



Rapport de Stage

Outil de gestion des licences logiciel

IUT de Nantes

Arthur MILLET

Tuteur Universitaire : Sébastien FAUCOU

Maître de stage : Corentin LHERIAUD

13 juillet 2024

Remerciements

Je tenais tout d'abord à exprimer ma plus profonde gratitude envers Manitou-Group qui m'a offert l'opportunité d'intégrer l'entreprise en tant que stagiaire, ce qui fût et restera, pour moi, une expérience des plus enrichissante et formatrice.

Je remercie particulièrement Monsieur Xavier GALLARD, le manager de l'équipe R&D IS, pour la confiance qu'il m'a accordée tout au long de cette période. Son encadrement, ses conseils avisés et sa disponibilité ont été déterminants pour mon apprentissage et m'ont permis de développer des compétences cruciales dans mon domaine d'études.

Je souhaite également exprimer ma reconnaissance la plus sincère à Monsieur Corentin LHE-RIAUD, mon maître de stage et développeur de l'équipe R&D IS, pour son soutien sans faille. Sa patience, son partage de connaissances et son accompagnement constant ont grandement facilité mon intégration au sein de l'entreprise. Grâce à lui, j'ai pu comprendre en profondeur le fonctionnement de Manitou et me familiariser avec ses processus et ses valeurs.

Je remercie chaleureusement Monsieur Julien JUHEL et Monsieur Oliver GUYOT, membres de l'équipe R&D IS, qui m'ont également aidé à comprendre le fonctionnement de l'entreprise. La présentation de leur métier respectif de Business Analyst et d'Expert Support m'a permis de découvrir d'autres domaines passionnants de l'informatique. Leur accompagnement dans le projet, notamment pour les tests et la planification, a été d'une grande aide et m'a permis de développer de nouvelles compétences pratiques.

Je tiens aussi à remercier toute l'équipe de Product Management pour l'accueil chaleureux que j'ai reçu. Cette expérience a été des plus positives et restera un moment clé de mon parcours professionnel.

Je souhaite exprimer ma gratitude envers mon tuteur universitaire, Monsieur Sébastien FAUCOU, qui m'a accompagné tout au long de ce stage. Sa disponibilité et ses précieux conseils ont été d'un grand soutien et m'ont aidé à surmonter les défis rencontrés durant cette période.

Enfin, je tenais à remercier les professeurs de l'IUT de Nantes pour leurs enseignements qui m'ont été salvateurs durant toute la durée du stage et qui le seront également pour la suite de ma vie professionnelle.

Résumé

This document aims to present the objectives, steps, and results of the internship I completed at Manitou-Group as part of my second year of the BUT Informatique program at IUT de Nantes.

The R&D IS team of the company manages a large number of license management tools, a situation complicated by the constant growth in the number of employees. This expansion leads to an increasing need for software licenses, making it essential to optimize the entire set of licenses to avoid unnecessary costs and ensure efficient use of resources.

To address this challenge, the company launched a project to develop an application capable of retrieving, standardizing, and consolidating data from all the license management tools used, then displaying them in the form of charts and tables to efficiently supervise all licenses within the R&D IS scope.

The main objective of this internship is therefore to create this application, which will allow for a comparison between the number of licenses purchased and those actually used and to obtain an almost instantaneous view of the users, thereby facilitating more effective license management and allocation.

This document aims to present the objectives, steps, and results of the internship I completed at Manitou-Group as part of my second year in the Computer Science program at the IUT of Nantes.

The company's R&D IS team manages a large number of license management tools, a situation complicated by the continuous growth in the number of employees. This expansion results in an increasing need for software licenses, making it essential to optimize the entire set of licenses to avoid unnecessary costs and ensure efficient use of resources.

To address this challenge, the company launched a project to develop an application capable of retrieving, standardizing, and consolidating data from all the license management tools used, then displaying it in the form of graphs and tables to efficiently supervise all licenses within the R&D IS scope.

The main objective of this internship is to create this application, which will allow for comparing the number of purchased licenses with the number of licenses actually used, and to obtain an almost instant view of the users, thereby facilitating more efficient management and allocation of licenses.

Toutes les expressions suivies d'un astérisque sont définies dans le glossaire

Table des matières

Table des matières	3
1 Contexte et introduction de la mission	7
1.1 Manitou-Group	7
1.1.1 Présentation de l'entreprise	7
1.1.2 Stratégie de l'entreprise	7
1.1.3 Constitution et fonctionnement du département IT&D	8
1.1.4 Stratégie de l'IT&D	9
1.1.5 L'équipe de Recherche et Développement et Système d'Information (R&D IS)	9
1.2 Présentation du stage	9
1.2.1 Énoncé du besoin	9
1.2.2 Présentation du problème	10
1.2.3 Présentation de la mission	10
1.3 Présentation de l'environnement de travail	10
1.3.1 Environnement physique	10
1.3.2 Culture et événements internes	11
1.4 Présentation de l'environnement de travail informatique	12
1.4.1 Infrastructure informatique	12
1.4.2 Infrastructure technique	12
1.4.3 Aspects organisationnels et initiatives stratégiques	13
1.5 Expérience personnelle sur l'environnement de travail	14
2 Phase de conception du projet	15
2.1 Précision du besoin	15
2.2 Contraintes	15
2.3 Choix des technologies	15
2.3.1 Powershell :	15
2.3.2 Talend :	16
2.3.3 Serveur FTP	16
2.3.4 PowerBi :	17
2.4 Brainstorming et collecte d'idées	17
2.4.1 Objectif du brainstorming	17
2.4.2 Méthodologie	17
2.5 Réflexion personnelle :	17
3 Phase de planification du projet	18
3.1 Priorisation et définition du périmètre	18
3.1.1 Évaluation des idées	18
3.1.2 Définition du périmètre	18
3.2 Faisabilité technique et complexité	18

