Dans une approche traditionnelle, le cycle de vie d'un logiciel passe par quatre phases principales : **initiation, développement, déploiement et exploitation**.

- **L'initiation** vise à déterminer la mission du système et à effectuer des travaux exploratoires permettant de vérifier la pertinence du projet.
- **Le développement** est la phase qui prend en charge la réalisation du logiciel. Elle précise les besoins, réalise le produit et valide son fonctionnement.
- **Le déploiement** rend le produit accessible à ses utilisateurs, par la mise en service du logiciel dans son environnement de production.
- **L'exploitation** est la période de vie utile du produit au cours de laquelle des améliorations peuvent être apportées au produit.

3.1.2.2 Les méthodes agiles

Une méthode agile est une approche itérative et incrémentale, qui est menée dans un esprit collaboratif, avec juste ce qu'il faut de formalisme. Elle génère un produit de haute qualité tout en prenant en compte l'évolution des besoins des clients[8].

Le Manifeste Agile déclare quatre valeurs dans toute approche agile. Chaque méthode adopte ensuite sa propre terminologie et recommande un certain nombre de pratiques. Les quatre valeurs du Manifeste sont[4, 8] :

- Les individus et leurs interactions avant les processus et les outils.
- Des fonctionnalités opérationnelles avant la documentation.
- Collaboration avec le client plutôt que contractualisation des relations.
- Acceptation du changement plutôt que conformité aux plans.

3.1.2.3 Synthèse

Bien que les approches traditionnelles présentent divers avantages, elles ont aussi certaines limites, en particulier dans les environnements évolutifs. En effet, les

3 MÉTHODOLOGIE DE GESTION DE PROJETS

environnements évolutifs nécessitent des adaptations constantes. Avec une planification rigoureuse, il est difficile d'apporter des changements à cause de leurs impacts, et plus un changement est fait tard, plus il est coûteux. Cette situation tardive exige de modifier l'enchaînement des étapes antérieures et seuls les changements les plus importants seront opérés.

Selon une nouvelle étude menée par Organize Agile[5] auprès de professionnels de 19 pays, près de la moitié des organisations utilisent des méthodes agiles depuis trois ans ou plus. Ces entreprises utilisent principalement Agile comme méthodologie pour leurs programmes de réforme - connus sous le nom de transformations agiles. Il s'agit de grands changements organisationnels qui adoptent le travail agile au sein de petites équipes multidisciplinaires qui s'attachent à fournir des résultats de manière rapide, expérimentale et itérative, ce qui révèle l'impact positif de ces méthodes sur les flux de travail des projets informatiques à l'échelle mondiale. Au Maroc, selon un rapport d'étude mené en 2011 impliquant 48 organisations participantes représentant les secteurs privé et public a révélé que le taux de satisfaction de la maîtrise d'ouvrage est plus élevé pour les DSI adoptant une des méthodes agiles ce qui démontre leur pragmatisme et leur efficacité par rapport aux méthodes traditionnelles en cascade[9].

Les approches agiles et traditionnelles convergent et divergent sur différents aspects. L'expérience de l'équipe peut faire diminuer les risques et contribuer à minimiser les exigences de suivi. Les approbations et le manque d'autonomie des équipes peuvent ralentir le projet. Une organisation peu hiérarchisée, ayant une culture collégiale, intégrera difficilement une structure traditionnelle.

3.2 Conduite de projet

3.2.1 Modèle de livraison

Aujourd'hui, Cegedim adopte une approche de livraison agile que tout projet est censé suivre. Le schéma ci-dessous montre les différentes phases de ce modèle (voir figure 3.1) :

3 MÉTHODOLOGIE DE GESTION DE PROJETS

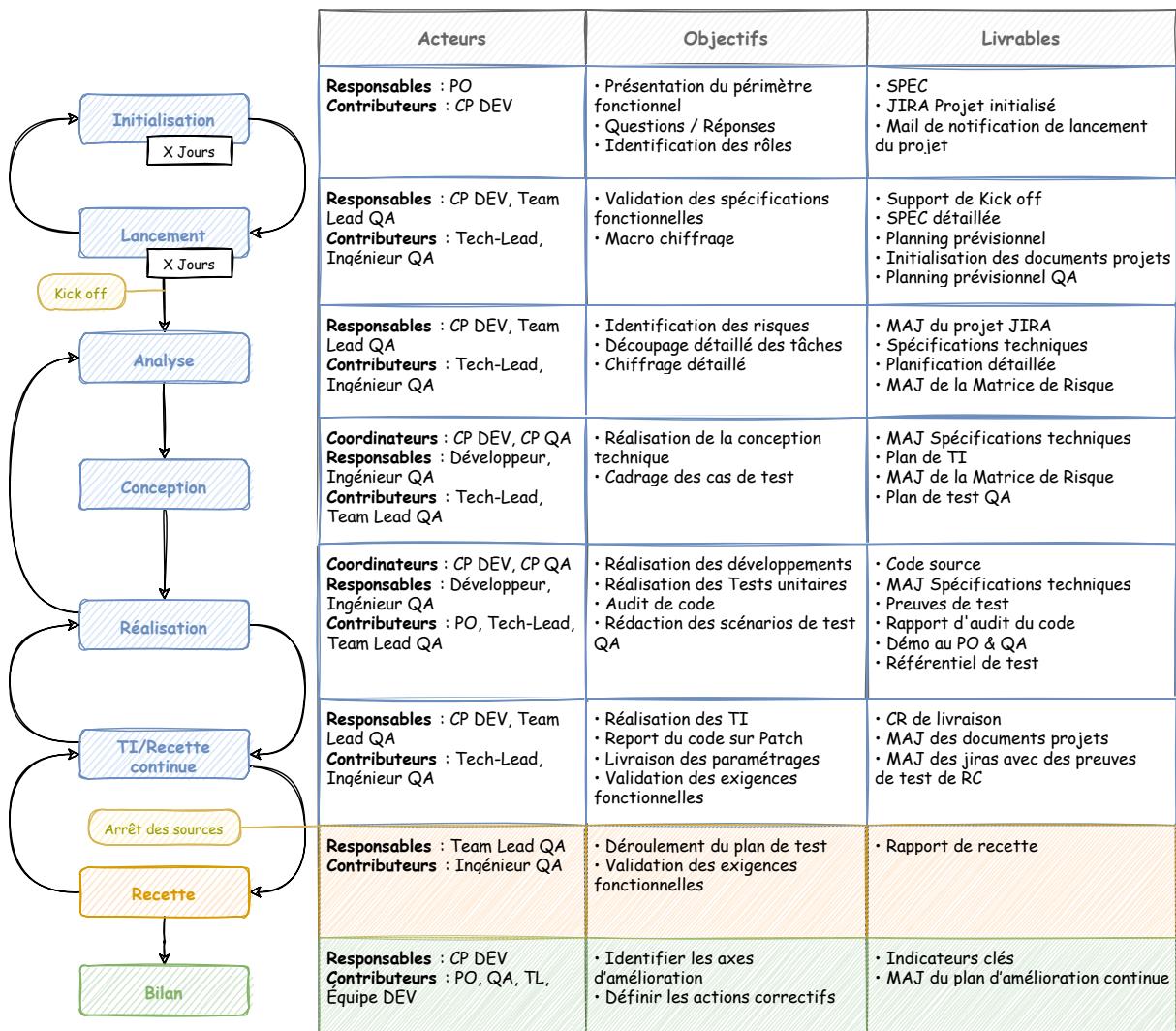


Figure 3.1. Modèle de livraison

Les acteurs impliqués dans le processus de livraison peuvent être répartis en deux grandes divisions (voir figure 3.2, tableaux A.1 et 3.2) :

3 MÉTHODOLOGIE DE GESTION DE PROJETS

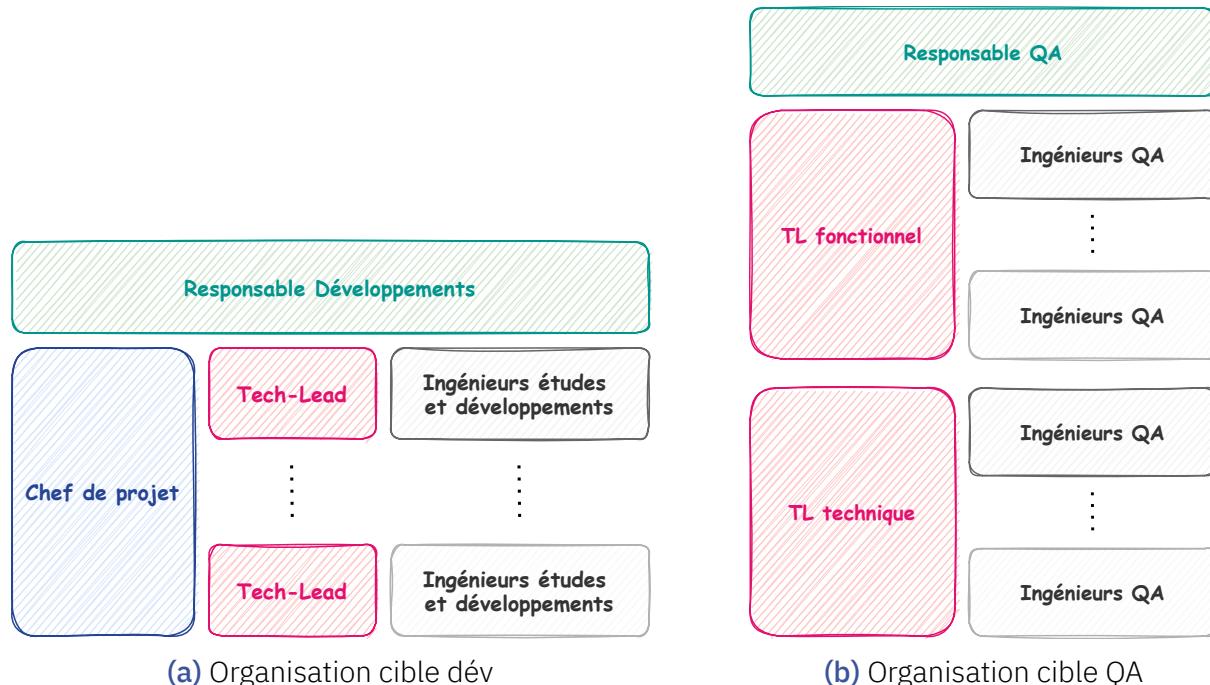


Figure 3.2. Modèle organisationnel visé pour les équipes de développement et de QA

Le tableau suivant (A.1) décrit les différentes activités et tâches confiées aux membres de l'équipe de développement impliqués dans le processus de livraison :

Table 3.1 : Responsabilités et missions des différents acteurs de l'équipe de dév

Équipe de développement

Responsable Développements

- Accompagner le chef de projet dans la gestion des projets.
- Produire des indicateurs sur l'activité de développement.
- Suivi des risques et gestion des alertes.
- Travailler en collaboration avec les autres responsables de division.

Chef de projet

- Valider les spécifications fonctionnelles.
- Assurer le suivi du processus de livraison et des indicateurs clés.
- Élaboration du plan de charge et planification des équipes.
- Garant des engagements (Qualité, Délai, Coût).

Suite à la page suivante

3 MÉTHODOLOGIE DE GESTION DE PROJETS

Table 3.1 : Responsabilités et missions des différents acteurs de l'équipe de dév (Suite)

Tech-Lead
<ul style="list-style-type: none">• Responsable de la qualité technique (Audit, conception, etc.).• Responsable des bonnes pratiques de développement.• Encadrement et assistance technique.• Participer aux développements.• Assister le chef de projet pour les estimations de charge.
Ingénieur étude et développement
<ul style="list-style-type: none">• Corriger les anomalies et développer de nouveaux modules.• Respecter les bonnes pratiques de développement.• Participer à la définition de la couverture de tests techniques.• Rédiger des documents techniques.

Le tableau suivant (3.2) décrit les différentes activités et tâches assignées aux membres de l'équipe d'assurance qualité (QA) participant au processus de livraison :

Table 3.2 : Responsabilités et missions des différents acteurs de l'équipe QA

Équipe QA
Responsable QA
<ul style="list-style-type: none">• Établir et piloter une stratégie de test.• Accompagner le Team Lead dans la gestion des projets.• Contribuer à l'amélioration des processus de test.• Produire des indicateurs sur l'activité de testing.• Suivi des risques et gestion des alertes.• Travailler en collaboration avec les autres responsables de division.

Suite à la page suivante

3 MÉTHODOLOGIE DE GESTION DE PROJETS

Table 3.2 : Responsabilités et missions des différents acteurs de l'équipe QA (Suite)

TL fonctionnel/technique
<ul style="list-style-type: none">• Responsable du suivi du processus de livraison et des indicateurs clés.• Élaboration du plan de charge et la planification des équipes.• Accompagner et suivre les testeurs dans la mise en place des bonnes pratiques et des outils.• Accompagner les ingénieurs QA dans l'élaboration des plans de test.• Réaliser le bilan des activités.• Accompagner l'intégration des nouveaux arrivants et veiller à la montée en compétence des équipes.• Participer aux activités de test.
Ingénieur QA
<ul style="list-style-type: none">• Formaliser des scénarios de test fonctionnels et automatisés.• Valider et vérifier le développement de l'application.• Respecter les bonnes pratiques de testing.• Rédiger des documents fonctionnels.

3.2.2 Déroulement de projets - Méthode SCRUM

Depuis 2019, les projets de R&D ont progressivement passé en mode Agile. Ils utilisent la partie Agile de JIRA (catégorie Software). Le mode Agile permet de gérer les projets en mode Scrum ou Kanban. Pour l'instant, Cegedim SRH utilise le mode Scrum.

3.2.2.1 L'équipe Agile

L'équipe Agile est une équipe qui entend être totalement autonome. Elle rassemble toutes les compétences nécessaires pour faire évoluer le produit. Le maître mot d'une équipe agile est la coopération. En effet, les membres de l'équipe ne travaillent pas séparément sur les fonctionnalités qui leur sont assignées, mais ensemble, y compris pour l'identification, la définition, la réalisation et les tests des fonctionnalités.

Les capacités d'écoute et d'entraide de l'équipe facilitent la montée en compétence. Chaque membre devient petit à petit plus ou moins polyvalent et en capacité d'aider les autres dans la réalisation des différentes fonctionnalités. L'équipe devient donc de plus en plus performante et efficace, tout en améliorant également le ressenti et les conditions de travail de chacun.

Une équipe Agile est en perpétuelle progression et évolution.

Il n'y a pas de composition type pour une équipe Agile. Celle-ci dépend totalement du produit à réaliser. Certains rôles clefs sont néanmoins nécessaires au bon fonctionnement de l'équipe. La composition d'une équipe Agile au sein de la R&D SRH suivra à minima le modèle suivant :

- **Equipe Agile R&D**

- 1 Product Owner;
- 1 Scrum Master (optionnel mais conseillé) ;
- X développeurs (il est recommandé d'avoir +2) ;
- 1 QA (optionnel) ;
- X testeur(s)

Bien entendu, selon les produits, cette structure sera susceptible d'évoluer.

3.2.2.2 Le Sprint

Le Sprint agile est le cœur de la méthode SCRUM. Tous les développements sont réalisés de manière incrémentale au sein des Sprints. Un périmètre de développement - l'objectif du Sprint - est défini au début du Sprint lors du **Sprint Planning** avec la liste des **User Stories** à traiter pendant le Sprint.

À la conclusion d'un Sprint, le bilan est réalisé lors du **Sprint Review**, puis un nouveau cycle démarre avec un nouveau Sprint. La périodicité d'un Sprint dépendra de l'équipe Agile, mais sera généralement entre 2 et 4 semaines.

3.2.2.3 Les Rituels

Les rituels (ou cérémonies) sont des réunions de travail et de suivi qui viennent rythmer un Sprint. Chaque réunion joue un rôle précis dans le Sprint et correspond à une temporalité particulière.

- **Sprint Planning** : Le Sprint Planning est la réunion de lancement d'un Sprint. Au début de chaque Sprint, le Product Owner présente les User Stories qu'il souhaite-rait intégrer au Sprint. Les User Stories sont examinées une par une dans l'ordre de priorité du Backlog jusqu'à ce que le Sprint soit complet en termes de vitesse. L'examen d'une User Story suit les étapes suivantes :

- Lecture de la User Story par le Product Owner.
- Questions des développeurs (optionnel).
- Poker Planning : les développeurs estiment le temps nécessaire pour le développement de la User Story en votant chacun pour une durée. Pour passer à l'étape suivante, il doit y avoir un consensus sur l'estimation. Si tel n'est

3 MÉTHODOLOGIE DE GESTION DE PROJETS

pas le cas, les développeurs ayant donné les estimations extrêmes doivent s'expliquer et un nouveau vote est réalisé.

- Intégration de la User Story au Sprint : passage en "A faire" dans le Board de l'équipe.

Note : Le Sprint Planning a lieu impérativement le premier jour du Sprint, de préférence le matin afin d'optimiser un maximum le temps de travail de l'équipe sur le Sprint.

Remarque : À l'étape "Questions", si les développeurs ne comprennent pas ce qu'il faut faire, la User Story est replacée dans le Backlog en vue d'être détaillée plus avant par le Product Owner. De même, lors du "Poker Planning", si ces derniers n'arrivent pas à se mettre d'accord sur une estimation de temps commune.