3.2.1	Analyse de complexité	18
3.2.2	Rejet des idées impossibles	18
3.3	Idées ressorties	19
3.4	Conclusion sur la phase de planification	19
4	Extraction des données des outils de gestion	21
4.1	Spécifications	21
4.1.1	Objectif de l'extraction	21
4.1.2	Contexte	21
4.1.3	Contrat d'échange	22
4.1.4	Architecture de l'application	22
4.2	Processus d'extraction des données	22
4.2.1	Transformation des données	23
4.2.2	Mise à disposition des données	24
4.3	Problèmes et solutions	25
4.3.1	Problèmes rencontrés et solutions apportées	25
4.4	Tests	26
4.4.1	Exemple de test structurel d'une fonction	26
4.5	Conclusion et réflexion personnelle	27
5	Intégration des données	28
5.1	Objectifs et contexte	28
5.1.1	Objectif de l'intégration	28
5.1.2	Contexte	28
5.2	Transfert des données vers le serveur FTP	28
5.2.1	Objectif du transfert	28
5.2.2	Processus de transfert	28
5.3	Processus d'intégration	29
5.3.1	Choix du dépôt des données	29
5.3.2	Google Cloud Platform	30
5.3.3	Les Jobs Talend	30
5.3.4	Présentation des jobs réalisés	31
5.4	Problématiques et solutions	32
5.4.1	Problèmes rencontrés et solutions apportées	32
5.5	Conclusion et réflexion personnelle	32
6	Consolidation des données	34
6.1	Objectifs et contexte	34
6.1.1	Objectif de la consolidation	34
6.1.2	Contexte	34
6.2	Framework Dbt	34
6.2.1	Présentation de Dbt	34
6.2.2	Justification du choix de Dbt	35
6.3	Processus de consolidation	35
6.3.1	Structuration des données	35
6.3.2	Utilisation de dbt	36
6.3.3	Architecture du framework	36
6.3.4	Configuration du projet	36
6.3.5	Fonctionnement global :	36
6.3.6	Fonctionnalités utilisées	37
6.3.7	Fonctionnalités utilisables dans le futur	37

6.4	Conclusion et réflexion personnelle	38
7	Création du rapport	39
7.1	Objectifs et contexte	39
7.1.1	Objectif du rapport Power BI	39
7.1.2	Contexte	39
7.2	Processus de création du rapport	39
7.2.1	Conception des visualisations	39
7.2.2	Personnalisation et interactivité	40
7.3	Problèmes rencontrés et solutions apportées	40
7.3.1	Données Ambiguës	40
7.4	Documentation	40
8	Déploiement de la solution	41
8.1	Mise en Qualif	41
8.2	Mise en production	41
8.2.1	Scripts de déploiement	42
9	Conclusion et Réflexion Personnelle sur l'ensemble du stage	43
9.1	Résultats	43
9.2	Améliorations futures	43
9.3	Conclusion personnelle sur l'ensemble du stage	44
9.4	Compétences Acquisées	44
9.5	Perspective de carrière	45
10	Sources	46
10.1	Sources publiques	46
10.2	Sources Internes	46
	Glossaire	46
A	Cahier des charges	49
A.1	Contexte et présentation de la mission	49
A.2	Acteurs Impliqués	49
A.3	Confidentialité	49
A.4	Livrables	50
B	Informations supplémentaires sur l'extraction des données	51
B.1	Présentation de l'extraction pour un outil :	51
B.1.1	Contexte	51
B.1.2	Contexte du projet Java	52
B.1.3	Architecture du projet Java :	52
B.1.4	Illustrations du code	53
B.1.5	Présentation des scripts PowerShell	53
C	Informations supplémentaires sur le serveur FTP	57
C.1	Les Web Requests en PowerShell	57
C.1.1	La Classe <code>System.Net.FtpWebRequest</code>	57
C.1.2	Principales Méthodes et Propriétés	57
C.1.3	Processus de Transfert de Fichiers FTP en PowerShell	58
C.1.4	Exemples d'utilisations	58

D Informations supplémentaires sur Talend	60
D.1 Normes de développement	60
D.2 Talend Context	60
D.3 Composants Talend	61
D.4 Autres jobs développés	61
E Informations supplémentaires sur dbt	63
E.0.1 Fichier de configuration	63
E.0.2 Exemple de modèle	63
E.0.3 Documentation	64
E.0.4 Tests	65
F Présentation et rendu des rapports Power BI	66
F.1 Softwares et informations sur la licence (Page 1)	66
F.2 Licences utilisées au maximum (Page 2)	67
F.3 Statistiques de licences (Page 3)	68
F.4 Vue par période (Page 4)	68
F.5 Utilisateurs connectés (Page 5)	69
F.6 Tous les utilisateurs (Page 6)	70
F.7 Informations sur les fournisseurs (Page 7)	71
G Déploiement	72

Chapitre 1

Contexte et introduction de la mission

1.1 Manitou-Group

1.1.1 Présentation de l'entreprise

Création de Manitou

En 1958, Marcel Braud conçoit un engin révolutionnaire en inversant le schéma d'un tracteur agricole, y ajoutant un mât de levage et une direction hydraulique. Ainsi naît le premier chariot élévateur tout-terrain, jetant les bases du succès de l'entreprise et de son rayonnement mondial. Il fallait choisir un nom pour ce chariot capable de tout manipuler et c'est ainsi qu'est née la marque Manitou.

Secteur d'activité et produits

Manitou-Group est une entreprise française basée à Ancenis qui conçoit, produit, distribue et assure le service de matériels de manutention destinés aux secteurs de la construction, de l'agriculture et de l'industrie.

Présence mondiale

En 1972, Manitou ouvre sa première filiale internationale à Chandlers Ford, près de Southampton en Angleterre, marquant ainsi son entrée sur la scène mondiale. Aujourd'hui, Manitou Group compte 32 filiales à travers le monde, de Hangzhou à Dubaï, en passant par New Delhi et la Finlande.

Manitou en chiffre

Aujourd'hui Manitou compte 4 pôles en France, 2 aux Etats-Unis, 1 en Italie et 1 en Inde. C'est également 5500 collaborateurs dans plus de 140 pays avec un chiffre d'affaires de plus de 1,6 milliards d'euros en 2020. Dans le cadre de son plan stratégique New Horizons 2025, Manitou prévoit une enveloppe d'investissements de 460 millions d'euros pour propulser son développement futur.

1.1.2 Stratégie de l'entreprise

En 2021, Manitou Group s'est fixé de nouvelles priorités stratégiques. Ambitieux en termes d'objectifs de croissance durable et d'investissements, ce plan repose sur quatre axes majeurs énoncés dans la feuille de route "New Horizons 2025" :

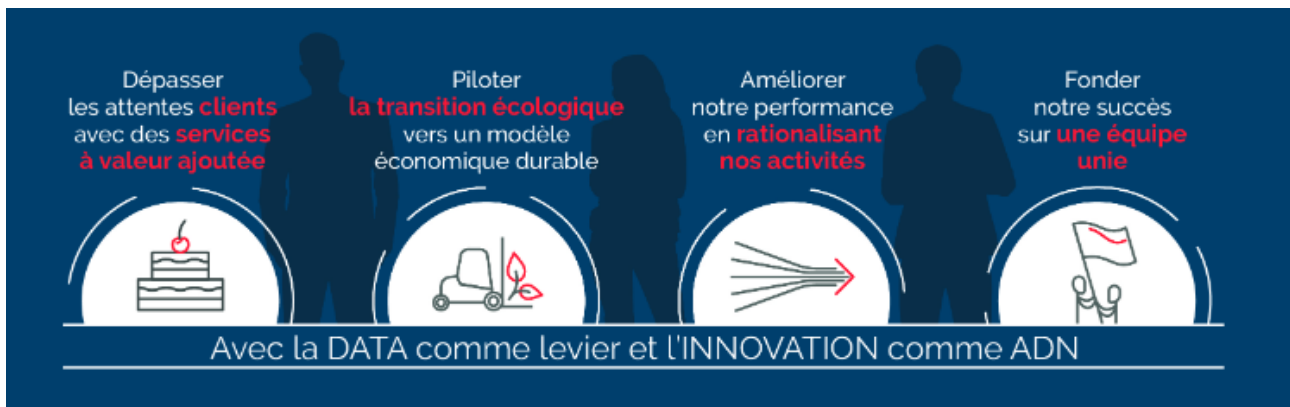


FIGURE 1.1 – Feuille de route New Horizons 2025

1.1.3 Constitution et fonctionnement du département IT&D

Le département Information Technology & Digital fût créé en juillet 2021 suite à la réorganisation de L'ISIT* pour répondre aux besoins de gouvernance et de communication entre le métier et l'entreprise. L'IT&D est donc un département de support pour les métiers composé de 6 équipes

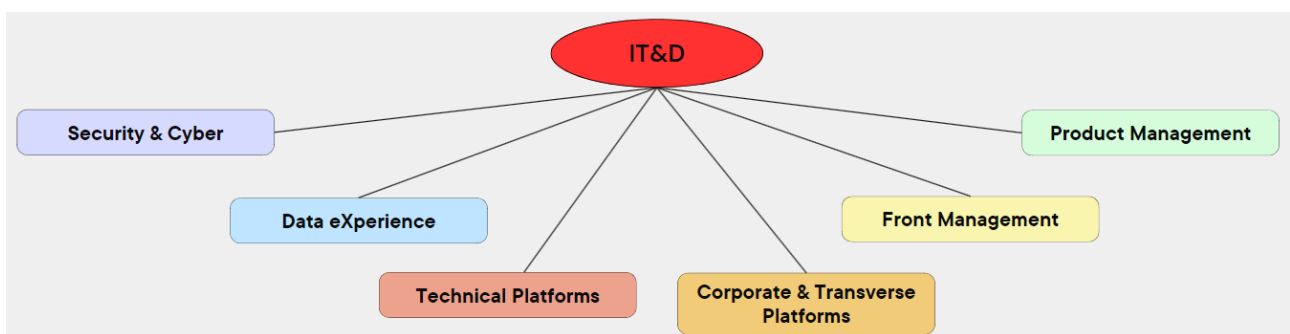


FIGURE 1.2 – IT&D Org Chart

La suite du rapport sera centrée sur l'équipe R&D IS, de l'équipe Product Management, car c'est celle dans laquelle j'évolue.