- **Daily Stand-Up Meeting :** Le Daily Stand-up meeting est une réunion quotidienne très courte (entre 5 et 15 minutes) à heure et lieu fixes qui rassemble l'ensemble des membres de l'équipe Agile où chaque membre de l'équipe prend la parole à tour de rôle. Chacun doit expliquer ce qu'il a fait depuis l'itération précédente du Stand-up meeting ou le Sprint Planning et ce qu'il prévoit faire jusqu'au prochain Stand-up meeting.

Remarque : Le temps de parole de chacun ne doit généralement pas dépasser deux minutes.

Le Stand-up meeting a pour but de favoriser la circulation des informations au sein de l'équipe. Elle permet à l'ensemble de l'équipe d'avoir une vue complète de l'avancée des tâches et d'identifier d'éventuels points de blocages. Ces échanges dynamiques contribuent également à la cohésion et l'implication de l'équipe.

Attention :

- La tenue du Stand-up meeting n'est pas facultative.
- Si un sujet est susceptible de déborder, il doit être traité en dehors du cadre du Stand-up meeting avec les membres concernés.

- **Sprint Retrospective :** La Sprint Retrospective est la réunion d'analyse du déroulement du Sprint par l'équipe. Elle est positionnée à la fin de chaque sprint, après le Sprint Review. L'idée pour l'équipe est de capitaliser sur le vécu du Sprint écoulé pour adapter son organisation dans le but de renforcer son efficacité. C'est un élément clef dans le processus de progression et d'apprentissage de l'équipe.

3 MÉTHODOLOGIE DE GESTION DE PROJETS

L'équipe doit définir un plan d'action et d'amélioration à partir des constats et idées remontés pendant la réunion.

- **Sprint Review** : Le Sprint Review est une réunion qui se situe en toute fin d'un Sprint juste avant le Sprint Retrospective. Elle a pour but de présenter le travail réalisé durant le Sprint courant. Le but est d'obtenir un maximum de retours sur le réalisé afin d'assurer qu'il est bien en accord avec les attentes.

Si l'avancée le permet, elle s'agrémente généralement en fin de Sprint Review d'une démonstration des nouveautés. Si tel n'est pas le cas, la démonstration peut être effectuer séparément du Sprint Review.

Note : Cette réunion inclut non seulement l'équipe Agile mais aussi les parties prenantes et les décideurs.

3.2.3 Processus de développement

Compte tenu de la diversité des applications développées par Cegedim SRH et afin de mener à bien la mise en place des solutions, cette dernière adopte un processus de développement où tout projet est amené à suivre. Je ne ferai référence dans cette partie que sur un périmètre limité de ce processus, et sur lequel je suis intervenu à collaborer et échanger.

- **Analyse fonctionnelle et définition des objectifs** : Lors de cette phase préalable au démarrage du projet, les parties prenantes définissent ensemble :
 - les objectifs et la portée du projet,
 - les livrables attendus,
 - les délais souhaités,
 - le degré de souplesse qui pourra être accordé.
- **Étude de faisabilité et formalisation des spécifications techniques** : Une étude de faisabilité peut être menée afin de cerner les contraintes susceptibles de peser sur la mise en œuvre du projet. Ensuite, les spécifications techniques sont formalisées, faisant état des méthodes, processus et technologies qui seront utilisés pour répondre aux contraintes du projet.
- **Découpage et chiffrage** : Il s'agit d'établir la liste des tâches en associant les besoins et les coûts correspondants, tout en incluant les sous-tâches et les tâches induites par la réalisation afin de chiffrer au plus juste le projet.
- **Planification** : La planification vise à ordonner les tâches et à indiquer leur enchaînement logique en tenant compte des ressources disponibles et de leur charge de travail maximale.
- **Codage** : La phase de codage, également appelée programmation, consiste à traduire en code les fonctionnalités et les exigences techniques préalablement

définies.

- **Tests unitaires** : Le concept de test unitaire n'est pas un élément nouveau. Depuis les prémisses de l'informatique, les tests font partie de l'activité quotidienne d'un développeur. Ce qui est nouveau aujourd'hui, c'est que cette activité, notamment les tests unitaires, est placée au cœur du processus de conception. En effet, un test unitaire permet de valider la conformité de chaque composant logiciel pris comme une unité par rapport à sa spécification détaillée. Autrement dit, un scénario de test unitaire ressemble à une expérience scientifique dans laquelle une hypothèse est examinée en fonction de trois éléments clés :
 1. Les données en entrée.
 2. L'objet à tester.
 3. Les observations attendues.
- **Audit de code** : L'étape d'audit de code permet de s'assurer de la qualité du codage en vérifiant que chaque module ou sous-ensemble de la solution informatique est conforme aux règles et bonnes pratiques de développement logiciel. Dans le cadre de ce sujet, nous nous référerons plus particulièrement aux règles établies par Cegedim SRH (voir la checklist A.2 en annexe) dans le but de systématiser les bonnes pratiques et d'éviter les erreurs classiques de développement au sein des équipes de dév.
- **Recette** La phase de recette est le processus de validation par l'équipe de validation et acceptance (QA) de la conformité des livrables avec les spécifications initiales.
- **Documentation** : À l'issue de la recette, une documentation de projet est produite afin de rassembler les informations nécessaires à l'utilisation de la solution informatique et en vue de ses développements ultérieurs.
- **Déploiement** : Une fois le projet qualifié, la solution informatique peut être déployée : il s'agit de la livraison du produit final et de sa mise en service.

3.2.4 Gestion du workflow git

Le workflow de travail pour les branches GIT choisi au niveau de Cegedim SRH est Gitflow. Git Flow est un modèle de dépôt git permettant d'améliorer les processus de développement et de mise en production d'un projet.

Gitflow sépare sur des branches isolées le code en cours de développement et le code validé et testé. Pour cela, il s'appuie sur la création de plusieurs branches dont le cycle de vie est bien défini. Voici une table contenant leurs noms, leurs cycles de vie et leurs fonctions (voir la table 3.3) :

3 MÉTHODOLOGIE DE GESTION DE PROJETS

Table 3.3 : Présentation des différentes branches définies sur gitflow

Branche	Nombre	Branche d'origine	Durée de vie	Fonction
master	Unique		Permanente	Code stable, testé et validé potentiellement éligible pour une MEP (Mise En Production)
feature	Plusieurs	develop	Développe-ment d'une fonctionnalité	Code en cours de développement destiné à réaliser une fonctionnalité à intégrer dans la prochaine version de l'application.
develop	Unique	master	Permanente	Code de la prochaine version de l'application. Une fois que le développement d'une fonctionnalité (feature) est fini, le code est fusionné sur cette branche.
release	Unique	develop	Recette	Branche sur laquelle on corrigera les bugs détectés pendant la phase de recette.
hotfix	Aucune / Plusieurs	master	Correction d'un bug	Branche où on fait les corrections des bugs sur le code présent sur la branche master (production).

Voici un schéma présentant l'organisation du dépôt git ainsi que les différentes interactions qu'il peut y avoir entre chaque branche (voir la figure 3.3) :

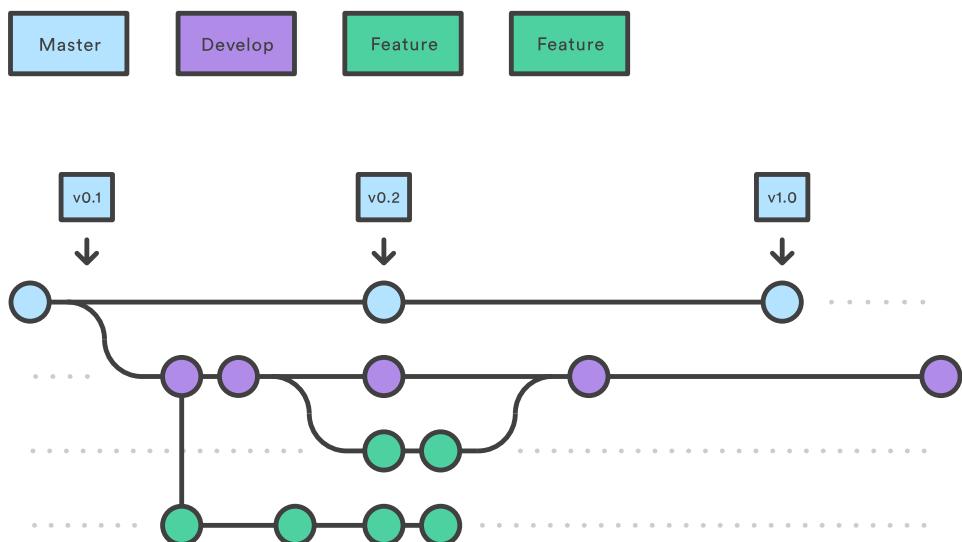


Figure 3.3. Schéma illustrant l'interaction entre les différentes branches du flux de travail gitflow

3.3 Planification et suivi du projet

La planification est parmi les phases d'avant-projet. Elle consiste non seulement à délimiter le périmètre temporel du projet, mais aussi à prévoir le déroulement des activités tout au long de la période allouée au stage.

3.3.1 Diagramme de Gantt

La figure suivante détaille la planification prévisionnelle du projet (voir figure 3.4) :

3 MÉTHODOLOGIE DE GESTION DE PROJETS

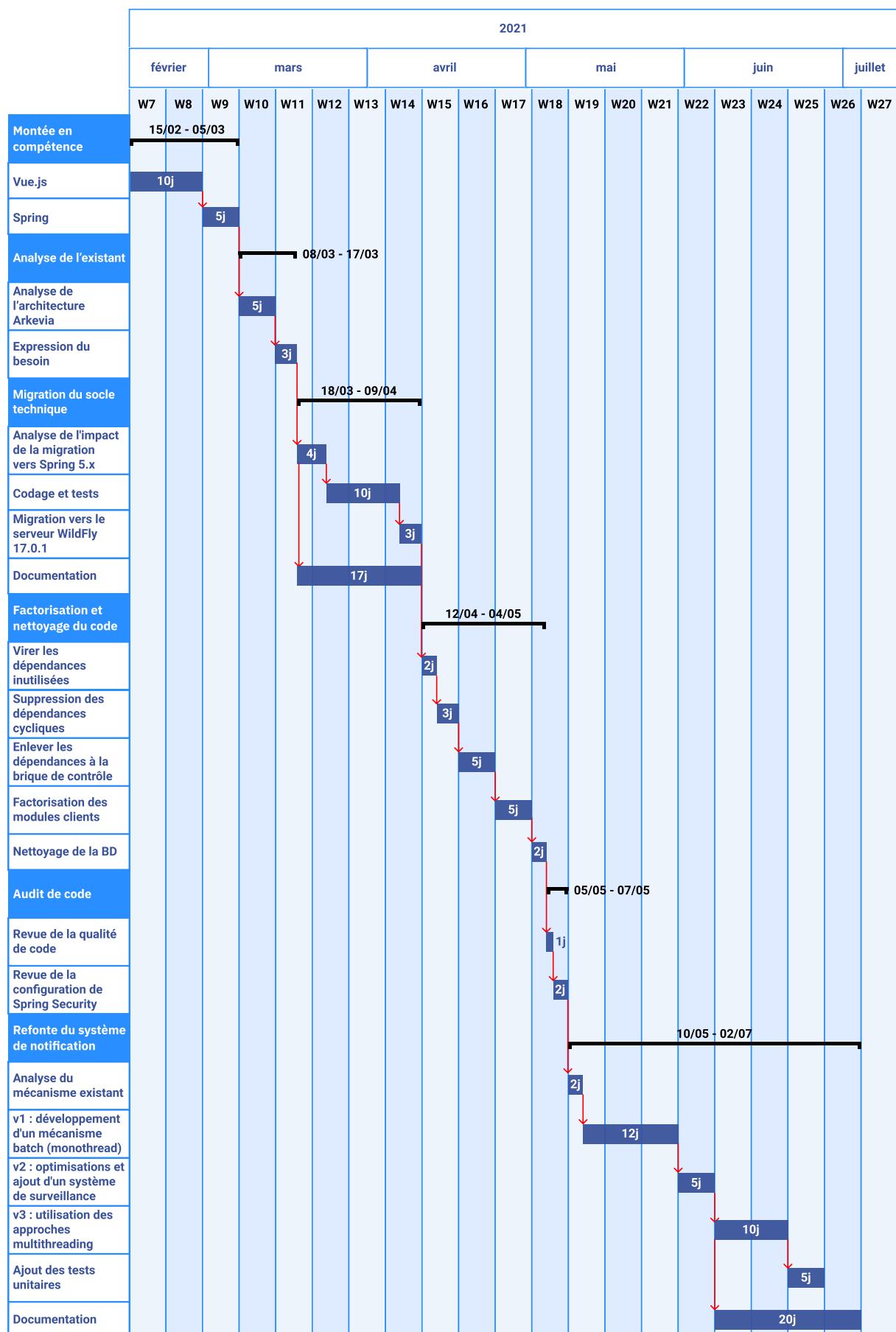


Figure 3.4. Diagramme de Gantt

Conclusion

La gestion de projets comme l'on peut le constater s'avère être d'une grande nécessité. Il est nécessaire pour mener à bien un projet de prendre en compte plusieurs dimensions dont toutes ne sont pas forcément quantifiables.

Cegedim SRH adopte désormais la méthode Scrum, l'une des méthodes Agile les plus répandues, pour le développement de ses propres solutions. L'Agilité, résolument basée sur l'efficacité et la création de valeur ajoutée, place véritablement clients et éditeurs dans un mode collaboratif avec comme vision un objectif commun.

En conséquence, grâce à son expertise avérée de cette méthodologie. Cegedim SRH a pu réduire les délais de conception et de production sans pour autant compromettre la qualité du produit final.

Chapitre 4

Conceptions et spécification des besoins

Introduction

Ce chapitre est voué à fournir une conception détaillée du portail Arkevia (le portail existant), afin d'appréhender son mode de fonctionnement et de bien cerner les besoins et exigences liés au sujet. Nous poursuivons cette conception par l'élaboration des besoins, des contraintes, et des méthodologies et solutions préconisées.

4.1 Étude conceptuelle du système existant

L'objectif de cette première section est de mener une étude conceptuelle approfondie du fonctionnement du système.

4.1.1 Méthodologie de conception

L'utilisation de la modélisation conceptuelle dans le développement des systèmes d'information permet de prendre en compte les besoins des applications d'une façon plus adéquate et de présenter d'une manière abstraite certains aspects des systèmes physiques et humains.

Afin de modéliser et décrire les différents aspects de notre système, nous utiliserons le langage graphique de modélisation unifié UML 2.0.

UML est un langage formel et normalisé en termes de modélisation objet. Son indépendance par rapport aux langages de programmation, aux domaines d'application et son caractère polyvalent ont fait de lui un langage universel. Il fournit un moyen astucieux permettant de représenter diverses projections, grâce aux diagrammes.

4.1.2 Identification des acteurs

Un acteur représente un rôle, c'est-à-dire une personne, un matériel ou un logiciel qui interagit directement avec le système. Les acteurs pouvant interagir avec l'application sont :

- **Employeur** : Personne morale adressant aux utilisateurs des documents.
- **Titulaire** : Salarié, actuel ou passé, de l'Employeur ayant ouvert et utilisant le coffre-fort électronique.
- **Utilisateur** : Titulaire ou, le cas échéant, les personnes physiques spécifiquement habilitées par le Titulaire.

- **Opérateur** : Personne morale qui met en œuvre un service de coffre-fort électronique et règle le fonctionnement opérationnel du système et des mesures de sécurité afférentes, en l'espèce CEGEDIM SRH.

4.1.3 Diagramme de cas d'utilisation général

Le diagramme de cas d'utilisation représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. C'est le premier diagramme du modèle UML, celui où s'assure la relation entre l'utilisateur et les objets que le système met en œuvre. Un cas d'utilisation est une description de l'application qui privilégie le point de vue de l'utilisateur. Il décrit de façon graphique (ou éventuellement textuelle) comment un acteur va utiliser l'application pour atteindre un objectif.