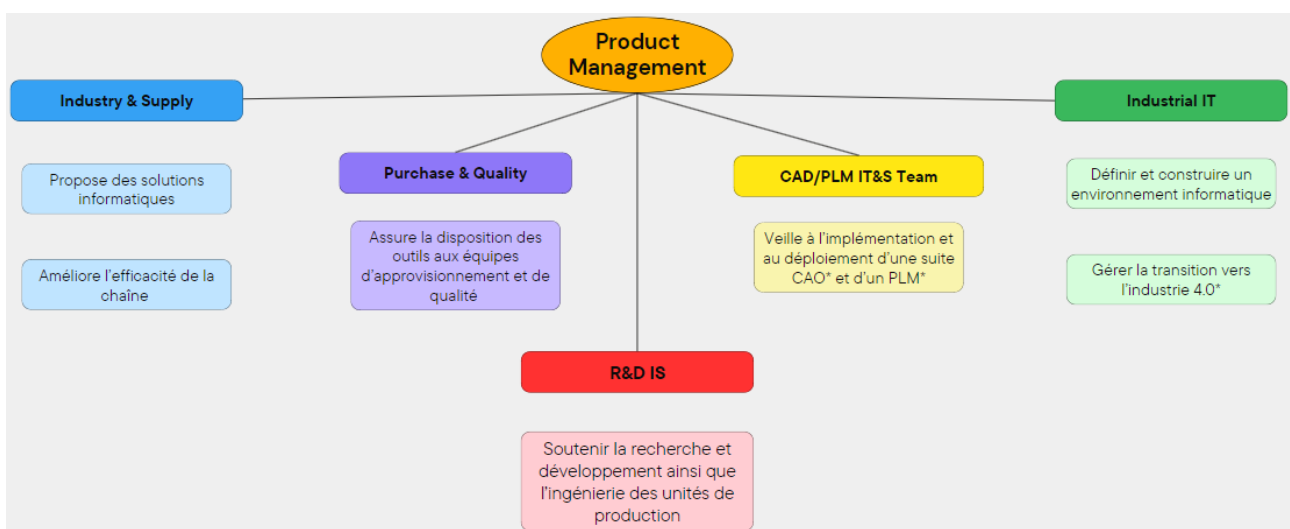


FIGURE 1.3 – Org Chart de l'équipe Product Management

L'objectif de l'équipe Product Management est d'assurer le lien entre la production de machine et le système d'information. Elle est donc présente en qualité de support pour l'équipe Product de Manitou.

1.1.4 Stratégie de l'IT&D

Afin de développer et d'entretenir le système d'information du groupe, l'IT&D se base sur 6 grands principes stratégiques :

- Améliorer l'expérience numérique de ses clients
- Compléter le déploiement de son ERP* commun
- Changer la culture du Groupe en matière de Digital
- Déployer, globaliser et maintenir ses plateformes clés
- Consolider, structurer et mieux gouverner ses données
- Acquérir les briques technologiques et de sécurité manquantes

1.1.5 L'équipe de Recherche et Développement et Système d'Information (R&D IS)

L'objectif de l'équipe de R&D IS est donc de soutenir l'équipe de Recherche et Développement côté métier. Cela signifie donc :

- **Fournir des outils et technologies avancés** : Développer et mettre en place des systèmes d'information pour améliorer les processus de R&D.
- **Améliorer l'efficacité et la productivité** : Optimiser les systèmes existants pour faciliter le travail et améliorer la productivité des équipes R&D.
- **Assurer l'intégration et l'alignement stratégique** : Veiller à ce que les solutions technologiques soient intégrées correctement avec les besoins spécifiques du métier R&D et qu'elles soient également en accord avec la stratégie globale de l'entreprise.

1.2 Présentation du stage

1.2.1 Énoncé du besoin

L'entreprise a besoin d'automatiser le processus de collecte et de stockage des données issues des différents outils de gestion des licences logicielles. Cette automatisation permettrait de générer des statistiques précises sur l'utilisation des licences, telles que le taux d'utilisation et le maximum d'utilisation. En comparant le nombre de licences achetées avec le nombre de licences effectivement utilisées, l'entreprise pourra optimiser ses coûts de licences, évitant ainsi de financer des licences inutilisées ou de sous-évaluer le nombre de licences utiles. En effet, disposer d'une vue quasi-instantanée des utilisateurs des licences permettra de réagir rapidement en cas de problème avec une licence. L'automatisation de ce processus libérera du temps pour les employés, leur permettant de se concentrer sur des tâches à plus forte valeur ajoutée.

1.2.2 Présentation du problème

L'entreprise utilise divers outils de gestion des licences logicielles, chacun générant des données dans des formats différents et non uniformes. Ces différences rendent difficile la collecte, le stockage et l'affichage de ces données. De plus, chaque outil ne renvoie pas toujours les mêmes types de données, ajoutant une couche de complexité supplémentaire. Il est donc impératif de trouver une solution pour uniformiser la récupération des données provenant de ces outils afin de les stocker et les afficher de manière standardisée. Cette uniformisation est essentielle pour faciliter l'analyse et l'interprétation des données, et assurer ainsi une gestion efficace des licences.

1.2.3 Présentation de la mission

En tant que stagiaire chez Manitou, ma mission est d'élaborer et de mettre en oeuvre une solution pour gérer les données des licences de l'entreprise. La mise en oeuvre de cette solution consiste en 3 grandes phases :

- **Récupération des données** : Extraire les données des licences fournies par le fournisseur indépendamment des particularités de chaque outil. Étant donné la diversité des outils de gestion des licences gérés par l'IT&D, il sera nécessaire de trouver une approche permettant d'extraire les données de manière cohérente pour chaque outil, malgré leurs variations de fonctionnement.
- **Consolidation des données récupérées** : Concevoir un système permettant de stocker les données dans un entrepôt afin d'en assurer leur disponibilité et leur fiabilité. L'objectif principal est de consolider les données de manière à maintenir leur pertinence et leur cohérence, tout en optimisant les performances et en préservant l'intégrité du stockage.
- **Visualisation des données récupérées** : S'assurer que les utilisateurs finaux bénéficient d'une visualisation aisée des données, répondant à une variété de besoins. Cela implique de concevoir des solutions flexibles, capables de présenter les informations de manière claire et compréhensible, tout en offrant une expérience utilisateur optimale.

Dans le cadre de la mission, il est important de documenter consciencieusement l'ensemble du travail réalisé. Cette responsabilité implique une attention particulière à la documentation des commits et du code. Chaque modification apportée au code sera soigneusement assortie d'un commentaire explicite, permettant de clarifier son objectif et son impact sur le projet. Cette approche méthodique contribue de manière significative à assurer la continuité et la pérennité du projet au sein de l'entreprise, garantissant ainsi sa compréhension et sa maintenabilité pour les futurs utilisateurs et contributeurs.

1.3 Présentation de l'environnement de travail

1.3.1 Environnement physique

Lieu de travail

Je suis basé au siège social de Manitou Group, situé à Ancenis, une ville Française de taille moyenne dans le département de la Loire-Atlantique, en région Pays de la Loire. Ancenis offre un environnement paisible et propice au travail. Cependant, en raison de sa taille moyenne, l'accès à la ville par les transports en commun est limité, ce qui peut influencer le processus

de recrutement. Afin de faciliter le trajet du personnel n'ayant pas de voiture, Manitou Group met à disposition une navette pour les employés reliant la gare à l'entreprise.