4 CONCEPTIONS ET SPÉCIFICATION DES BESOINS

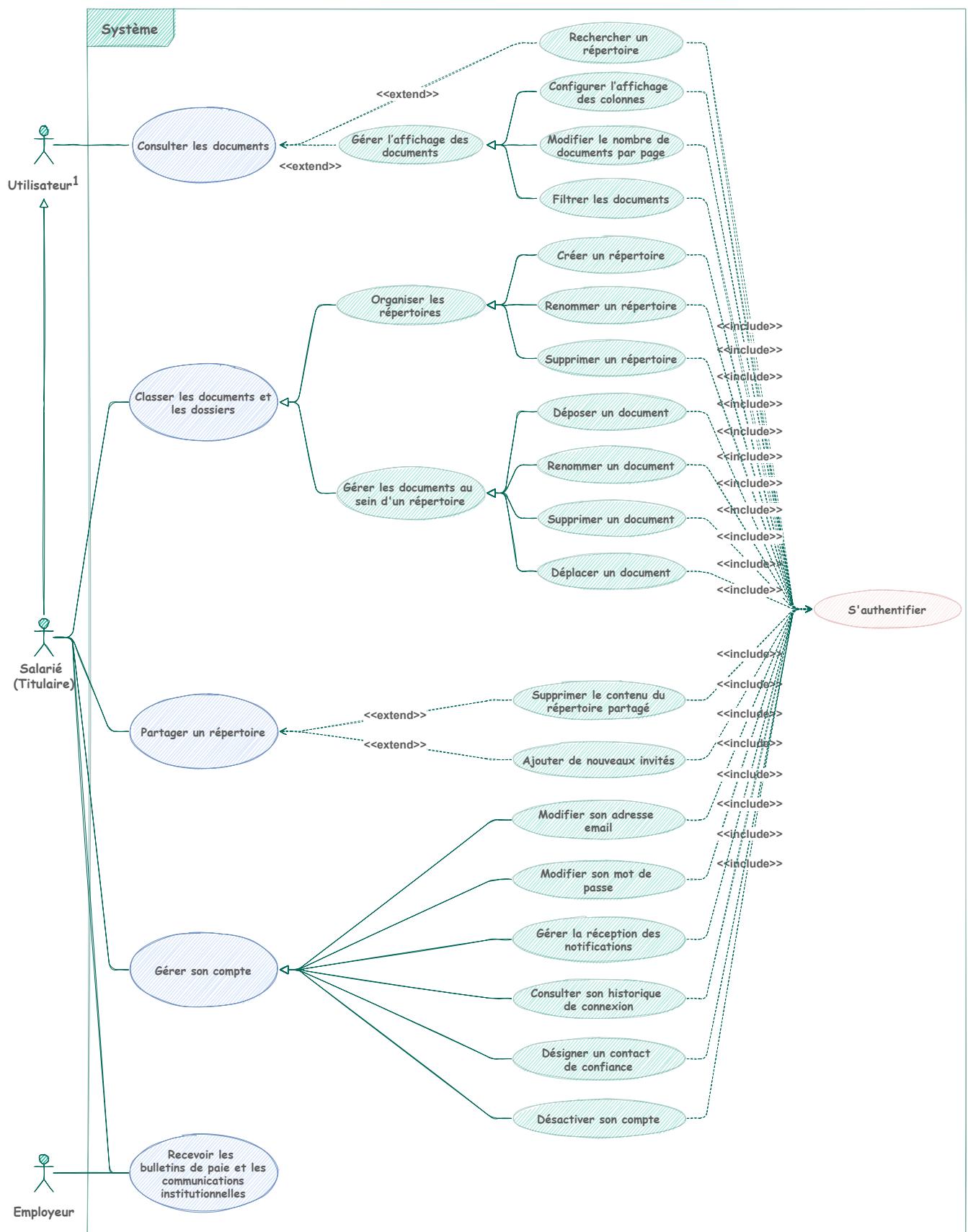


Figure 4.1. Diagramme de cas d'utilisation général

1. représente le cas d'un acteur (non propriétaire) autorisé à utiliser le compte.

4.1.4 Raffinement des cas d'utilisation prioritaires

Cas d'utilisation : « S'inscrire »

Table 4.1 : Description textuelle du CU « S'inscrire »

Acteur	Salarié (Titulaire)
Objectif	L'inscription permet aux salariés d'activer la mise à disposition de leurs bulletins de paie électroniques dans leurs coffres-forts.
Pré-condition	Disposer d'une connexion Internet et d'un navigateur.
Scénario	<ol style="list-style-type: none"> 1. Le salarié se rend sur www.myarkevia.com à l'aide d'un navigateur. 2. Le salarié clique sur JE M'INSCRIS. 3. Le salarié renseigne le matricule et le code secret qui lui ont été communiqués par le service RH, soit par des courriers spécifiques, soit par son dernier bulletin de salaire papier. 4. Le salarié doit tenir compte de la convention de mise à disposition du bulletin de paie électronique par l'employeur et cocher la case <input checked="" type="checkbox"/> J'ai lu et j'accepte les conditions de la convention pour pouvoir passer à l'étape suivante. 5. Le salarié renseigne les champs obligatoires concernant son profil. 6. Le salarié doit tenir compte des Conditions Générales d'Utilisation et cocher la case <input checked="" type="checkbox"/> J'accepte les conditions générales d'utilisation d'ARKEVIA pour pouvoir passer à l'étape suivante. 7. Le salarié saisit son mot de passe conformément aux règles de sécurité¹, puis le confirme. 8. Enfin, le salarié clique sur VALIDER MON INSCRIPTION pour confirmer son inscription.
Exception	<ul style="list-style-type: none"> ● Ouvert aux seuls salariés des entreprises ayant conclues un contrat de services avec CEGEDIM SRH (« Service ARKEVIA ») ● Le salarié saisit un matricule ou un code secret invalide. ● Le salarié saisit un mot de passe non conforme aux règles de sécurité définies par le système.
Post-condition	Le système redirige automatiquement le salarié vers la page de connexion.

¹ Le mot de passe doit comporter un minimum de 8 caractères et un maximum de 20 caractères, au moins une lettre, au moins un chiffre, au moins un caractère spécial parmi @#\$%&+=?_!;,.: et ne doit pas contenir d'espaces.

4 CONCEPTIONS ET SPÉCIFICATION DES BESOINS

Cas d'utilisation : « Se connecter »

Table 4.2 : Description textuelle du CU « Se connecter »

Acteur	Salarié, actuel ou passé, ayant ouvert et utilisant le coffre-fort électronique, et le cas échéant, les personnes physiques spécifiquement habilitées par le titulaire.
Pré-condition	<ul style="list-style-type: none"> • Avoir préalablement suivi la procédure d'inscription. • Disposer d'une connexion Internet et d'un navigateur.
Scénario	<ol style="list-style-type: none"> 1. L'utilisateur se rend sur www.myarkevia.com à l'aide d'un navigateur. 2. L'utilisateur saisit son adresse e-mail et son mot de passe. <ul style="list-style-type: none"> • Si nécessaire, il peut cliquer sur l'oeil  dans le champ Mot de passe pour voir son mot de passe en toutes lettres et ainsi éviter des erreurs de saisie. 3. Le salarié clique sur JE ME CONNECTE pour confirmer son inscription.
Exception	<ul style="list-style-type: none"> • Le salarié entre un mot de passe ou un e-mail incorrect. • Au-delà de trois tentatives erronées, le système suspecte un abus ou une utilisation illégale de la part de l'utilisateur et bloque donc l'accès au compte pour une période de 15 minutes.
Post-condition	Le système redirige l'utilisateur vers la page d'accueil de son compte.

Cas d'utilisation : « Créer un nouveau répertoire »

Table 4.3 : Description textuelle du CU « Créer un nouveau répertoire »

Acteur	Salarié (Titulaire)
Objectif	Permettre au salarié de créer ses propres répertoires afin de stocker et classer ses documents (bulletins de paie ou documents personnels qu'il a déposés dans son coffre-fort).
Pré-condition	S'authentifier avec un identifiant correct.

Suite à la page suivante

4 CONCEPTIONS ET SPÉCIFICATION DES BESOINS

Table 4.3 : Description textuelle du CU « Créer un nouveau répertoire » (Suite)

Scénario	<ol style="list-style-type: none"> Dans la section répertoire de l'écran, le salarié sélectionne le dossier sous lequel il souhaite créer un répertoire. Le dossier sélectionné est alors marqué en vert. Le salarié clique sur l'icône Créer un nouveau répertoire. Le salarié renseigne le nom du nouveau répertoire. Le salarié clique sur Créer.
Exception	<ul style="list-style-type: none"> Le nom du répertoire est invalide². Création d'un répertoire avec un nom qui existe déjà.
Post-condition	Le système renvoie un message indiquant à l'utilisateur qu'il peut désormais déposer des fichiers dans son nouveau répertoire.

² Les noms de répertoire ne peuvent pas contenir d'espaces, ni de caractères non conformes aux règles de nommage des fichiers Unix.

Cas d'utilisation : « Déposer un document »

Table 4.4 : Description textuelle du CU « Déposer un document »

Acteur	Salarié (Titulaire)
Objectif	Permettre aux salariés d'importer leurs documents importants.
Pré-condition	S'authentifier avec un identifiant correct.
Scénario	<ol style="list-style-type: none"> Le salarié sélectionne un dossier sous lequel il souhaite importer ses documents. Le salarié clique sur Déposer un document. Dans la fenêtre Ajout d'un document, le salarié clique sur Choisir un fichier. Le salarié parcourt ses répertoires et sélectionne un fichier. Il peut éventuellement sélectionner plusieurs fichiers en appuyant sur la touche Ctrl et en cliquant simultanément avec la souris sur les fichiers souhaités. Le salarié clique sur Ajouter.
Exception	Le fichier dépasse la taille limite autorisée de 50 Mo.

Suite à la page suivante

4 CONCEPTIONS ET SPÉCIFICATION DES BESOINS

Table 4.4 : Description textuelle du CU « Déposer un document » (Suite)

Post-condition	Le système renvoie un message indiquant à l'utilisateur que les documents sont en cours d'importation, puis un autre message indiquant le statut de l'opération lorsqu'elle est terminée.
-----------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Cas d'utilisation : « Gérer un document »

Table 4.5 : Description textuelle du CU « Gérer un document »

Acteur	Salarié (Titulaire)
Objectif	Permettre aux salariés d'importer leurs documents importants.
Pré-condition	S'authentifier avec un identifiant correct.
Scénario	<ol style="list-style-type: none"> 1. Le salarié sélectionne un répertoire pour accéder à son contenu. 2. Dans la partie Détail du répertoire de l'écran, Il peut cliquer sur : <ul style="list-style-type: none"> • L'icône Consulter  pour consulter le document. • L'icône Renommer  pour renommer le document. • L'icône Supprimer  pour supprimer le document. • Le nom du document et, tout en maintenant le clic, il peut le glisser-déposer dans un répertoire de son choix.
Exception	<ul style="list-style-type: none"> • Les documents professionnels déposés par l'employeur ne peuvent pas être supprimés par le salarié. • Le nom du document ne peut pas être renommé avec un nom qui existe déjà dans le répertoire parent.
Post-condition	Pour chaque action effectuée, le système renvoie un message indiquant à l'utilisateur l'état de l'opération lorsqu'elle est terminée.//

Cas d'utilisation : « Gérer l'affichage des documents »

Table 4.6 : Description textuelle du CU « Gérer l'affichage des documents »

Acteur	Salarié (Titulaire)
Objectif	Permettre aux salariés de configurer l'affichage des colonnes du tableau Détail du répertoire , de filtrer les documents ainsi que d'ajuster le nombre de documents affichés par page.
Pré-condition	S'authentifier avec un identifiant correct.

Suite à la page suivante

4 CONCEPTIONS ET SPÉCIFICATION DES BESOINS

Table 4.6 : Description textuelle du CU « Gérer l'affichage des documents » (Suite)

	<ul style="list-style-type: none"> ● Pour configurer l'affichage des colonnes : <ol style="list-style-type: none"> 1. Le salarié sélectionne un répertoire pour en afficher le contenu dans le tableau Détail du répertoire. 2. Le salarié clique sur la liste déroulante Gérer les colonnes et coche les colonnes qu'il souhaite afficher et décoche celles qu'il souhaite masquer. ● Si le répertoire contient un grand nombre de documents, ils sont alors affichés sur plusieurs pages que le salarié peut parcourir à l'aide des boutons situés sous le tableau. Pour plus de commodité, le salarié a la possibilité d'afficher plus de lignes dans le tableau et ainsi de réduire le nombre de pages. Pour ce faire : <ol style="list-style-type: none"> 1. Le salarié clique sur la liste déroulante [X] lignes. 2. Le salarié sélectionne 5, 10 ou 20 lignes. ● Le salarié a également la possibilité de filtrer les documents pour n'afficher que ceux correspondant aux critères de sélection qu'il a choisis. Pour ce faire : <ol style="list-style-type: none"> 1. Le salarié clique sur le bouton Filtrer. 2. Parmi les 5 filtres proposés, le salarié clique sur les filtres souhaités. 3. Dans le champ qui s'affiche, le salarié renseigne les critères de filtrage. 4. Le salarié clique sur Appliquer les filtres³. 5. Le salarié clique sur la croix X pour fermer le volet des filtres. ● Pour désactiver le filtrage : <ol style="list-style-type: none"> 1. Le salarié clique sur le bouton Filtrer pour afficher le volet des filtres. 2. Pour désactiver tous les filtres, le salarié clique sur le bouton Annuler. 3. Le salarié peut également désactiver certains filtres en cliquant sur le filtre à désactiver puis sur la croix X pour supprimer les critères saisis. A la fin, l'employé clique sur Appliquer les filtres pour rafraîchir la liste des documents. 4. Le salarié clique sur la croix X pour fermer le volet des filtres.
Exception	Aucune
Post-condition	Pour chaque action effectuée, l'affichage est instantanément modifié.

³ Les filtres actifs s'affichent au-dessus de la section **Détail du répertoire**.

Cas d'utilisation : « Gérer le partage des documents »

Table 4.7 : Description textuelle du CU « Partager un répertoire »

Acteur	Salarié (Titulaire)
Objectif	Donnez aux salariés la possibilité de créer des répertoires de partage afin qu'ils puissent partager des documents avec des tiers en dehors de leur entreprise.
Pré-condition	S'authentifier avec un identifiant correct.
Scénario	<p>1. Le salarié doit d'abord créer un répertoire partagé, cela peut être fait à travers le scénario suivant :</p> <ul style="list-style-type: none"> (a) Dans la liste des répertoires, le salarié clique sur le dossier Partage. (b) Le salarié clique sur l'icône Ajouter un répertoire partagé . (c) Dans la fenêtre Nouveau partage, le salarié renseigne les champs suivants : <ul style="list-style-type: none"> ● Nom du répertoire. ● Liste des invités : il saisit les adresses e-mail des personnes avec lesquelles il souhaite partager ses documents, séparées par une virgule. ● Commentaire. ● Date d'expiration du partage : il saisit la date⁴ jusqu'à laquelle le tiers pourra accéder au contenu de son répertoire de partage. (d) Le salarié clique sur . <p>2. Pour ajouter des documents dans le répertoire partagé, le salarié sélectionne et glisse les documents vers le répertoire partagé précédemment créé⁵.</p> <p>3. Le salarié clique sur l'icône Invitation partage .</p> <p>4. Éventuellement, le salarié peut :</p> <ul style="list-style-type: none"> ● Ajouter de nouveaux e-mails d'invités dans le champ Liste d'invités. ● Modifiez la date d'expiration du partage. <p>5. Le salarié coche la case Envoyer une notification aux invités⁶.</p> <p>6. Le salarié clique sur .</p>

Suite à la page suivante

4 CONCEPTIONS ET SPÉCIFICATION DES BESOINS

Table 4.7 : Description textuelle du CU « Partager un répertoire » (Suite)

Exception	L'utilisateur inscrit un e-mail dont le format est considéré comme invalide.
Post-condition	Les invités reçoivent un mail contenant un lien de téléchargement des documents partagés qui sera actif jusqu'à la date d'expiration du partage. Une fois la date d'expiration atteinte, le répertoire partagé est supprimé du dossier Partage .

⁴ La date saisie doit être postérieure à la date du jour.

⁵ En ajoutant des documents au dossier partagé, l'utilisateur crée simplement une copie des documents d'origine. Ils ne sont en aucun cas supprimés du répertoire d'origine.

⁶ Il est impératif de cocher la case Envoyer une notification aux invités, sinon ils ne recevront pas de notification pour télécharger les documents qui leur sont partagés.

Table 4.8 : Description textuelle du CU « Supprimer un partage »

Acteur	Salarié (Titulaire)
Objectif	Supprimer tout ou partie des documents partagés avec un tiers avant la date d'expiration du partage.
Pré-condition	S'authentifier avec un identifiant correct.
Scénario	<ol style="list-style-type: none"> 1. Le salarié sélectionne le dossier partagé dans le dossier Partage. 2. Dans la section Détail du répertoire, le salarié clique sur l'icône Supprimer  pour supprimer le document du répertoire⁷.
Exception	Aucune.
Post-condition	<p>Le répertoire partagé restera visible dans la liste des répertoires partagés jusqu'à ce que la date d'expiration du partage soit atteinte.</p> <p>Si les invités cliquent sur le lien de téléchargement, le répertoire téléchargé ne contiendra pas les documents supprimés ou sera vide si le salarié a choisi de supprimer tous les documents.</p>

⁷ Pour supprimer tous les documents du dossier partagé, le salarié doit supprimer individuellement chaque document contenu au sein du répertoire.

Gestion de compte

Pour accéder à la gestion de son compte, le salarié clique sur l'icône **Mon compte**. Plusieurs fonctionnalités sont proposées :

4 CONCEPTIONS ET SPÉCIFICATION DES BESOINS

Table 4.9 : Description textuelle du CU « Modifier son adresse email »

Acteur	Salarié (Titulaire)
Objectif	L'adresse e-mail est utilisée comme login pour le coffre-fort d'Arkevia. C'est également l'adresse où l'acteur reçoit les notifications lorsqu'un document est déposé dans son coffre-fort.
Pré-condition	S'authentifier avec un identifiant correct.
Scénario	<ol style="list-style-type: none"> 1. Le salarié clique sur l'onglet Mes informations personnelles. 2. Ensuite, il remplit la nouvelle adresse dans le champ E-mail, puis la confirme dans le champ Confirmer E-mail*. 3. Enfin, le salarié clique sur MODIFIER.
Exception	<ul style="list-style-type: none"> • L'employé introduit une adresse électronique qui ne suit pas les règles du format email standard. • Les deux champs E-mail et Confirmer E-mail* ne sont pas identiques.
Post-condition	La nouvelle adresse électronique devient alors le nouveau identifiant du salarié.

Table 4.10 : Description textuelle du CU « Modifier son mot de passe »

Acteur	Salarié (Titulaire)
Objectif	La politique de sécurité impose un changement de mot de passe tous les deux mois. Dès que le délai est expiré, il sera nécessaire de le changer lors de la connexion au coffre-fort. Cependant, il est possible de le changer librement à tout moment.
Pré-condition	S'authentifier avec un identifiant correct.
Scénario	<ol style="list-style-type: none"> 1. Le salarié clique sur l'onglet Mon mot de passe. 2. Le salarié saisit son mot de passe actuel. 3. Ensuite, il saisit son nouveau mot de passe⁸, puis le confirme. 4. Enfin, le salarié clique sur MODIFIER.
Exception	Le salarié saisit un mot de passe non conforme aux règles de sécurité définies par le système.
Post-condition	L'ancien mot de passe est changé.

⁸ Le nouveau mot de passe doit respecter les règles de sécurité listées dans la partie droite de l'écran

4 CONCEPTIONS ET SPÉCIFICATION DES BESOINS

Table 4.11 : Description textuelle du CU « Consulter son historique de connexion »

Acteur	Salarié (Titulaire)
Objectif	Chaque connexion au coffre-fort est historisée. Le salarié peut recevoir une notification par email à chaque connexion et ainsi être averti de tout accès involontaire à son coffre-fort.
Pré-condition	S'authentifier avec un identifiant correct.
Scénario	<ol style="list-style-type: none"> 1. Le salarié clique sur l'onglet Historique de connexion. 2. Le salarié peut cliquer sur Exporter en Excel ou Exporter en PDF pour récupérer la liste complète des connexions à son coffre-fort. 3. Le salarié peut également cocher/décocher la case <input checked="" type="checkbox"/> Être averti par e-mail des connexions sur mon compte pour recevoir ou non une notification à chaque fois que son compte est ouvert, puis cliquer sur MISE À JOUR pour enregistrer la modification.
Exception	Aucune.
Post-condition	Un tableau affiche les six dernières connexions (date et heure de connexion ainsi que l'adresse IP à partir de laquelle a eu lieu la connexion).

Table 4.12 : Description textuelle du CU « Désigner un contact de confiance »

Acteur	Salarié (Titulaire)
Objectif	Pouvoir désigner une personne de confiance qui pourra récupérer le contenu du coffre-fort du salarié si ce dernier n'est plus en mesure de l'utiliser.
Pré-condition	S'authentifier avec un identifiant correct.
Scénario	<ol style="list-style-type: none"> 1. Le salarié clique sur l'onglet Contact de confiance. 2. Le salarié remplit le formulaire avec l'identité de la personne de confiance. 3. Il confirme son choix en cochant la case autorisant le contact à accéder au contenu de son coffre-fort. 4. Enfin, il clique sur VALIDER pour enregistrer cette modification.
Exception	Aucune.

Suite à la page suivante

4 CONCEPTIONS ET SPÉCIFICATION DES BESOINS

Table 4.12 : Description textuelle du CU « Désigner un contact de confiance » (Suite)

Post-condition	Le contact de confiance est désigné.
-----------------------	--------------------------------------

Table 4.13 : Description textuelle du CU « Désactiver son compte »

Acteur	Salarié (Titulaire)
Objectif	Demander la désactivation du compte ARKEVIA.
Pré-condition	S'authentifier avec un identifiant correct.
Scénario	<ol style="list-style-type: none">1. Le salarié clique sur l'onglet Désactivation du compte.2. Le salarié tenir compte et lire attentivement les informations relatives à la désactivation de son compte.3. Le salarié coche la case <input checked="" type="checkbox"/> Désactivation du compte ARKEVIA4. Enfin, il clique sur ACCEPTER .
Exception	Si le salarié a une inscription activée aux bulletins de paie électroniques auprès de son employeur, il doit au préalable se désinscrire auprès de son service RH. Sans cette désinscription avant désactivation du compte, l'employeur continuera d'y déposer ses bulletins de paie électroniques sans que lui puisse y accéder.
Post-condition	Le salarié recevra un lien par email, actif pendant une heure, pour confirmer la désactivation de son compte.

4.2 Analyse des besoins

4.2.1 Migration du socle technique

La migration du socle technique consiste à faire évoluer le stack applicatif d'un projet, pour évoluer généralement sur une technologie plus récente. Une telle migration de projet applicatif englobe un certain nombre de variables qu'il convient maîtriser (migration de données, de technologies, de fichiers...), ce dernier peut vite devenir complexe.

Il y a deux principaux cas qui conduisent à entamer un projet de migration :

- **La migration de projet contrainte par les éditeurs et l'évolution des briques applicatives (bases de données, frameworks, technologies...).** Un projet applicatif est composé d'un ensemble de briques applicatives qui suivent chacune leur propre cycle, qui est souvent très rapide! En effet, les éditeurs mettent régulièrement à jour leurs solutions, et très vite, les anciennes versions ne sont plus

supportées (ce qui pose des problèmes en termes de sécurité, de maintenabilité et de compatibilité). Il est donc impératif de suivre de très près l'évolution des briques applicatives.

- **La migration volontaire de projet (plus rare)** : permet d'anticiper l'évolution du projet et/ou du SI, et de mieux préparer l'avenir sur des bases solides et récentes.

Dans ce contexte, j'ai dû entreprendre le sujet de la migration afin d'évoluer vers une solution plus optimale tout en préservant la cohérence globale du système en question. Ces migrations représentent un double défi : d'une part rattraper le retard technologique des anciennes briques du système et d'autre part s'assurer que l'application fonctionnera bien, et qu'elle ne sera pas confrontée à des limites de développement de nouvelles fonctionnalités.

Nul doute que tous ces défis s'accompagnent de réflexions globales sur le métier, les infrastructures, les applications et la culture pour assurer la cohérence totale des futurs processus créés. Dans cet opus, nous allons nous intéresser à une partie de ces chantiers qui est la migration du framework de base Spring.

Avant d'entamer un chantier de transformation d'une telle ampleur, la compréhension du besoin client est fondamentale pour une migration d'un bloc applicatif Legacy vers une nouvelle base. Pour clôturer cette section, nous allons identifier les principales étapes qui serviront à la réalisation de la migration au cours de ses différentes phases que nous allons détailler ci-dessous :

1. La première phase est celle de l'audit et de la récolte d'information. Le but est de se familiariser avec le contexte global de la migration :
 - (a) Audit global du périmètre SI à migrer;
 - (b) Étude des documents d'architecture techniques et fonctionnels;
 - (c) Déterminer les scores techniques des différentes briques applicatives : chaque brique applicative sera évaluée selon plusieurs critères. Parmi ces critères évalués, nous pouvons citer les suivants :
 - La criticité de l'application;
 - Les technologies et Frameworks utilisés;
 - Les besoins métiers futurs (les évolutions des processus);
 - Les prérequis de l'infrastructure cible;
 - Les transformations globales en cours de réalisation (Agilité, conteneurisation ou autre évolution technique, changement d'infrastructure en cours, etc.);
 - L'estimation de l'effort de migration;

- La conformité avec les standards de développement.
- (d) Revoir les contraintes techniques et opérationnelles;
- (e) Analyser les risques d'une manière plus détaillée pour chaque périmètre applicatif;
- (f) Évaluer les points de blocage.
2. Au terme de la première phase, nous disposerons d'une vision détaillée du SI qui décrira précisément, via des spécifications techniques, les actions qui seront menées, les modalités de la stratégie de migration, et la trajectoire que l'on voudra lui donner à court, à moyen, ou à long terme, ce qui nous permettra au final de bien procéder aux travaux de développement de la migration.

4.2.2 Refactorisation et nettoyage du code

4.2.2.1 Refactorisation des modules applicatifs

Arkevia est composée de huit modules permettant de gérer les différents aspects de l'application :

- **HS-SY-Portal** : Ce module contient tous les éléments qui composent le corps du front-end Arkevia, y compris les composants Vue, les scripts, les images et les feuilles de style.
- **HS-SY-Client-Arkevia** : Ce module regroupe tous les services propres au client Arkevia, notamment l'inscription, la désactivation du compte, l'exportation de documents, etc.
- **HS-SY-Client-Standard** : Ce module comprend tous les services standards et les classes d'écran utilisés dans l'application.
- **HS-SY-Ged-Master** : Ce module comprend les services de gestion des répertoires et des documents, y compris la consommation de l'API SignArchive.
- **HS-SY-Master** : Ce module inclut les services de configuration, d'authentification, de gestion des comptes, etc.
- **HS-SY-Master-Domain** : Ce module permet de gérer les interactions avec la base de données, il rassemble toutes les entités qui sont utilisées pour faire le mapping avec les tables. On y trouve aussi tous les objets d'accès aux données.
- **HS-SY-Ged-Domain** : Ce module regroupe toutes les entités spécifiques à la gestion des documents, qui serviront à faire le mapping avec les tables.
- **HS-SY-Referentiel** : Ce module expose les services et les méthodes nécessaires pour établir la connexion avec l'API de SignArchive par le biais de requêtes Solr.

L'objectif est de réorganiser, restructurer et clarifier le code existant. Le besoin est de factoriser les modules clients pour rendre l'application plus générique. Pour ce faire, nous devons d'abord étudier la possibilité de virer le module **HS-SY-Client-Standard**.

Si cela est faisable, nous serons amenés à fédérer toute la partie gestion des écrans clients du module `HS-SY-Client-Standard` au module `HS-Sy-Client-Arkevia`, puis à retirer entièrement le module `HS-SY-Client-Standard` du projet.

D'autre part, pour garder plus de clarté et d'organisation dans les fichiers de POM, nous pourrons également détacher les liens de dépendance directe entre les modules pour ne garder au final qu'un seul lien de dépendance transitif entre ces modules (voir figure 4.2).

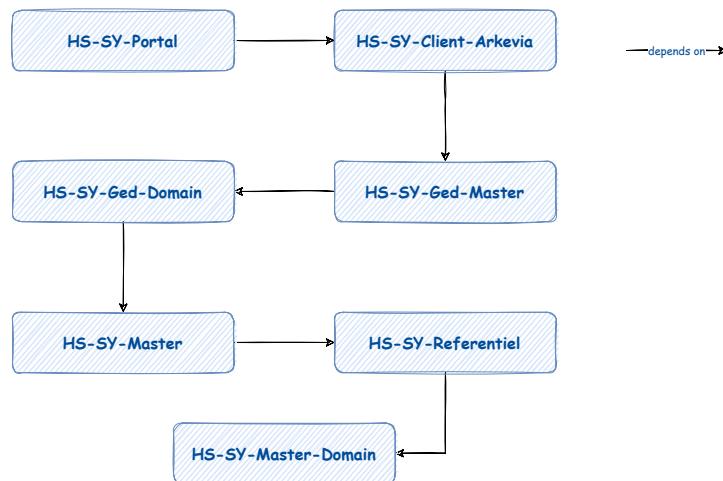


Figure 4.2. Architecture modulaire d'Arkevia

4.2.2.2 Renoncer à recourir à la brique de contrôle

La brique de contrôle (Semantic Model And Business Control) est un service web tiers utilisé pour charger des métadonnées qui servent principalement d'en-tête pour le tableau de description des documents.

Suite à un audit mené sur l'application Arkevia afin d'identifier et de déboguer les réponses des appels provenant de cette brique, un total de douze appels a été relevé dans le code de l'application. En effet, cette brique de contrôle soulève des problèmes de performance, notamment un problème de temps d'arrêt très élevé. Afin d'éviter ce goulot d'étranglement, il a été convenu d'éliminer cette dépendance et de la remplacer par un simple appel à la base de données compte tenu que les réponses provenant de cette brique ne sont rien d'autre que des données et des configurations statiques.

4.2.2.3 Retrait des dépendances inutilisables

C'est un scénario typique, lorsque l'application est en cours de développement, et que plusieurs membres de l'équipe travaillent dessus, on risque de se trouver face à une série de dépendances inutiles dans les fichiers pom.xml du projet.

Il est alors ardu de déterminer manuellement quels sont les jars que l’application ne requiert pas. Pour résoudre ce problème, nous pouvons nous servir du plugin maven-dependency, en particulier l’objectif `dependency:analyze`, qui permet d’analyser les dépendances du projet et de déterminer celles qui sont : utilisées et déclarées ; utilisées et non déclarées ; utilisées et déclarées.

4.2.3 Optimisation du mécanisme d’envoi des notifications

Le mécanisme de notification est responsable de prévenir un utilisateur par email lorsqu’il y a un nouveau dépôt de document (par exemple un bulletin de salaire) sur son coffre-fort par l’entreprise à laquelle il est abonné.

Prérequis nécessaire à la compréhension de sujet :

- **Document** : bulletins de paie, ou tout autre document RH au format dématérialisé.
- **Lot de documents** : un nombre limité (batch) de documents (ajustables).
- **Notification** : Un mail générique transmis à un utilisateur pour le notifier du dépôt d’un document.
- **Scan** : une requête lancée à l’API SignArchive afin de vérifier s’il y a des dépôts de documents.
- **Cron** : une tâche planifiée qui déclenche l’exécution du Scan toutes les deux heures (configurable).

Récupération de données

Le processus de notification est déclenché toutes les deux heures grâce à l’annotation `@Scheduled` fournie par Spring.

Tout d’abord, nous récupérons la date du dernier scan effectué, et nous enregistrons le nouveau scan avec la date du jour dans la table `GED_BATCH_SCAN_UPLOAD_DOC`. La date du dernier scan est utilisé par la suite, pour récupérer tous les documents déposés à partir de cette date jusqu’à le moment de lancement du nouveau scan.

4 CONCEPTIONS ET SPÉCIFICATION DES BESOINS

Table 4.14 : Exemple de déroulement du Cron

Scan	Date	Documents à traiter
1	03/02/2021 10:00:00	Tout dépôt entre 03/02/2021 08:00:00 et 03/02/2021 10:00:00
2	03/02/2021 12:00:00	Tout dépôt entre 03/02/2021 10:00:00 et 03/02/2021 12:00:00
3	03/02/2021 14:00:00	Tout dépôt entre 03/02/2021 12:00:00 et 03/02/2021 14:00:00
...

La récupération des données est faite à l'aide de la méthode `getLastPushedFiles(Date lastDateScan, int indexStart, int count, Client client)`. Cette méthode permet de récupérer un nombre défini de dépôts émis dans un intervalle de temps donné.

Table 4.15 : Description des paramètres de la fonction `getLastPushedFiles`

Paramètre	Description
lastDateScan	c'est la date du dernier scan enregistré dans la table <code>GED_BATCH_SCAN_UPLOAD_DOC</code>
indexStart	permet de spécifier l'indice du premier élément du lot
count	permet de spécifier la taille du lot
client	le nom du client (toujours "arkevia")

Le fonctionnement dernière consiste à :

- Construire les critères de recherche.
- Appeler la méthode `findDocumentByCriteria` du service `GedService`.
- Interroger l'API SignArchive au moyen d'une requête Solr.

Traitement de données

Une boucle explore chaque document de la façon suivante :

```
do {Traitement}  
while (totalPageNumber > pageIndex)  
// totalPageNumber : nombre total des documents divisé par 20  
// pageIndex : nombre des lots traités
```

Traitement :

- Vérifier la validité du matricule et de son propriétaire.
- Rechercher l'utilisateur lié à ce matricule et récupérer son adresse mail.
- Vérifier s'il y a déjà une notification envoyé vers cet utilisateur pour ce document - les notifications sont persistés dans la table `DocumentUploadNotification`.
- Préparer un nouveau objet de notification avec les informations (date, id du document, id de l'utilisateur, etc.).
- Persister cet objet dans la base de données et l'ajouter dans une file d'attente (sans envoyer la notification immédiatement).