Bâtiments et installations

L'IT&D, en raison de sa croissance rapide de 40 à 150 personnes, a dû se répartir entre deux bâtiments, CL et L, un seul ne suffisant plus. Chaque bâtiment est équipé d'une salle de pause avec des machines à café et des distributeurs de nourriture et boissons. Les salles de réunion, de différentes tailles, sont équipées de webcams et d'écrans pour faciliter les visioconférences. Un restaurant interne est disponible pour le déjeuner. Bien que la délocalisation de l'IT&D ait été une option, il a été jugé plus avantageux pour les employés de rester sur le site de production pour maintenir un contact rapide avec les équipes. De plus, un bâtiment de co-working a été ouvert à Nantes pour rester dans la dynamique de recrutement.

Postes de travail

Dans le bâtiment CL, les bureaux sont aménagés suivant le principe des open space. En revanche, dans le bâtiment L, les open spaces sont légèrement plus petits. Chaque poste de travail est équipé de deux écrans ou d'un grand écran incurvé. Il y a très peu de bureaux attitrés car avec l'augmentation du télétravail, de nombreux bureaux restaient inoccupés au cours de la semaine. Pour remédier à cela, un mode flex a été instauré, permettant à chaque employé de choisir librement son emplacement de travail.

1.3.2 Culture et événements internes

Culture d'entreprise

La culture d'entreprise chez Manitou est profondément axée sur l'innovation, la recherche et la découverte de talents. Un objectif majeur de Manitou est de créer un environnement de travail où l'employé se sent à l'aise et motivé à accomplir ses tâches. Le concept d'open space et la présence de salles de pause encouragent les interactions entre employés, favorisant ainsi un esprit d'équipe dynamique et collaboratif.

Pour attirer les talents, Manitou propose divers avantages tels que des chèques vacances, un comité d'entreprise orienté sur la famille, ainsi que des places en crèche. Au sein de l'entreprise, un comité de vie au travail, et plus particulièrement un comité de vie à l'IT&D, organise des événements pour renforcer la cohésion, comme ceux mentionnés dans la partie suivante.

Pour favoriser l'entente et l'efficacité, les nouveaux cadres bénéficient d'une semaine d'immersion dans les usines afin de mieux comprendre le fonctionnement de Manitou. Cela permet de saisir que Manitou est avant tout axé sur la production de machines de manutention, et peu sur le développement logiciel.

Événements internes

Manitou organise régulièrement divers événements internes pour renforcer la cohésion d'équipe et promouvoir un environnement de travail positif, tels que des séminaires, des voyages d'entreprise et des activités sportives. Dans les équipes R&D IS et Product Management, plusieurs types de réunions sont régulièrement organisés. Les réunions quotidiennes ("daily") permettent de discuter des actions de la journée précédente, de planifier les besoins du jour et de résoudre les problèmes éventuels. Des points d'équipe bihebdomadaires offrent à l'équipe R&D IS l'occasion de discuter des aspects positifs et négatifs de la vie professionnelle. Des réunions hebdomadaires avec l'équipe Product Management permettent de planifier la semaine à venir et de maintenir une communication fluide entre les équipes. Enfin, des réunions mensuelles avec l'équipe

Product Management servent à définir les objectifs à venir et à faire le point sur les objectifs actuels, assurant une vision claire et alignée sur la stratégie de l'entreprise.

1.4 Présentation de l'environnement de travail informatique

1.4.1 Infrastructure informatique

Équipements utilisés

Lors de mon arrivée, l'entreprise m'a attribué un ordinateur portable ainsi qu'une souris. Un doc permet de se connecter aux écrans. En fonction du poste occupé et des besoins il est possible d'effectuer des demandes de matériel comme des casques de réalité virtuelle, des claviers...

Environnements

L'entreprise dispose de 3 environnements informatiques distincts :

- **L'environnement de Développement** : C'est l'environnement le plus isolé qui permet de faire toutes les créations et le développement de projet.
- **L'environnement de Production** : C'est l'environnement final et aucune modification n'est permise. Il s'agit d'un environnement où se situent les applications fonctionnelles.
- **L'environnement de Qualif** : Ce dernier est un environnement se situant entre celui de "développement" et celui de "production". Il va permettre d'effectuer des tests des applications avec les versions de l'environnement de production, permettant aux métiers de tester et ainsi d'être plus proche d'une réalité.

1.4.2 Infrastructure technique

Système d'exploitation

Le système d'exploitation utilisé par l'entreprise est Windows. Il s'agit d'un système d'exploitation utilisé depuis très longtemps par l'entreprise. Microsoft propose un service de support intégré dans les licences. Ce système est couramment utilisé et assez simple d'utilisation pour les non initiés.

Outils de développement

Environnements de développement L'environnement de développement intégré (IDE) utilisé est Visual Studio Code (VScode), qui est apprécié pour sa flexibilité et ses nombreuses extensions. D'autres outils, comme TalendStudio pour l'intégration de données et la Console Google pour la gestion des services Google Cloud, sont également employés.

Langages de programmation et frameworks Pour le développement backend, le langage C# est largement utilisé en raison de sa robustesse et de ses capacités d'intégration avec divers systèmes. Pour le frontend, TypeScript est privilégié, notamment avec le framework Angular, qui offre une structure solide et modulaire pour les applications web.

Adaptation aux projets Le choix des technologies et des langages dépend grandement des spécificités de chaque projet. C'est pourquoi l'entreprise dispose d'un architecte logiciel, M.PERROIN dont le rôle est de se tenir au courant des nouvelles solutions informatiques disponibles. Il aide à sélectionner les technologies les plus appropriées pour chaque projet, en veillant à rester à la pointe des innovations dans le domaine du développement logiciel.