Le schéma suivant (figure 4.3) illustre l'approche employée pour traiter ces lots de données :

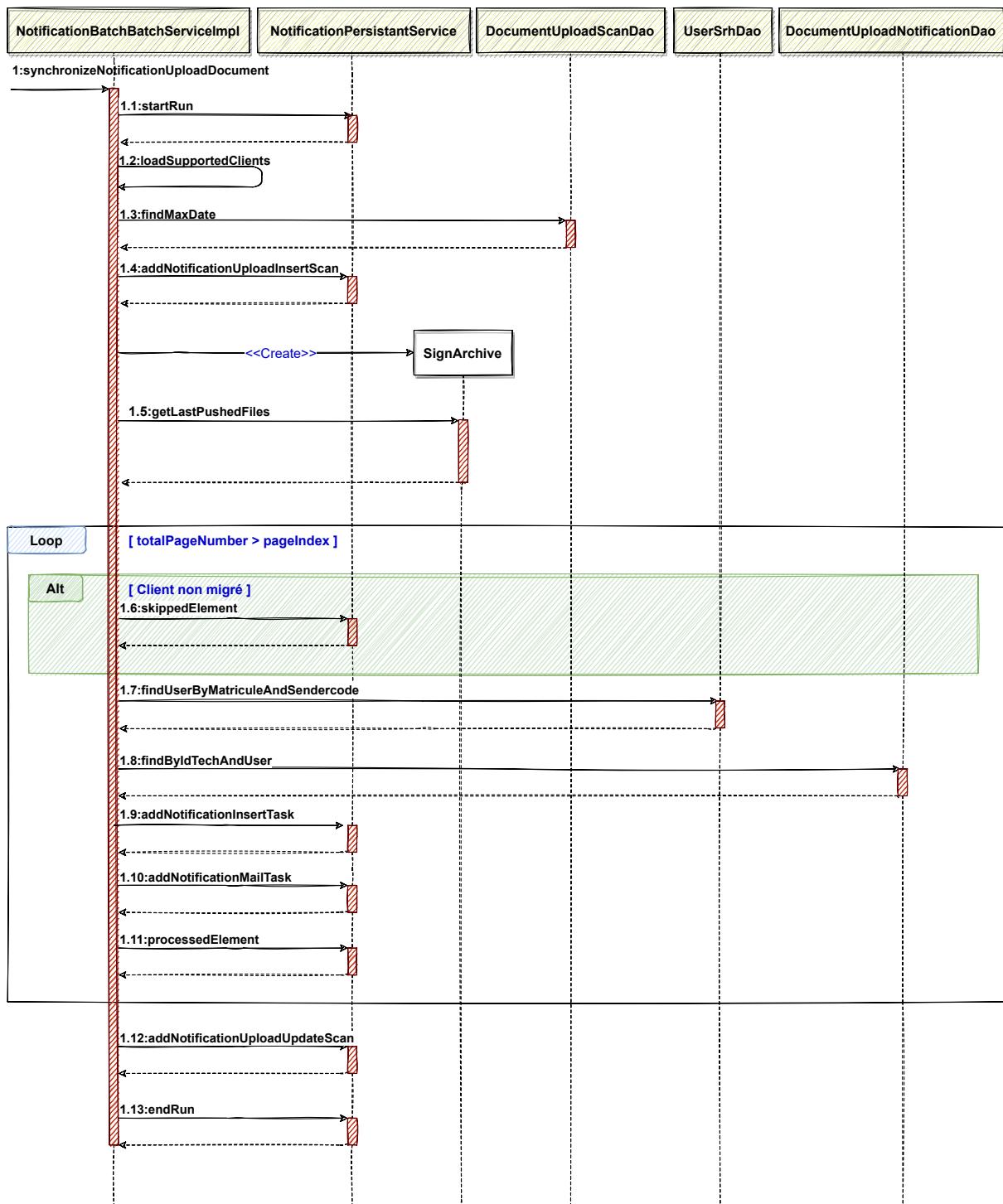


Figure 4.3. Diagramme de séquence modélisant le processus d'envoi de notifications

Problématique et limites

- Mauvaise gestion des indices de recherche pour la méthode getLastPushedFiles, ce qui conduit à boucler sur les documents déjà traités, et par conséquent à dépenser plus de ressources et de temps.

- L'appel Solr est couteuse en terme de temps, l'ancien mécanisme sollicite Solr pour chaque 20 documents (c'est-à-dire, si on a 1000 documents \Rightarrow on doit le solliciter 50 fois).
- La réponse de Solr ne parvient pas à distinguer les clients migrés des clients non migrés, et l'ancien mécanisme prend en compte tous les documents, y compris ceux des clients qui n'ont pas encore migré. Le traitement de documents non pertinents exige un temps considérable.

La nouvelle approche

Le nouveau système prévu devrait reposer sur des paradigmes de multithreading permettant le traitement parallèle de plusieurs lots de données. Ce mécanisme devrait aussi engendrer des changements majeurs dans le processus d'extraction et d'exploitation des données, notamment en ignorant les documents associés aux clients non migrés et en intégrant un service de monitoring qui permettra d'envoyer régulièrement aux superviseurs des courriels comprenant des statistiques sur chaque fichier traité.

Conclusion

En clair, comme nous venons de le voir, ce chapitre a été largement consacré à une conception détaillée de l'application à travers les diagrammes UML et les descriptions textuelles des cas d'utilisation qui nous ont permis notamment de bien appréhender le fonctionnement de l'application, mais surtout de faciliter le processus d'identification des exigences liées au sujet lesquelles ont fait l'objet de la deuxième partie du chapitre. À présent, nous sommes en mesure d'entamer la partie réalisation.

Chapitre 5

Réalisation

Introduction

Après avoir présenté les différentes étapes de conception et la méthodologie suivie, nous présenterons dans ce chapitre les outils de développement et les différents composants logiciels et matériels, pour lesquels constituent la mise en œuvre effective des différentes tâches du sujet.

Le nouveau système d'envoi des notifications fera l'objet d'une démonstration dans la deuxième partie de ce chapitre.

5.1 Environnements et outils

L'objectif de cette section est de révéler les principales technologies présentes dans Arkevia, ainsi que celles appliquées lors du développement des différentes solutions.

5.1.1 Composants logiciels

5.1.1.1 Front-end

Table 5.1 : Technologies utilisées au niveau du front-end

B Bootstrap	Bootstrap est un framework front-end extrêmement robuste permettant de développer des applications et des sites Web de manière plus rapide et plus conviviale. Il comprend des modèles de conception basés sur HTML et CSS pour les composants d'interface utilisateur courants tels que les formulaires, les boutons, les tableaux, les navigations, les alertes, les onglets et bien d'autres, ainsi que les extensions JavaScript optionnelles. Bootstrap offre également la possibilité de créer des mises en page réactives (responsive design) avec un effort minimal.
------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Suite à la page suivante

5 RÉALISATION

Table 5.1 : Technologies utilisées au niveau du front-end (Suite)

 Element UI	<p>Element est une bibliothèque de composants d'interface utilisateur basée sur Vue 2.0 et qui jouit du soutien d'une large communauté. Elle n'est pas seulement destinée aux développeurs front-end, mais fournit également un kit d'interface utilisateur complet avec lequel les concepteurs et les chefs de produit peuvent travailler. Elle est spécifiquement conçue pour la création d'interfaces utilisateur de bureau, mais prend en charge certaines fonctionnalités réactives telles que le masquage des éléments en fonction de la taille de la fenêtre et la création de grilles.</p>
 jQuery	<p>jQuery est une bibliothèque JavaScript libre créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web. Elle propose comme principales fonctionnalités :</p> <ul style="list-style-type: none"> ● La manipulation du Document Object Model (DOM). ● La gestion des événements (mouvements de souris, clics, etc.) et de l'AJAX. ● La création d'effets d'animation. ● La manipulation des feuilles de style en cascade.
 jQuery UI	<p>jQuery UI est une bibliothèque JavaScript basée sur jQuery, fournissant une collection d'éléments utiles au développement d'interfaces utilisateur. Ces éléments comprennent :</p> <ul style="list-style-type: none"> ● Des interactions comme le "drag & drop" (glisser-déposer) ● Des "widgets" (composants d'interface graphique) telles que les barres de progression, les infobulles, etc. ● Des effets pour modifier dynamiquement l'apparence des éléments de l'interface (par exemple, changer la couleur, faire apparaître/disparaître un élément, etc.) ● Des thèmes avec des propriétés CSS pour la mise en page des éléments interactifs.
 Sass	<p>Sass (Syntactically Awesome Style Sheets) est une extension de CSS intégrant des fonctionnalités telles que les règles imbriquées, les variables, les mixins et les extensions de classe. Cela permet aux développeurs d'écrire des CSS structurés, lisibles et réutilisables. Sass est compilé en CSS standard. Il s'agit principalement d'un langage de préprocesseur CSS qui accepte à la fois le CSS et sa syntaxe personnalisée d'écriture de codes de conception visuelle.</p>

Suite à la page suivante

Table 5.1 : Technologies utilisées au niveau du front-end (Suite)

 VueJs	<p>Vue (prononcé /vju:/, comme le terme anglais view) est un framework évolutif pour construire des interfaces utilisateur. À la différence des autres frameworks monolithiques, Vue a été conçu et pensé pour pouvoir être adopté de manière incrémentale. Le cœur de la bibliothèque se concentre uniquement sur la partie front-end. D'un autre côté, Vue est tout à fait capable de faire tourner des applications web mono-pages quand il est couplé avec des outils modernes et des bibliothèques complémentaires.</p>
---------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

5.1.1.2 Back-end

Table 5.2 : Technologies utilisées pour les solutions back-end

 HIBERNATE Hibernate	<p>Hibernate est une bibliothèque de mappage objet-relationnel (ORM) pour le langage Java permettant aux développeurs d'utiliser des modèles de domaine de style POJO dans leurs applications d'une manière qui va bien au-delà du mappage objet/relationnel.</p>
 Java	<p>JAVA est un langage de programmation de haut niveau, orienté objet, fonctionnel, indépendant de la plate-forme et un environnement d'exécution. Le langage Java tire une grande partie de sa syntaxe du C et du C++, mais son modèle objet est plus simple que celui de ce dernier et il a moins de facilités de bas niveau. Les applications Java sont généralement compilées en bytecode (appelés fichiers de classe) qui peuvent être exécutés par une JVM (Java Virtual Machine), indépendamment de l'architecture informatique. La JVM compile souvent le code en code machine natif pour optimiser les performances.</p>
 Spring	<p>Spring est un framework open source fournissant une boîte à outils très riche permettant de structurer, d'améliorer et de simplifier l'écriture d'application Java. Spring est également livré avec une variété de modules dédiés à l'exécution de différentes tâches. Certains d'entre eux sont Spring Test, Spring Security, Spring Web, Spring JDBC, Spring AOP, Spring MVC et Spring ORM.</p>

Suite à la page suivante

Table 5.2 : Technologies utilisées pour les solutions back-end (Suite)

 Spring Batch	<p>Spring Batch est un framework open source basé sur Spring pour permettre le développement d'applications batch qui sont essentielles au fonctionnement quotidien des systèmes d'entreprise. En général, les applications batch font référence à des systèmes automatisés conçus pour traiter des données de masse. Spring Batch automatise cette itération de base des lots, en offrant la possibilité de traiter des transactions similaires comme un ensemble, souvent dans un environnement isolé, sans aucune interaction avec l'utilisateur.</p>
 Spring Boot	<p>Spring Boot permet de créer facilement une application alimentée par Spring avec un minimum d'effort. Une application créée avec Spring Boot peut être :</p> <ul style="list-style-type: none"> ● Créée sans requérir aucune configuration xml. ● Crée sans aucune exigence de serveur d'application puisque Spring Boot fournit un serveur d'application (Tomcat intégré, Jetty ou Undertow). ● Largement configuré avec quelques valeurs par défaut et des POM de démarrage pour simplifier la configuration Maven du projet. ● Fournit des solutions prêtes pour la production telles que les métriques, l'intégrité de performance et la configuration externalisée.

5.1.1.3 Système de gestion de base de données

Arkevia est conçu pour fonctionner sur une instance Oracle 10g (ou plus). Le SGBDR **Oracle** est utilisé par toutes les applications de la solution ARKEVIA. Il dispose aujourd'hui de l'une des meilleures prestations en termes de performance, de scalabilité et d'administration.

5.1.1.4 Stockage et sécurité des données

Le coffre-fort Arkevia est un espace de stockage sécurisé qui repose sur la technologie **Morpho Secure Storage** de la société **MORPHO** qui assure :

- L'intégrité des documents, au moyen d'une fonction de signature électronique.
- La confidentialité des documents, au moyen d'une fonction de chiffrement de données.
- La traçabilité des actions effectuées (dépôts, restitutions, demandes de copies, etc.).
- Les documents ont ainsi une valeur probante (juridiquement opposable).

- ✓ Le dépôt ou l'extraction de fichiers ne peut se faire qu'à partir de l'application ARKEVIA, via les Web Services avec authentification SSL Client/Serveur.
- ✓ Les fichiers sont horodatés, signés, chiffrés et stockés dans un espace sécurisé du Datacenter de Cegedim.
- ✓ Le contenu est chiffré avec l'algorithme AES 128 GCM, la clé appartient à Cegedim. Le chiffrement des flux est sécurisé en HTTPS / TLS 1.2 (AES 256) avec un certificat SHA-256 appartenant à Cegedim.
- ✓ Les mots de passe des utilisateurs sont protégés par « hashage » via l'algorithme SHA-256.

5.1.2 Environnement de développement

5.1.2.1 Environnement matériel

Les tâches assignées ont été élaborées sur un ordinateur de bureau conçu pour réaliser les différentes activités liées aux thèmes du stage, soit directement, soit par le biais du protocole RDP. L'ordinateur fourni a les spécifications matérielles et logicielles suivantes :

- **Fabricant** : Dell Inc.
- **Modèle du système** : OptiPlex 7040
- **Processeur** : [01] : Intel64 Family 6 Model 94 Stepping 3 GenuineIntel 3312 MHz
- **Mémoire physique totale** : 16 309 Mo
- **Système d'exploitation** : Microsoft Windows 10 Professionnel pour les Stations de travail.

5.1.2.2 Environnement logiciel et outils

Dans cette partie, nous dévoilerons les différents environnements et outils adoptés pour la collaboration, l'échange, la gestion et le suivi des activités, ainsi que pour le développement applicatif des solutions retenues tout au long du processus de développement logiciel. Le tableau suivant (voir tableau 5.3) détaille ces environnements et outils par ordre alphabétique :

5 RÉALISATION

Table 5.3 : Environnements et outils de développement et de collaboration

 Confluence	<p>Atlassian Confluence est un système de collaboration et de wiki pour les entreprises. Atlassian Confluence est utilisé pour la collaboration, la gestion de la base de connaissances, la rédaction technique et en tant qu'intranet social ou gestionnaire de documents.</p>
 GitLab	<p>GitLab est une plateforme DevOps complète proposée sous la forme d'une application unique. Elle révolutionne le développement, la sécurité, l'exploitation et la collaboration entre les équipes.</p>
 IntelliJ IDEA (Ultimate Edition)	<p>IntelliJ IDEA est un IDE complet développé par JetBrains (anciennement « IntelliJ ») axé sur la productivité avec des systèmes d'autocomplétion intelligente, d'analyse de code en temps réel, de refactoring avancé; l'intégration d'outils de tests et de debugging; et une pléthore de raccourcis clavier permettant de réaliser rapidement presque toutes les tâches.</p>
 Jira	<p>Jira est une plateforme multifonctionnelle qui vise à faciliter la gestion de projet en aidant à suivre les tâches, à identifier les points de blocage et à diffuser l'information entre les différentes parties prenantes. En pratique, les cas d'utilisation les plus courants de JIRA sont les suivants :</p> <ul style="list-style-type: none"> ● La gestion du support et des activités de développement logiciel. ● Le suivi des anomalies. ● Le suivi d'activité. ● La gestion des centres de services.
 Microsoft Excel	<p>Microsoft Excel est un logiciel tableur de la suite bureautique Microsoft Office développé et distribué par l'éditeur Microsoft. Il comprend des outils de calcul, de création de graphiques, de tri et de filtrage des données, de création de tableaux croisés dynamiques et un langage de programmation de macros appelé "Visual Basic for Applications".</p>

Suite à la page suivante

5 RÉALISATION

Table 5.3 : Environnements et outils de développement et de collaboration (Suite)

 Microsoft Outlook	<p>Microsoft Outlook est un gestionnaire d'informations personnelles de Microsoft (utilisé principalement pour gérer le courrier électronique), disponible à la fois en tant qu'application distincte et en tant que partie de la suite Microsoft Office. Il intègre un client de messagerie, un calendrier (gestionnaire de rendez-vous) et d'autres outils d'organisation des informations personnelles.</p>
 Microsoft PowerPoint	<p>PowerPoint est un logiciel de présentation édité par Microsoft. Il est principalement utilisé pour créer des présentations destinées à être projetées. Toutefois, en raison de ses larges aptitudes, il est également utilisé pour l'animation, l'apprentissage en ligne, la diffusion sur le Web, les rapports commerciaux, etc.</p>
 Oracle SQL Developer	<p>Oracle SQL Developer est un outil gratuit conçu pour améliorer la productivité et simplifier les tâches de développement des bases de données Oracle. Il s'agit d'un outil graphique entièrement pris en charge pour le développement des bases de données Oracle, y compris le parcours des objets de la base de données, l'exécution des instructions/scripts SQL, la modification et le débogage des instructions PL/SQL. En outre, il est possible d'exécuter un nombre quelconque de rapports fournis, ainsi que de créer et d'enregistrer des rapports personnalisés.</p>
 Postman	<p>Postman est un environnement de développement d'API complet permettant de concevoir, de mocker, de tester, de surveiller et de publier des API à partir de l'interface utilisateur Postman.</p>
 SonarQube	<p>SonarQube est une plateforme open source dédiée à l'analyse et à la mesure en continu de la qualité du code source. Elle permet aux développeurs de détecter les bugs et les vulnérabilités ainsi que de réduire les mauvaises pratiques de code, et de ce fait d'améliorer systématiquement la qualité du code.</p>
 Sourcetree	<p>Développé par Atlassian, SourceTree est un client Git qui intègre très efficacement le workflow Gitflow avec une interface graphique pour gérer presque tout sans avoir à passer par le terminal.</p>

Suite à la page suivante

5 RÉALISATION

Table 5.3 : Environnements et outils de développement et de collaboration (Suite)

 WildFly	WildFly, autrefois connu sous le nom de JBoss Application Server, est un serveur d'applications open source (LGPL), développé par Redhat, pouvant être utilisé sur tout système d'exploitation fournissant une machine virtuelle Java (JVM).
 Zoom	La crise sanitaire et la généralisation du télétravail ont largement encouragé l'adoption des outils de visio-conférence. L'un de ces outils est Zoom, qui est essentiellement un outil permettant aux professionnels de mener facilement des visio-conférences et des réunions en ligne, ainsi que d'échanger des messages, des documents, des images, des vidéos et d'autres types de données avec les participants d'un groupe de discussion.

5.2 Démonstration de la fiabilité du nouveau système d'envoi de notifications

Le traitement par lot, communément appelé Batch dans le jargon informatique, est une problématique très répandue et quasiment incontournable au sein des entreprises et industries qui manipulent d'énormes masses de données. Dans cette section, nous allons vous présenter à travers un exemple d'application, la technologie Spring Batch/Spring Boot permettant de répondre à ce type de besoin.

Le système nouvellement conçu est une application autonome indépendante de l'application mère Arkevia, créée avec Spring Boot, une technologie qui facilite le développement d'applications basées sur Spring, et le framework Spring Batch pour orchestrer le traitement des notifications. Cela n'aurait pas été concevable sans recourir à des paradigmes de programmation avancés tels que le multithreading, qui permet la gestion et l'exécution d'un grand nombre d'instances de manière concurrente, ainsi que les concepts d'inversion de contrôle et d'injection de dépendances, qui facilitent la coordination et le contrôle de l'activité de l'application. Enfin, ce mécanisme a engendré des changements majeurs dans le processus d'extraction et l'exploitation des données, que nous allons expliquer dans la suite de cette section.

Le graphique suivant (voir figure A.3) illustre de manière concrète la relation existant entre les différents composants impliqués dans le batcher lors de son exécution.

5 RÉALISATION

Note : À noter que dans ce framework, lorsque l'on parle de batch on parle plus précisément de Job.

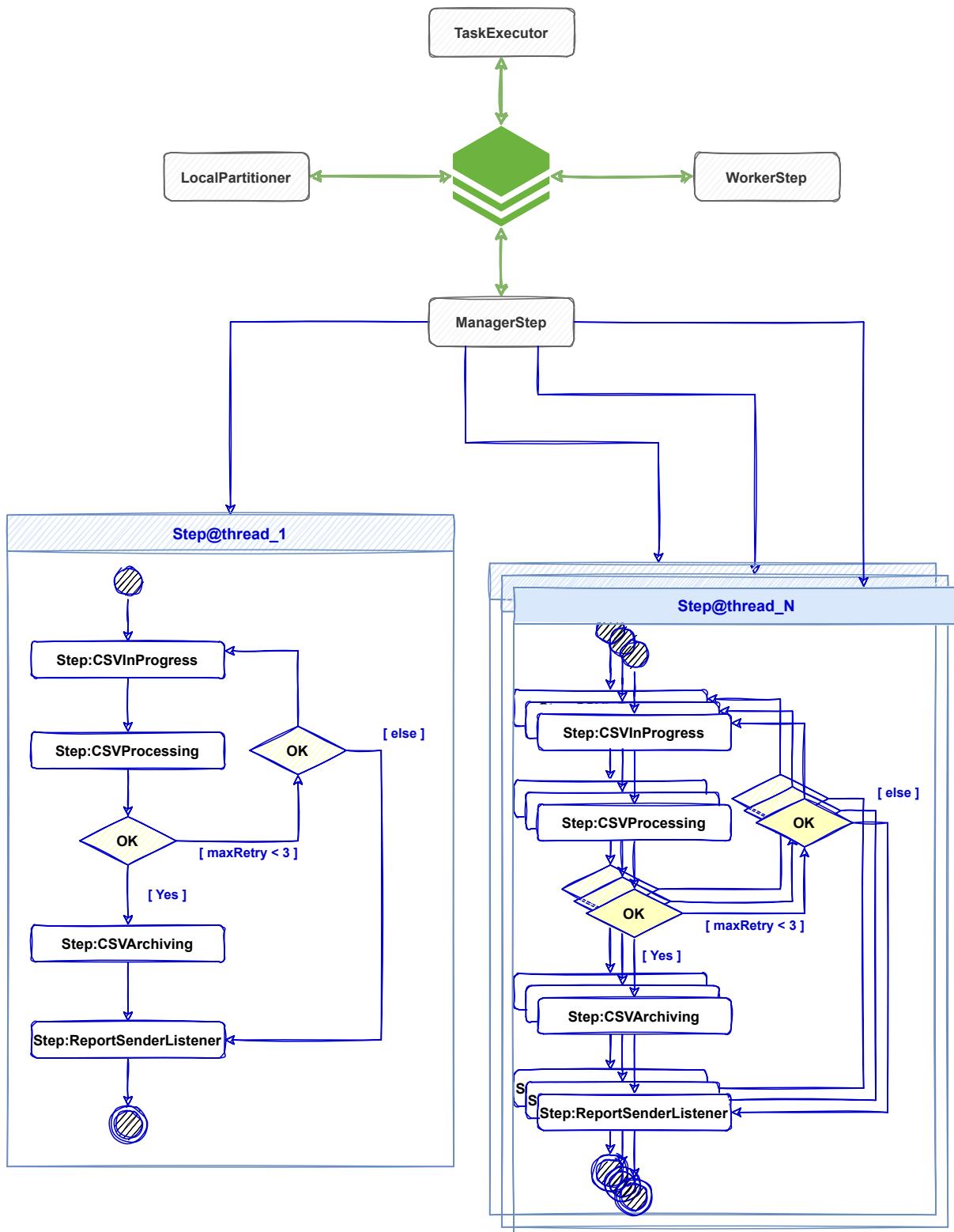


Figure 5.1. Liens entre les divers composants impliqués dans le traitement par lots

- **ManagerStep** : Ce composant permet de créer des copies des instances de step de flux basées sur une clé pour distribuer les données des fichiers provenant du Partitioner, également il se base sur le composant **taskExecutor** et principalement sur la valeur de `maxPoolSize` pour définir le nombre maximum de threads à exécuter.
- **Partitioner** : est un composant qui permet de définir un ensemble de valeurs d'entrée sur une collection `Map` pour chacun des fichiers existants. En d'autres termes, la logique pour diviser les tâches en threads respectifs va ici.
- **TaskExecutor** : est un JavaBean qui fournit une abstraction autour de l'instance `java.util.concurrent.ThreadPoolExecutor` et l'expose en tant que Spring Bean. Grâce à ce composant, vous pouvez configurer le nombre de threads créés, le nombre de threads à conserver dans la file d'attente, le préfixe du thread, etc.
- **WorkerStep** : Représente le flux d'étapes qui seront appliquées et exécutées parallèlement pour accomplir le traitement d'un fichier donné.

Analyse du flux de données et traitement par lots

L'application batch est planifiée par défaut pour s'exécuter automatiquement toutes les deux heures. Une fois le Job est lancé, le composant **Partitionneur** scanne le répertoire d'entrée afin de trouver des éventuels fichiers csv à traiter ;
À ce stade, il y a deux cas :

1. Si le répertoire d'entrée **input** est vide, le programme scanne le répertoire **inprogress** pour vérifier s'il y a des fichiers qui ne sont pas traités correctement ou s'ils ont été interceptés lors d'une exécution antérieure ou encore bloqués pour un autre motif, dans ce contexte il y a deux hypothèses :
 - (a) Le répertoire **inprogress** est vide, dans ce cas le job se termine avec le statut **COMPLETED**.
 - (b) Le répertoire **inprogress** n'est pas vide, dans ce cas le programme reprendra les fichiers CSV trouvés et les traitera une autre fois dans le cadre d'une politique de relance établie (fixée à 3 essais maximum).
2. Si le répertoire d'entrée contient des fichiers, le programme créera des instances du Job (threads) pour traiter chaque fichier indépendamment ; le nombre de threads créés dépend de la valeur donnée au programme et du nombre de fichiers trouvés (pour altérer cette valeur, veuillez se référer à la propriété `maxPoolSize` dans le fichier `application.properties`) ; si le nombre de fichiers dépasse le nombre maximum de threads autorisés, les fichiers restants seront ajoutés à la file d'attente (dès qu'un thread sera libre, il prendra en charge un fichier dans la file d'attente).

5 RÉALISATION

Pour chaque thread lancé, le programme entreprend à différentes phases d'appliquer une série de validations aux fichiers associés afin de gérer efficacement le traitement et les cas de récupération en cas d'échec :

1. La première phase est appelée **step:CSVInProgress**, elle permet de déplacer le fichier courant vers le répertoire **inprogress**, pour commencer son traitement.
2. La phase successive est celle du traitement des fichiers CSV, elle constitue le cœur du processus batch et c'est dans cette phase que s'effectuent la lecture, la validation et la transformation des données, et finalement l'envoi des emails pour les données approuvées.
3. Au terme de la deuxième phase, si le traitement du fichier s'est déroulé avec succès, le fichier sera acheminé à la dernière phase qui est celle de l'archivage du fichier. Dans le cas contraire, le fichier sera ignoré jusqu'à ce que le batch finisse le traitement des fichiers existants dans le répertoire d'entrée **input**, puis il le reprend ensuite pour un second traitement (fixée à trois tentatives maximum).
4. Tout fichier ayant atteint ce stade d'archivage signifie que le fichier a été traité avec succès et il sera simplement déplacé vers le répertoire **done** et renommé selon le format suivant `file_name@date_processing.csv`.
5. À la fin du Job, un email de contrôle incluant le statut de la dernière exécution, le nombre d'emails envoyés par client, ainsi qu'un fichier Excel contenant la liste des utilisateurs auxquels l'email de notification n'a pas été expédié, sera envoyé au responsable.

5.2.1 Préparation de l'environnement et paramétrage du système

Afin de préparer la démonstration, nous commencerons par la mise en place de l'environnement d'exécution :

1. Établir les données d'entrée : À ce stade, et pour des fins de test, nous allons générer systématiquement quatre fichiers CSV de taille aléatoire. Ces fichiers sont composés de 3 champs :
 - **La matricule** : valeur numérique servant comme identifiant du salarié.
 - **Le code client** : valeur alphanumérique servant comme identifiant du client (société).
 - **Le IdTech** : valeur alphanumérique servant pour identifier le document déposé.

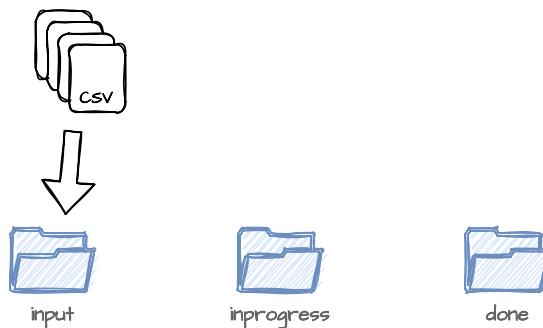
5 RÉALISATION

Exemple :

Table 5.4 : Exemple de fichiers CSV générés pour le test

Matricule	CodeClient	IdTech
1	CLIENT1	id_tech_0jeG8Y
2	CLIENT1	id_tech_J99qYD
3	CLIENT1	id_tech_80HcGQ
...
100	CLIENT1	id_tech_wG1X7C

2. Créez trois répertoires portant comme noms "input", "inprogress" et "done".
3. Déplacez les fichiers CSV préalablement créés dans le répertoire  input afin de les traiter avec le batcher.



4. Assurez-vous que les propriétés `folder.input`, `folder.inprogress` et `folder.done` pointent vers les chemins corrects pour les répertoires  input,  inprogress et  done, respectivement. Pour ce faire, allez consulter le fichier  src/main/resources/application.properties et vérifiez les paramètres suivants :

```
folder.input = C:/arkeviavars/input/  
folder.inprogress = C:/arkeviavars/inprogress/  
folder.done = C:/arkeviavars/done/
```

5 RÉALISATION

5. Vérifiez également d'autres configurations telles que le nombre maximum de threads pouvant être lancés concurremment sur une même exécution ou encore la possibilité d'activer/désactiver le mode de supervision permettant d'envoyer au gestionnaire un rapport détaillé sur l'exécution de chaque fichier abordé. Pour cette manipulation, nous allons activer le mode de supervision, ainsi que spécifier une valeur maximale pour les threads concurrents.

```
arkevia.monitoring.email = hamza.talaghzi@cegedim-srh.com  
mode.monitoring = true  
task_executor.core_pool_size = 10
```

5.2.2 Exécution, résultats et interprétations

- Ouvrez une invite de commande et rendez-vous à la racine du projet.
 - Afin de rendre la commande `mvn` opérationnelle, la variable d'environnement `JAVA_HOME` doit pointer vers le répertoire d'installation du JDK `jdk-11.0.X` :
Entrez `set JAVA_HOME` = suivi du chemin du répertoire d'installation JDK et appuyez sur .
 - Entrez `mvn spring-boot:run` et appuyez sur .
 - Lorsque vous exécuterez la commande, vous obtiendrez quelque chose qui ressemble à ceci :

5 RÉALISATION

```
26 [ notif@thread-2] c.c.s.n.b.BatchConfiguration : Reading file : C:/arkeviavars/inprogress/client4.csv
27 [ notif@thread-4] c.c.s.n.b.BatchConfiguration : Reading file : C:/arkeviavars/inprogress/client3.csv
28 [ notif@thread-3] c.c.s.n.b.BatchConfiguration : Reading file : C:/arkeviavars/inprogress/client1.csv
29 [ notif@thread-1] c.c.s.n.b.BatchConfiguration : Reading file : C:/arkeviavars/inprogress/client2.csv
30 [ notif@thread-1] o.s.b.c.s.AbstractStep : Step: [csv-processor] executed in 963ms
31 [ notif@thread-1] o.s.b.c.j.SimpleStepHandler : Executing step: [csv-archiver]
32 [ notif@thread-1] o.s.b.c.s.AbstractStep : Step: [csv-archiver] executed in 476ms
33 [ notif@thread-1] o.s.b.c.s.AbstractStep : Step: [slaveFlow:file 1] executed in 4s240ms
34 [ notif@thread-2] o.s.b.c.s.AbstractStep : Step: [csv-processor] executed in 6s45ms
35 [ notif@thread-2] o.s.b.c.j.SimpleStepHandler : Duplicate step [csv-archiver] detected in execution of
      job=[ notification -manager]. If either step fails , both will be executed again on restart.
36 [ notif@thread-2] o.s.b.c.j.SimpleStepHandler : Executing step: [csv-archiver]
37 [ notif@thread-2] o.s.b.c.s.AbstractStep : Step: [csv-archiver] executed in 474ms
38 [ notif@thread-2] o.s.b.c.s.AbstractStep : Step: [slaveFlow:file 3] executed in 9s382ms
39 [ notif@thread-4] o.s.b.c.s.AbstractStep : Step: [csv-processor] executed in 13s281ms
40 [ notif@thread-3] o.s.b.c.s.AbstractStep : Step: [csv-processor] executed in 13s278ms
41 [ notif@thread-4] o.s.b.c.j.SimpleStepHandler : Duplicate step [csv-archiver] detected in execution of
      job=[ notification -manager]. If either step fails , both will be executed again on restart.
42 [ notif@thread-4] o.s.b.c.j.SimpleStepHandler : Executing step: [csv-archiver]
43 [ notif@thread-3] o.s.b.c.j.SimpleStepHandler : Duplicate step [csv-archiver] detected in execution of
      job=[ notification -manager]. If either step fails , both will be executed again on restart.
44 [ notif@thread-3] o.s.b.c.j.SimpleStepHandler : Executing step: [csv-archiver]
45 [ notif@thread-4] o.s.b.c.s.AbstractStep : Step: [csv-archiver] executed in 482ms
46 [ notif@thread-3] o.s.b.c.s.AbstractStep : Step: [csv-archiver] executed in 476ms
47 [ notif@thread-4] o.s.b.c.s.AbstractStep : Step: [slaveFlow:file 2] executed in 16s647ms
48 [ notif@thread-3] o.s.b.c.s.AbstractStep : Step: [slaveFlow:file 0] executed in 16s933ms
49 [ scheduling-1] o.s.b.c.s.AbstractStep : Step: [partitionStep] executed in 18s94ms
50 [ scheduling-1] c.c.s.n.b.l.JobListener : Job Finished.
```

Interprétations :

- **Ligne 11** : Démarrage du Job, tout en chargeant certains paramètres qui seront utiles plus tard dans les étapes qui sont exécutées. Ces paramètres ne seront accessibles que dans le contexte du Job.
- **Ligne 13** : Exécution de l'étape de morcellement, cette étape examine le contenu des dossiers `input` et `inprogress` pour déterminer le nombre de fichiers à traiter afin de déduire le nombre de threads nécessaires au traitement de ces fichiers.
- **Lignes 14-17** : Comme nous avons déposé dans cette expérience quatre fichiers CSV dans le dossier `input`, quatre threads ont été créés en conséquence (Les threads sont nommés selon le pattern suivant `notif@thread-id`) et cela explique pourquoi nous observons dans les logs `Executing step : [csv-inprogress]` dupliquée 4 fois. Autrement dit, cela signifie que le batcher applique de manière simultanée la première phase du traitement des fichiers (`csv-inprogress`).
- **Lignes 18-21** : Ces logs indiquent que la première phase de traitement des fichiers est terminée.
- **Lignes 22-25** : Ces logs annoncent le début d'une nouvelle phase de traitement des fichiers CSV. Il s'agit de la phase de traitement CSV, qui concrétise le traitement

5 RÉALISATION

effectif du batch. C'est dans cette étape que s'effectuent la lecture, la validation et la transformation des données, et enfin l'envoi des emails pour les données approuvées.