Outils de collaboration L'entreprise utilise divers outils pour améliorer la collaboration et la communication interne. La suite Google Workspace est essentielle, offrant des applications telles que Google Drive pour le stockage et le partage de documents, Google Docs et Google Sheets pour l'édition collaborative en temps réel, et Google Meet pour les visioconférences.

Pour la gestion de versions, Git est l'outil principal utilisé. Il permet de suivre les modifications du code source, de gérer les branches de développement, et de faciliter la collaboration entre les développeurs.

De plus, l'entreprise utilise Azure DevOps principalement pour l'intégration et le déploiement des applications.

Pourquoi Azure DevOps et pas simplement Git ?

Azure DevOps fournit des outils puissants pour l'intégration continue et le déploiement continu, ce qui permet d'automatiser et de gérer efficacement le processus de déploiement. De plus Azure DevOps propose d'autres avantages significatifs :

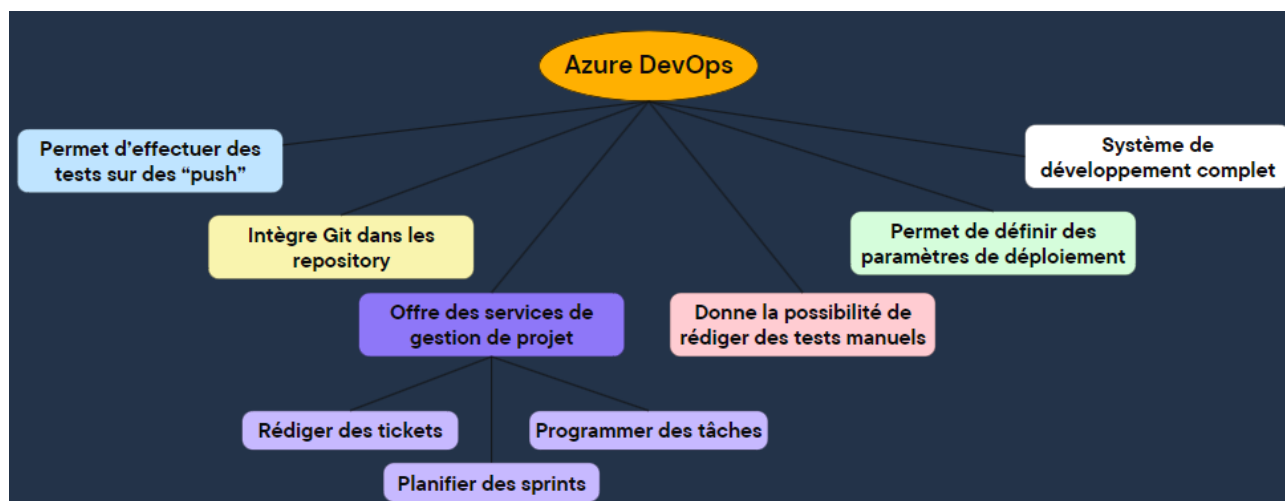


FIGURE 1.4 – Services proposés par Azure DevOps

1.4.3 Aspects organisationnels et initiatives stratégiques

Politiques de sécurité

Au sein de Manitou Group, une authentification à double facteur* est en place, cependant, la sécurité repose principalement sur les outils utilisés, étant donné que Manitou est principalement un intégrateur et non un développeur.

Formation et sensibilisation

L'entreprise organise des formations sur la sécurité informatique, en mettant particulièrement l'accent sur les aspects liés au phishing*. De plus, des conférences sont organisées sur différents thèmes comme, par exemple, la gestion des risques et le droit à l'erreur.

Responsabilité sociétale et environnementale

L'entreprise s'engage activement dans des actions environnementales pour réduire son impact écologique. Cela inclut l'électrification des machines pour diminuer les émissions de gaz à effet de serre, la décarbonisation des matières premières, et la limitation de l'échange de mails non essentiels pour réduire l'empreinte numérique. De plus, l'entreprise privilégie l'utilisation de machines recyclables et encourage la réutilisation via le marché de l'occasion et de la pièce détachée.

1.5 Expérience personnelle sur l'environnement de travail

L'environnement de travail chez Manitou est particulièrement propice à la productivité grâce à plusieurs facteurs clés. Les équipements fournis, comme, par exemple les multiples écrans, me permettent de travailler dans d'excellentes conditions. L'utilisation de la suite Google Workspace est également très pratique, ces outils sont intuitifs, faciles à utiliser, et l'adaptation a été rapide.

La collaboration entre les équipes est facilitée par la configuration en open space qui favorise les échanges et la communication. J'ai eu l'opportunité d'assister à de nombreuses réunions ce qui m'a permis d'exprimer mon ressenti et de recevoir de l'aide lorsque cela était nécessaire, notamment au début de mon stage pour éviter d'être perdu.

Cependant, l'infrastructure de l'entreprise reste assez complexe. J'ai éprouvé des difficultés à tout assimiler au début et certains détails restent encore flous à ce jour. Malgré cela, j'apprécie l'organisation rigoureuse de l'entreprise, bien que cette rigueur impose un processus d'autorisation hiérarchique pouvant, parfois, ralentir le rythme de travail pour des actions mineures.

Cette phase de planification m'a permis d'appliquer les concepts étudiés en cours de Gestion de Projet et de Management des Systèmes d'Informations. Sans cette pratique préalable, il m'aurait été difficile de participer à l'élaboration du projet avec l'équipe.

En conclusion tout n'est évidemment pas parfait, tant dans l'organisation que dans l'infrastructure mais j'ai néanmoins senti la volonté des équipes à travailler ensemble et dans la même direction. Je pense que c'est ce qui importe le plus !