- **Lignes 23-48 :** Ces logs exposent les différentes phases par lesquelles transite chaque fichier, à ce stade vous allez sûrement remarquer qu'il y a des threads qui ont pu passer plus rapidement les étapes et terminer tout le processus avant les autres, cela est dû au fait que chaque fichier a une taille différente et donc des délais d'exécution différents.
- **Ligne 49 :** Cette ligne révèle le temps d'exécution total écoulé lors du traitement des fichiers, qui est de 18s94ms.
- **Ligne 50 :** Fin du Job.

Au terme de cette exécution, le superviseur recevra des mails récapitulatifs sur le statut de traitement de chaque fichier. (voir figure 5.2).



Job execution report

jobId	20762					
fileName	client3.csv					
parentDir	input					
validLines	87					
skippedLines	nullFieldValue	unsupportedClientKeys	unfoundedUser	invalidEmail	alreadySent	total
	0	35	18	0	0	53

Figure 5.2. Exemple d'alerte envoyée au superviseur

Par ailleurs, les utilisateurs concernés par les notifications recevront un mail générique les informant du dépôt (voir figure 5.3).



Figure 5.3. Modèle générique de mail pour l'envoi de notifications

5.2.3 Tests unitaires

Les tests unitaires servent à tester une partie (une unité) d'un système afin de s'assurer qu'il fonctionne correctement (build the system right).

Comme Java est un langage orienté Objet, les tests JUnit sont regroupés dans des classes de test. Généralement, on groupe dans une classe les tests ayant la même classe comme point d'entrée et on nomme la classe de test à partir du nom de la classe testée préfixé ou suffixé par Test(s). Par exemple, pour tester la classe `FlatFileItemReader`, on créera une classe `TestFlatFileItemReader` OU `FlatFileItemReaderTests`.

Exécution des tests JUnit

L'exécution des tests unitaires se fait via Maven, et plus particulièrement avec le plugin **maven-surefire**. Une des méthodes pour lancer l'exécution des tests est d'exécuter la commande `mvn test` dans le terminal du répertoire parent du projet. Cela permet notamment d'analyser le projet afin de découvrir les classes de test pour ensuite les exécuter.

```
1 $ mvn test
2 [INFO] Scanning for projects ...
3 ---- certains logs sont omis par souci de concision ----
4 [INFO] -----
5 [INFO] T E S T S
6 [INFO] -----
7 [INFO] Running com.cegedim.srh.notification.batch.FlatFileItemReaderTests
8 ---- certains logs sont omis par souci de concision ----
9 [INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 10.926 s - in
   com.cegedim.srh.notification.batch.FlatFileItemReaderTests
10 [INFO] Running com.cegedim.srh.notification.batch.GedCSVAnalyticsProcessorTests
11 ---- certains logs sont omis par souci de concision ----
12 [INFO] Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.373 s - in
   com.cegedim.srh.notification.batch.GedCSVAnalyticsProcessorTests
13 [INFO] Running com.cegedim.srh.notification.batch.MainApplicationTests
14 ---- certains logs sont omis par souci de concision ----
15 [INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 2.287 s - in
   com.cegedim.srh.notification.batch.MainApplicationTests
16 [INFO]
17 [INFO] Results:
18 [INFO] -----
19 [INFO] Tests run: 7, Failures: 0, Errors: 0, Skipped: 0
20 [INFO] -----
21 [INFO] BUILD SUCCESS
22 [INFO] -----
23 [INFO] Total time: 22.359 s
24 [INFO] Finished at: 2021-11-07T10:17:09+01:00
25 [INFO] -----
26 [INFO]
```

Dans notre cas, les résultats obtenus révèlent que sur un total de 7 tests, il y a eu 7 tests réussis face à 0 test échoué et 0 test ignoré.

5.2.4 Audit de code

La vérification d'un ensemble de règles de programmation est généralement faite lors d'une revue de code ou bien d'un audit de code, manuellement ou bien automatiquement afin de trouver les violations de règles de programmation, dans le but de les corriger. Pour cela, nous utiliserons **SonarQube**, un outil d'analyse statique qui vise à mesurer la qualité du code d'un applicatif. Pour un projet donné, il fournit des métriques portant sur la qualité du code et permet d'identifier précisément les points à corriger (code mort, bugs potentiels, non-respect des standards, carence ou trop-plein de commentaires, etc.).

Remarque : L'analyse SonarQube ne remplace en aucun cas les tests unitaires, mais permet d'identifier rapidement certains défauts du code.

Prérequis

- La seule condition préalable à l'exécution de SonarQube est l'installation de Java (Oracle JRE 11 ou OpenJDK 11) sur votre machine.
- Installer SonarQube (pour plus de détails, voir [la documentation officielle](#)).

Installation d'une instance locale de SonarQube

Vous pouvez essayer SonarQube en utilisant une installation traditionnelle avec le fichier zip :

1. [Téléchargez](#) le fichier zip de SonarQube Community Edition.
2. Décompressez-le, dans un emplacement tel que C:/sonarqube ou /opt/sonarqube.
3. En tant qu'utilisateur non root, démarrez le serveur SonarQube :

```
# Sous Windows, exédez :  
C:\sonarqube\bin\windows-x86-64\StartSonar.bat
```

```
# Sur les autres SE, en tant qu'utilisateur non root, exédez :  
/opt/sonarqube/bin/[OS]/sonar.sh console
```

Mise en place d'une procédure d'analyse de projet

1. Une fois que votre instance est opérationnelle, connectez-vous à <http://localhost:9000> en utilisant les informations d'identification de l'administrateur système :
 - **login** : admin
 - **mot de passe** : admin

2. Maintenant que vous êtes connecté à votre instance locale de SonarQube, vous pouvez analyser un projet (Dans notre cas, nous allons poursuivre notre démonstration en appliquant l'analyse Sonar sur le nouveau système d'envoi de notification) :
 - (a) Créez un nouveau projet en choisissant le mode manuel.
 - (b) Donnez à votre projet une clé de projet et un nom d'affichage et cliquez sur le bouton **Set Up** (voir figure 5.4).

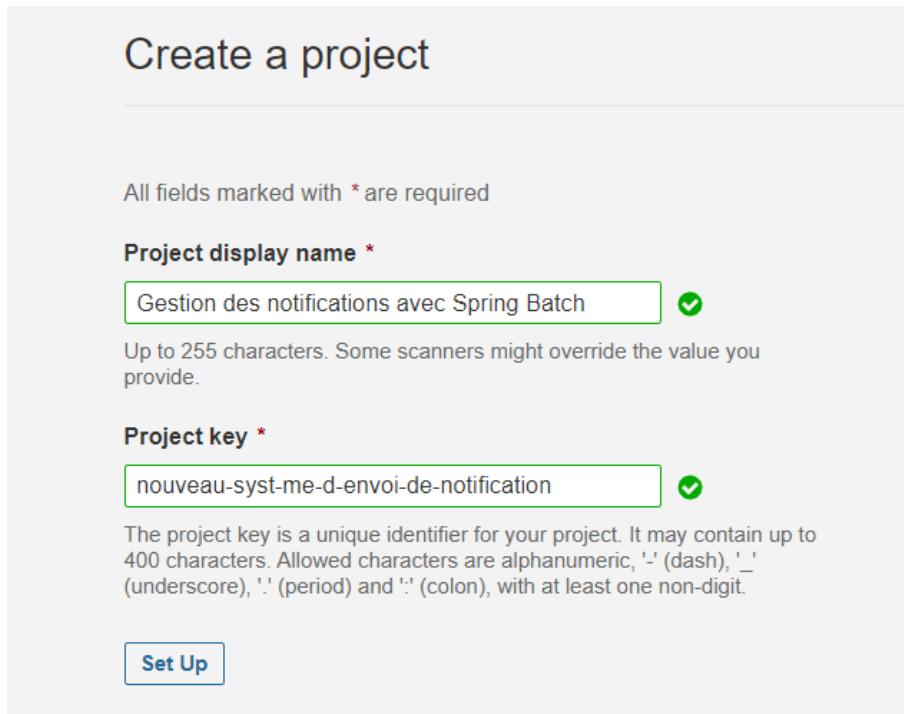


Figure 5.4. Mise en place d'un projet Sonar

- (c) Pour analyser votre projet localement, cliquez sur le bouton **Locally**.
- (d) Sous **Provide a token**, sélectionnez **Generate a token**. Donnez un nom à votre jeton, cliquez sur le bouton **Generate**, puis cliquez sur **Continue**.
- (e) Sélectionnez la langue principale de votre projet sous **Run analysis on your project**, et suivez les instructions pour analyser votre projet.
- (f) L'exécution d'une analyse SonarQube avec Maven est très simple. Dans notre cas, il suffit d'exécuter la commande suivante dans le dossier racine du projet :

```
mvn clean verify sonar:sonar \
-Dsonar.projectKey=nouveau-syst-me-d-envoi-de-notification \
-Dsonar.host.url=http://localhost:9000 \
-Dsonar.login=8c6f1a4ff703b9701af4606125ef587d836e5ef8
```

5 RÉALISATION

Analyse des résultats SonarQube

Une fois le projet analysé avec succès, nous pouvons consulter le résultat de cette première analyse sur SonarQube. Les résultats de l'analyse seront consultables sur <http://localhost:9000/>. Chaque projet dispose d'un espace d'analyse dédié :

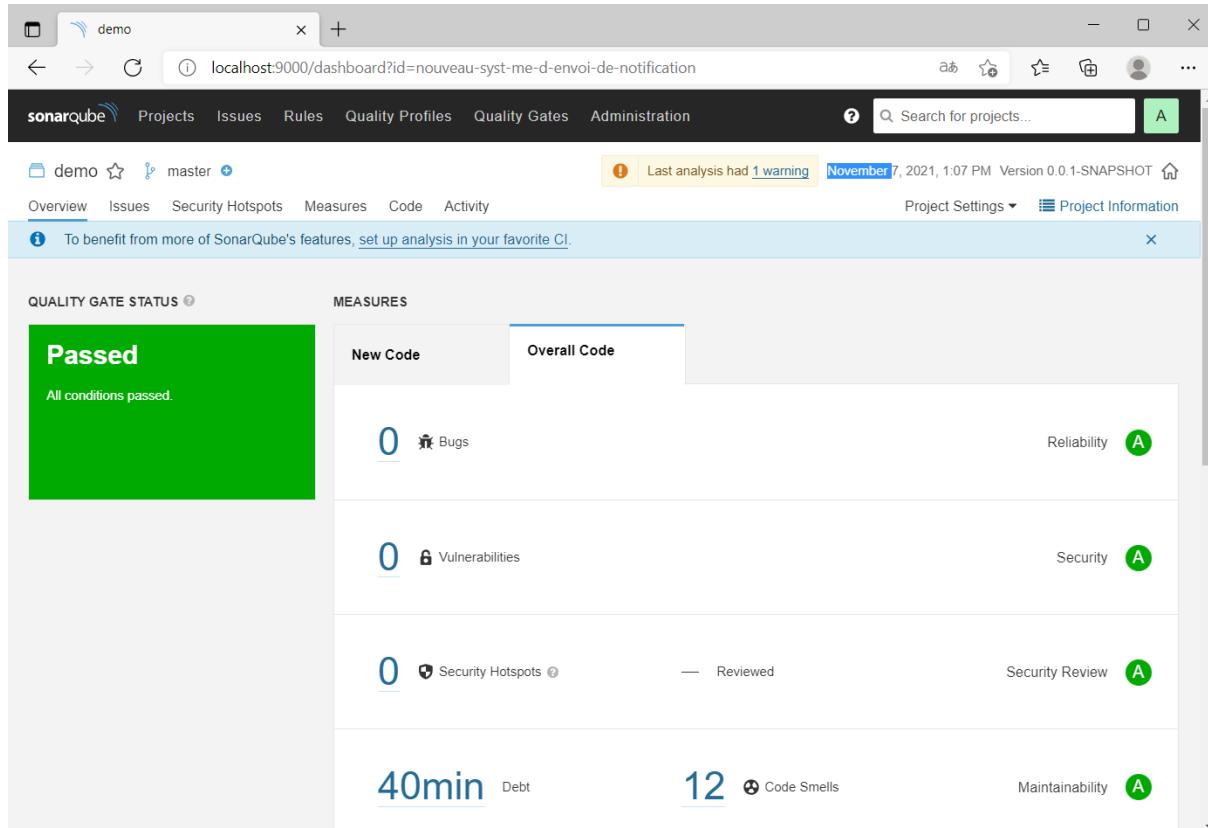


Figure 5.5. Résultats récapitulatifs d'analyse SonarQube

SonarQube classe les défauts logiciels selon 3 catégories :

- **Les bugs** : anomalies évidentes du code. Impactent la fiabilité (reliability) de l'application.
- **Les vulnérabilités** : faiblesses du code pouvant nuire au système. Impactent la sécurité de l'application.
- **Les « code smells »** : anti-patrons (ou anti-patterns). Impactent la maintenabilité de l'application.

La mention « Passed » signifie que le projet satisfait les exigences minimum définies dans l'espace « Barrières qualité ». Les paramètres par défaut peuvent être adaptés. Pour plus de détails sur les définitions des différentes métriques, veuillez consulter le lien de [la documentation officielle](#).

5 RÉALISATION

5.2.5 Benchmark

Dans le cadre de notre étude, le Benchmark réalisé vise à donner une idée sur le temps d'exécution et sa relation avec les méthodes déployées lors de l'exploration des données ainsi que par rapport aux nombres de threads exécutés, il nous permettra par la suite de comparer les pratiques de développement logiciel instaurées dans l'ancien système à celles adoptées dans le nouveau système.

Afin d'obtenir des résultats significatifs, nous allons tester les deux systèmes d'envoi de notifications avec un volume de données similaire (**1052 enregistrements**) et nous mesurerons ensuite le temps d'exécution nécessaire par les deux systèmes pour compléter le traitement de ce volume. Le tableau suivant (voir le tableau 5.5) précise les résultats des différentes expériences réalisées ainsi que le temps d'exécution écoulé pour chacune de celles-ci.

Table 5.5 : Comparaison des délais d'exécution entre l'ancien et le nouveau système d'envoi de notifications

Système	Type de données d'entrée	Occurrences	Nombre de threads	Environnement d'exécution	Temps d'exécution
Ancien	Requête Solr (objet JSON en retour)	53 appels	1	Production	9min2s
Nouveau	Fichier CSV	1 fichier	1	Local	1min17s
Nouveau	Fichier CSV	10 fichiers	10	Local	21s978ms

Synthèse

Bien qu'il semble inapproprié de faire ce benchmark, étant donné qu'il existe certaines variables qui pourraient altérer les résultats des tests que nous n'avons pas pris en compte, en particulier l'environnement d'exécution de chacun des systèmes qui était supposé avoir un effet négligeable, ce benchmark a montré qu'il existe un écart important dans les délais obtenus. En effet, le temps d'exécution est passé de **9min2s** dans l'ancien système à **1min17s**, soit **7 fois** plus rapide, et cela, en utilisant uniquement le mode monothread dans le nouveau système ce qui témoigne déjà des améliorations apportées par ce dernier par rapport à l'ancien. En répartissant ce volume sur dix fichiers et en activant le mode multithreading fixé à 10 threads maximum, le temps d'exécution diminue à nouveau vers **21s987ms**, soit **25 fois** plus rapide que l'ancien système, ce qui prouve cette fois, l'efficacité et la rentabilité du bon usage du système grâce aux paradigmes du multithreading.

Conclusion

Dans ce chapitre, nous avons exposé les outils de développement et les différents composants logiciels et matériels étudiés durant toute la période du stage, puis dans une deuxième partie, nous avons abordé la démonstration de l'un des travaux assignés, à savoir le système de notification tout en spécifiant le processus suivi partant de la configuration, l'exécution, les tests unitaires à l'audit de code. En fait, nous nous sommes efforcés d'implémenter le travail conformément à la conception et aux pratiques préconisées. Enfin, avec ce chapitre, nous terminons la phase d'implémentation.

Conclusion générale

Dans l'ensemble, je retiens un bilan positif de mon stage de fin d'études : J'ai pu intégrer une équipe de recherche et développement (R&D) polyvalente travaillant principalement autour des technologies Java et JavaScript. Cela m'a donné une idée concrète des exigences de Cegedim en matière d'ingénierie de développement orienté objet, ainsi que sur les règles et les bonnes pratiques de la gestion de projet agile, allant de la phase de recueil des besoins jusqu'à celle de la livraison du produit final.

La formation acquise dans le cadre de mon Master à la Faculté des Sciences de Rabat, notamment le module d'ingénierie du développement logiciel, m'a été très utile dans diverses situations que ce soit sur le plan conceptuel (élaboration des besoins, étude de faisabilité, conception générale, etc.) ou sur le plan technique (codage, complexité du code, bonnes pratiques de développement, etc.) ou encore sur le plan relationnel (travail collaboratif, planification et gestion de projets, etc.).

Ce stage a été très enrichissant pour moi, car il m'a permis de découvrir le monde du SIRH, un domaine vaste, nécessitant à la fois un esprit d'analyse sceptique et de solides compétences techniques. De surcroît, cette expérience m'a permis de participer concrètement à ses enjeux à travers des missions de développement et de refonte.

Ce rapport présente les différentes tâches réalisées durant ma période de stage, à savoir la migration du socle technique et le refactoring du code ainsi que la refonte du système délégué d'envoi des notifications. L'élaboration de ce rapport a été entamé par une présentation de l'organisme d'accueil. Par la suite, une étude de l'existant a été réalisée afin d'identifier les problèmes et d'exposer les solutions et les axes d'amélioration. Après cela, une étude conceptuelle détaillée des différentes fonctionnalités a été mise en place de même que la définition des besoins et les contraintes du projet. Enfin, ce rapport se termine par l'exposition des outils et de l'environnement de travail et aussi par une démonstration sur une solution réalisée pour remédier aux inconvénients de l'ancien système d'envoi de notifications.

Au terme de ce mémoire, je pense que ce stage en entreprise m'a offert une bonne préparation pour mon insertion professionnelle, car il fut une expérience enrichissante et complète qui conforte le désir d'exercer un futur métier dans le domaine de l'informatique. D'autre part, l'environnement professionnel dans lequel s'est déroulé ce stage m'a fait découvrir de nouvelles connaissances et compétences et a renforcé mon esprit d'équipe et mon professionnalisme. Cette expérience a aiguisé mes capacités d'analyse et de synthèse et a surtout renforcé mon esprit critique, ma motivation et mon ambition.

CONCLUSION GÉNÉRALE

Enfin, je tiens à exprimer ma satisfaction d'avoir pu travailler dans de bonnes conditions matérielles et un environnement agréable.

ANNEXES

A.1 Présentation des principales interfaces d'Arkevia

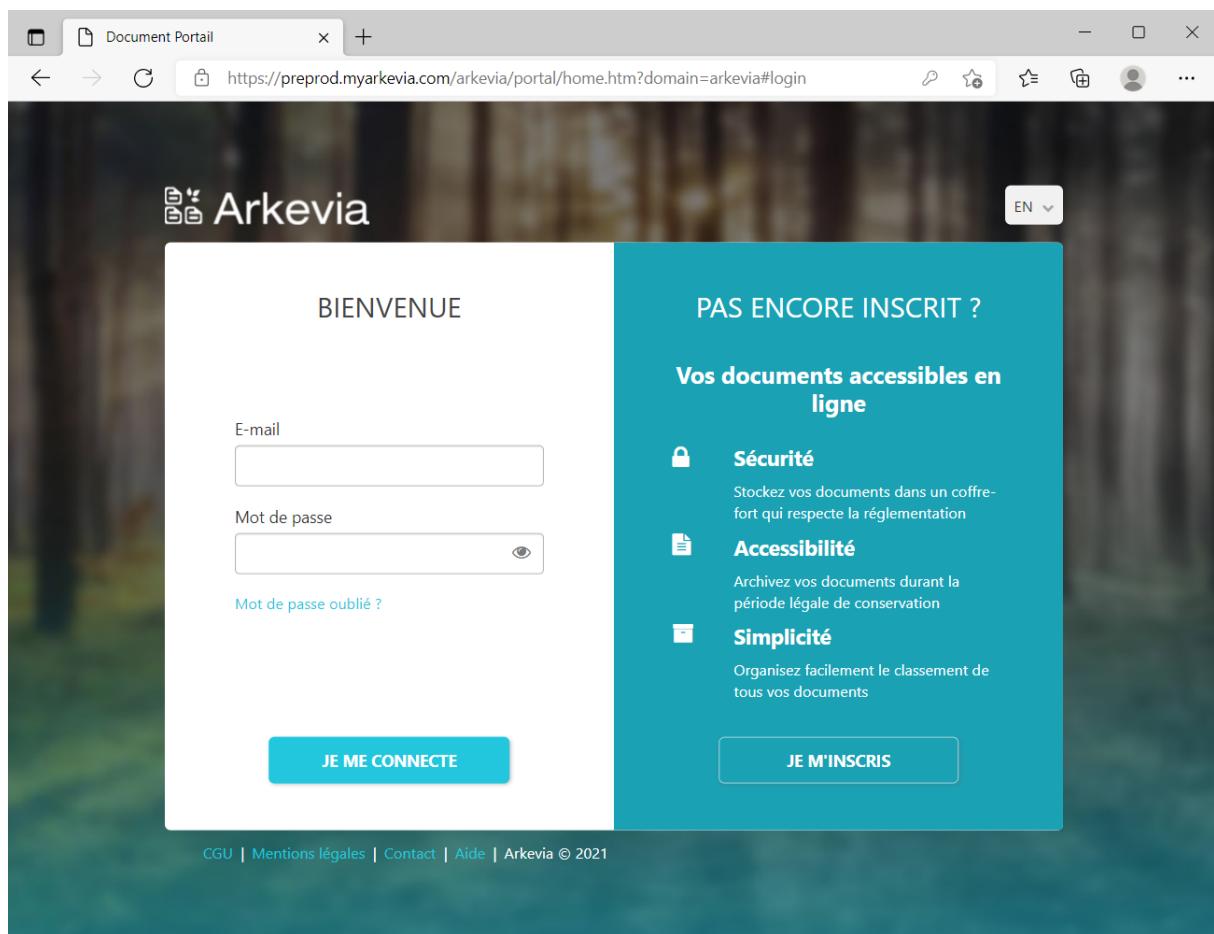


Figure A.1. Page d'authentification Arkevia

ANNEXES

The screenshot shows a web browser window titled "Document Portail" displaying the Arkevia portal at <https://preprod.myarkevia.com/arkevia/portal/home.htm?domain=arkevia#gedlist>. The user is logged in as hamza.talaghzi@cegedim-srh.com (Hamza Talaghzi). The interface includes a sidebar with navigation links like "Mes documents", "Partage", and "Boîte de réception". The main area shows a list of files in the "Boîte de réception" folder. One file, "ant-jmeter-1 1_1.jar", is selected, showing its details: Author: Hamza Talaghzi, Date: 28/08/2021. There are buttons for "Déposer un document", "Gérer les colonnes", "5 lignes", and "Filtrer". The bottom of the page contains footer links for CGU, Service commercial CEGEDIM, Mentions légales, cegedim.fr, contact, Aide, and CegeDIM © 2021.

Figure A.2. Page de gestion des documents

ANNEXES

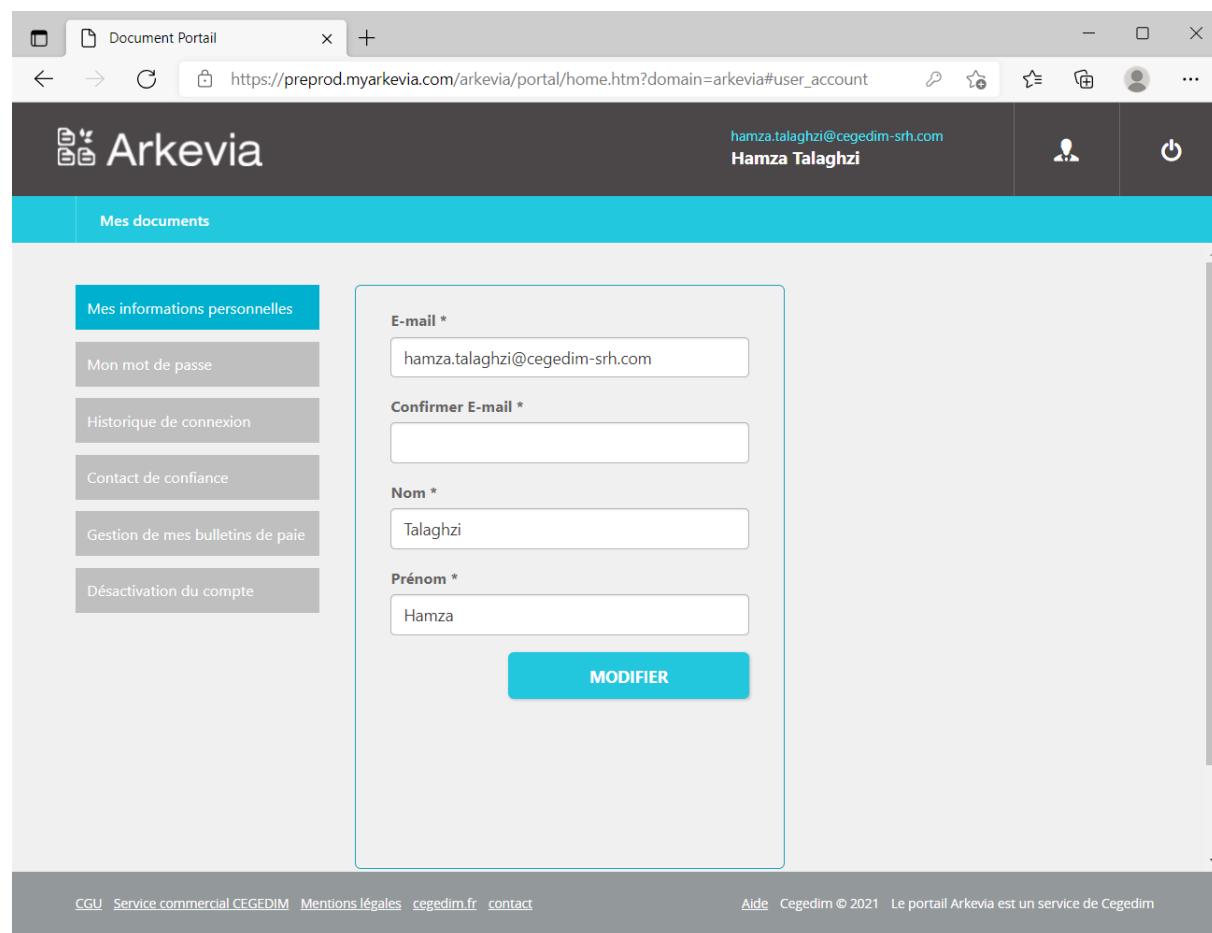


Figure A.3. Page de gestion de compte Arkevia

A.2 Audit de code : checklist élaborée par Cegedim SRH

Table A.1 : Extrait des règles et pratiques de développement logiciel instaurées par Cegedim SRH

Réf.	Description de la règle
RG_DEV_GN_001	Single responsibility principle : Une fonction doit être responsable d'une et d'une seule tâche. Lorsqu'une fonction fait plus d'une tâche, elle est plus difficile à composer, à tester et à comprendre.

Suite à la page suivante

ANNEXES

Table A.1 : Extrait des règles et pratiques de développement logiciel instaurées par Cegedim SRH (Suite)

RG_DEV_GN_002	Don't repeat yourself : Souvent, nous avons du code en double parce que nous avons deux ou plusieurs choses légèrement différentes, qui partagent beaucoup en commun, mais leurs différences nous obligent à avoir deux ou plusieurs fonctions distinctes qui font en grande partie les mêmes choses. Supprimer le code en double signifie créer une abstraction qui peut gérer cet ensemble de choses différentes avec une seule fonction / module / classe. Factoriser le code en double.
RG_DEV_GN_003	Open/Closed principle : Les nouvelles fonctionnalités ne doivent pas modifier le code existant.
RG_DEV_GN_004	Test unitaire : Chaque fonction doit être testée avec au minimum un cas passant et un cas non passant.
RG_DEV_GN_005	Respect des warnings eslint sonarLint : Veiller au maximum à ce qu'il n'y a pas de warnings eslint et/ou sonarlint dans la classes/ou parties de classes impactées dans le cadre d'une évolution, l'objectif est d'atteindre 0 Warnings dans le possible.
RG_DEV_GN_006	Correction des NullPointerException : Éviter de contourner le problème de nullité en ajoutant des tests, et aller plus en profondeur dans le diagnostic pour détecter l'origine du problème et le résoudre d'une manière radicale.
RG_DEV_GN_007	Arguments de fonctions : Limiter le nombre de paramètres d'une fonction est extrêmement important, un ou deux arguments est le cas idéal, et trois devraient être évités si possible. Rien de plus que cela devrait être utilisé. Utiliser des objets si nous avons besoin de beaucoup d'arguments.
RG_DEV_GN_008	Cas des dépassements de capacité : Éviter de contourner le problème en tronquant des valeurs, et aller plus en profondeur dans le diagnostic pour détecter l'origine du problème et le résoudre d'une manière radicale.

A.3 Outils utilisés pour la réalisation de ce rapport

Ce rapport est basé sur un modèle de rapport pédagogique conçu pour le département d'informatique de l'Université d'Avignon, par son auteur Vincent Labatut.

Pour obtenir ce modèle, veuillez visiter le site suivant : <https://fr.overleaf.com/latex/templates/modele-de-rapport-avignon-universite/pdbgdpzsgwrt>.

Le rapport est compilé à l'aide de **LuaLaTeX**, une variante de **LaTeX** utilisant le langage de script **Lua**. **LuaTeX** a été choisi comme successeur de **pdfTeX**. La structure générale et la plupart des commandes sont similaires à celles de LaTeX.

Il a également été utile d'utiliser certains supports pour produire les ressources et les illustrations incluses dans le rapport. Le tableau suivant rassemble les différents outils utilisés dans la mise en œuvre de ce rapport.

Table A.2 : Supports utilisés pour la réalisation de ce rapport

 Adobe Illustrator	Adobe Illustrator est un logiciel de création graphique vectorielle. Il fait partie de la gamme Adobe, peut être utilisé indépendamment ou en complément de Photoshop, et offre des outils de dessin vectoriel puissants. Les images vectorielles sont constituées de courbes générées par des formules mathématiques.
 Draw.io	Draw.io est une application de création de diagrammes et schémas sous licence Apache disponible sous Windows, macOS, Linux, sous forme d'application web et intégrée à des services cloud tels Nextcloud ou Google Drive.
 Figma	Figma est un outil collaboratif de conception et de prototypage d'interfaces.
 Visual Studio Code	Visual Studio Code est un éditeur de code source puissant, développé par Microsoft. Il est disponible pour Windows, Mac et Linux.

Références

- [1] Dino Mandrioli **Carlo Ghezzi** Mehdi Jazayeri. *Fundamentals of Software Engineering*. 2^e éd. Pearson, Prentice Hall, 2003, p. 1-5. **isbn** : 0133056996 , 8120322428.
- [2] Jacques **CHAUMIER**. « Gestion électronique de documents ». In : CACALY, Serge et al. *Dictionnaire encyclopédique de l'information et de la documentation* Nouvelle édition. Paris : Nathan (2001), p. 250-252.
- [3] *Chiffre d'affaires (CA) et bilan CEGEDIM CGM Euronext Paris*. Boursorama. **url** : <https://www.boursorama.com/cours/1rPCGM/>.
- [4] L'équipe produit **CMMI**. *CMMI® pour le développement, Version 1.3 - Amélioration des processus pour le développement de meilleurs produits et services*. 2011. **url** : https://resources.sei.cmu.edu/asset_files/WhitePaper/2010_019_001_28803.pdf.
- [5] **Consultancy.eu**. *Half of companies applying Agile methodologies & practices*. consulté le 26 septembre 2021. Organize Agile, Consultancy.eu analysis. **url** : <https://www.consultancy.eu/news/4153/half-of-companies-applying-agile-methodologies-practices>.
- [6] Jacques **PRINTZ**. *LOGICIELS*. consulté le 26 septembre 2021. Encyclopædia Universalis [en ligne]. **url** : <https://www.universalis.fr/encyclopedie/logiciels/>.
- [7] Jean-Claude **Rault**. *La Lettre d'ADELI n°76 – Été 2009*. consulté le 26 septembre 2021. ADELI [en ligne]. **url** : <https://espaces-numeriques.org/wp-content/uploads/2019/01/176p08.pdf>.
- [8] Véronique MESSAGER **ROTA**. *Gestion de projet - Vers les méthodes agiles*. Eyrolles, 2008. **isbn** : 978-2-212-12165-0.
- [9] Badr **TALAGHZI**. « Processus du développement logiciel - Paradigmes & Perspectives ». In : *Omn.univ.europ*. Editions universitaires européennes, 2016. **isbn** : 9783639543124, 978-3639543124.

Résumé

Ce rapport synthétise l'ensemble des travaux réalisés au cours du stage. La première phase de ces travaux a consisté à auditer et analyser l'architecture d'un portail de gestion électronique de documents appelé **Arkevia**, puis à identifier et corriger les problèmes et anomalies détectés, ainsi qu'à proposer des axes d'amélioration suite à l'audit architectural réalisé. Ensuite, dans une seconde phase, les travaux ont porté sur le mécanisme de notification chargé d'informer les utilisateurs des documents récemment déposés dans leurs coffres-forts Arkevia. En effet, étant donné l'importance que porte Cegedim à la qualité de ses solutions, il a été convenu de refondre ce mécanisme et de le rendre indépendant de l'application mère Arkevia. L'application produite est principalement basée sur une approche de traitement par lots multithread via les technologies Spring Boot et Spring Batch.

Mot clés : spring, batch, processing, sirh, ged.

Abstract

This report summarizes all the work done during the internship. The first phase of the work involved auditing and analyzing the architecture of an electronic document management portal called **Arkevia**, then identifying and repairing detected problems and anomalies, as well as suggesting new areas of enhancement based on the architectural review. In a second phase, the work focused on the notification mechanism responsible for informing users of documents recently deposited into their Arkevia safes. Indeed, given the importance that Cegedim places on the quality of their solutions, it was agreed to overhaul this mechanism and make it independent of the Arkevia parent application. The resulting application is primarily based on a multi-threaded batch approach using Spring Boot and Spring Batch technologies.

Keywords : spring, batch, processing, sirh, ged.