Chapitre 2

Phase de conception du projet

2.1 Précision du besoin

Comme expliqué dans la partie 1.2.2 - **Enoncé du besoin**, l'entreprise doit automatiser la collecte et le stockage des données des différents outils de gestion des licences logicielles afin d'optimiser les coûts de licences, générer des statistiques d'utilisation précises, réagir rapidement en cas de problème, et libérer du temps pour des tâches à plus forte valeur ajoutée.

2.2 Contraintes

Durant ce projet, il est essentiel de prendre en compte certaines contraintes lors de la planification afin de prioriser les demandes, d'évaluer leur faisabilité et de sélectionner les technologies appropriées. Une contrainte majeure concerne les outils déjà en usage au sein de l'entreprise, qui devront être privilégiés. De plus, certaines normes de développement pourront également influencer le choix des technologies.

2.3 Choix des technologies

2.3.1 Powershell :

Le choix qui a été fait pour extraire les données est Powershell. PowerShell est un langage de script et un environnement de ligne de commande développé par Microsoft. Initialement conçu pour les administrateurs système et les professionnels de l'informatique, il est utilisé pour automatiser la gestion des systèmes et le traitement des tâches administratives sur les systèmes Windows.

Pourquoi Powershell ?

PowerShell a été choisi pour ce projet en raison de ses nombreux avantages, notamment son intégration étroite avec les systèmes Windows et sa capacité à automatiser des tâches complexes grâce à ses scripts. Sa compatibilité avec différents langages et environnements en fait un outil flexible et puissant pour extraire et standardiser les données des divers outils de gestion des licences utilisés par l'entreprise. La qualité de la documentation associée à ce langage a également constitué un critère déterminant dans le choix de PowerShell pour ce projet.

Pourquoi cette technologie plus qu'une autre ?

La principale raison de ce choix réside dans la nécessité d'exécuter des commandes spécifiques pour interroger directement les serveurs des fournisseurs de licences afin de récupérer les données nécessaires.

2.3.2 Talend :

Talend est une plateforme d'intégration de données qui permet de collecter, transformer et charger des données provenant de différentes sources pour faciliter l'analyse et la prise de décision dans les entreprises. Les composants de Talend sont fabriqués en utilisant principalement le langage de programmation Java. Dans le cas de Manitou Group, c'était la possibilité de connecter très facilement différentes applications de leur système d'informations grâce aux nombreux composants Talend existants.

Pourquoi Talend ?

Talend est choisi pour ses multiples avantages. Il offre la possibilité d'intégrer une grande variété de sources de données. Son interface permet une prise en main rapide, tandis que sa capacité à maintenir une norme de développement cohérente simplifie la gestion des projets. De plus, Talend garantit la qualité des données grâce à ses outils de profilage et de nettoyage, tout en étant une solution robuste pour les processus d'ETL*.

Pourquoi cette technologie plus qu'une autre ?

Une des raisons est la montée en compétence bien plus rapide que sur des framework de développement classiques comme SpringBatch*. De plus cette technologie est déjà utilisée au sein de l'entreprise pour la grande majorité des projets ce qui me permet de recevoir de l'aide facilement, d'avoir beaucoup d'exemples à disposition et enfin, de faciliter la maintenabilité future.

2.3.3 Serveur FTP

Un serveur FTP (File Transfer Protocol) permet le transfert de fichiers entre ordinateurs via un réseau en utilisant le protocole FTP. Dans le cadre du projet, c'est la technologie qui va faire le lien entre l'extraction des données et l'intégration en base.

Pourquoi un serveur FTP ?

L'intérêt d'utiliser un serveur FTP est de permettre à n'importe quel ordinateur ayant les informations de connexions au serveur de récupérer les fichiers sur le FTP et ce, même s'il n'est pas dans le même réseau ou sur le même environnement. En l'occurrence, pour le projet en cours, cela permettra d'intégrer les données en bases depuis n'importe quel ordinateur et environnement utilisé.

Pourquoi cette technologie plus qu'une autre ?

La raison principale est de pouvoir accéder au serveur FTP via des script PowerShell, ce qui n'est pas forcément le cas pour d'autres outils permettant de stocker des fichiers. De plus, Talend possède déjà des éléments permettant d'aller chercher sur le FTP car il s'agit d'une technologie régulièrement utilisée par l'entreprise. Enfin la dernière raison est une volumétrie plutôt faible ne nécessitant que très peu d'espace de stockage.

2.3.4 PowerBi :

Power BI est une plateforme d'analyse de données et de visualisation développée par Microsoft, qui permet aux utilisateurs de transformer leurs données en informations exploitables.

Pourquoi PowerBi ?

Power BI est choisi pour sa capacité à intégrer facilement plusieurs types de données, offrant ainsi une solution complète pour représenter les outils de licences. Avec ses nombreuses options pour la création de tableaux, graphiques et filtres, Power BI nous permettra de visualiser efficacement les données et d'identifier rapidement les tendances importantes.

Pourquoi cette technologie plus qu'une autre ?

PowerBi est une technologie largement utilisée dans toute l'entreprise, elle garantit une familiarité parmi les membres de l'équipe, facilitant ainsi la collaboration sur les analyses. Cela permet également d'éviter aux employés d'avoir un surplus d'outils pour faire le même travail.

2.4 Brainstorming et collecte d'idées

2.4.1 Objectif du brainstorming

L'objectif du brainstorming, dans le cadre de ce projet, est de recueillir un maximum d'idées sur les fonctionnalités possibles. Cette méthode permet à chacun de partager librement ses idées, sans se préoccuper des contraintes techniques. Toutes les pensées, qu'elles soient bonnes ou mauvaises, doivent être extériorisées afin de couvrir pleinement tous les besoins potentiels du projet.

2.4.2 Méthodologie

Pour ce brainstorming, toute l'équipe R&D IS a participé afin d'obtenir un recueil global des idées concernant le projet, indépendamment du métier de chacun. Pour centraliser au mieux les idées, nous avons utilisé un tableur collaboratif Google Sheets*.

2.5 Réflexion personnelle :

Je n'ai pas beaucoup participé à l'élaboration de cette partie, car le projet avait été lancé il y a un an mais avait été mis en pause faute de moyens et de personnel. En finalité, j'ai seulement contribué au brainstorming pour collecter des idées. J'ai particulièrement apprécié la méthodologie utilisée pour le brainstorming, qui commençait par une collecte d'idées sans contraintes. En tant que développeur, il est parfois difficile de ne pas intégrer immédiatement les aspects techniques dans la réflexion, et cette approche m'a permis de penser plus librement.

De plus, pendant la présentation des technologies, j'étais partagé : D'un côté, j'appréhendais le fait de pouvoir maîtriser rapidement toutes ces nouvelles technologies, mais de l'autre, j'étais enthousiaste à l'idée d'apprendre de nouvelles compétences en étant entouré de personnes qualifiées.

Chapitre 3

Phase de planification du projet

3.1 Priorisation et définition du périmètre

3.1.1 Évaluation des idées

Le processus d'évaluation et de priorisation des idées s'est appuyé sur des critères prédéfinis, en mettant l'accent sur l'importance des besoins identifiés. La priorité est accordée aux fonctionnalités permettant de comparer l'utilisation actuelle des licences avec le nombre maximum disponible, et ce dans le but de déterminer si l'entreprise disposait d'un surplus ou d'une pénurie de licences.

Parallèlement, une attention particulière est portée à la maintenabilité et à la réutilisabilité du système, ce qui a conduit à accorder une priorité élevée à la documentation afin de garantir sa compréhension et sa pérennité. Les discussions au sein de l'équipe ont permis d'identifier et de hiérarchiser ces critères de priorité, assurant ainsi une allocation efficace des ressources et une orientation claire du développement du projet.

3.1.2 Définition du périmètre

Le périmètre du projet a été défini en tenant compte des priorités identifiées lors de l'évaluation des idées ainsi que des technologies choisies. Pour ce faire, les différentes idées ont été regroupées en lots, chacun représentant un niveau de priorité spécifique. Chaque lot est associé à un numéro distinct, permettant ainsi de visualiser clairement les différents niveaux de priorité du plus prioritaire au moins prioritaire.

3.2 Faisabilité technique et complexité

3.2.1 Analyse de complexité

Une fois les idées rédigées et priorisées, il est crucial de déterminer la faisabilité technique de chacune d'entre elles afin de pouvoir planifier au mieux le temps qu'allait me prendre chaque tâche pour éviter de me surcharger de travail. Pour ce faire, une consultation avec M.LHERIAUD a été initiée afin d'analyser les ressources nécessaires et les défis techniques à relever. Cependant, évaluer la complexité de chaque tâche s'est avéré délicat, notamment en raison de l'utilisation de technologies et de logiciels m'étant inconnus.

3.2.2 Rejet des idées impossibles

Lors du brainstorming, beaucoup d'idées ont émergé, mais certaines se sont révélées difficiles,

voire impossibles à mettre en oeuvre en raison de contraintes techniques. D'autres idées sortaient du cadre du projet de part le temps imposé par la durée du stage ou encore la gestion qui est déléguée à d'autres logiciels déjà existant.

3.3 Idées ressorties

En conclusion, voici un résumé du contenu de chaque lot :

Lot 1

Se concentre sur la gestion des données par technologie en limitant le nombre d'extractions à trois pour respecter les délais impartis, avec une documentation du code pour chaque type de technologie.

Lot 2

Mettre à disposition des vues instantanées et synthétiques de l'utilisation des licences, et planifier des présentations de maquettes pour l'équipe.

Lot 3

Afficher des informations clés telles que les dates d'expiration et d'effectivité, les refus de connexion ainsi que les utilisateurs en cours.

Lot 4

Se focalise sur l'analyse avancée de l'utilisation des licences, en intégrant des calculs de l'usage moyen et en tenant compte du calendrier et des jours actifs pour identifier les tendances et les pics d'utilisation.

Lot 5

Mise en place d'alertes en cas d'atteinte du nombre maximum de licences disponibles et gestion des paramètres d'alerte pour chaque licence.

Idées rejetées

Enfin, plusieurs idées ont été rejetées, notamment en raison de leur complexité, de limitations techniques ou de leur gestion par d'autres systèmes comme par exemple connaître la répartition géographique des utilisateurs avec leur adresse IP.

3.4 Conclusion sur la phase de planification

En conclusion, on se retrouve avec des lots à développer chacun ayant une priorité basée sur la nécessité et la complexité. Ces lots permettent de suivre l'avancement du projet tout en gardant une ligne directrice cohérente. Dans cette conclusion je tenais à rajouter une aide à la compréhension car la barrière entre "outil de licence", "technologie de licence" et "licence logiciel" est relativement floue. Voici un schéma permettant de mieux comprendre ces différents termes.

Pour conclure la partie concernant la planification de projet, je tenais à souligner que l'accompagnement de l'équipe m'a été extrêmement bénéfique. J'ai eu l'opportunité d'observer,

d'aider et de comprendre en profondeur le déroulement de la planification d'un projet informatique au sein d'une équipe. Initialement, il m'était difficile de cerner clairement les attentes du projet. Si on m'avait fourni directement les spécifications, j'aurais probablement rencontré des difficultés, ne connaissant pas encore l'environnement de l'entreprise, ses valeurs et les attentes spécifiques liées au projet. Participer activement à cette phase de planification m'a donc permis de bien saisir les attentes et de m'y conformer au mieux tout au long de mon stage.

De plus, toute la phase de faisabilité réalisée avec M.LHERIAUD m'a grandement aidé à comprendre le périmètre technique du projet, m'évitant ainsi une dispersion lors des phases de développement.

Si je devais donner mon avis sur la méthodologie durant la phase de brainstorming, je pense qu'il aurait peut-être été bénéfique de tenir une réunion directement avec les utilisateurs finaux du côté métier pendant la phase de brainstorming. Bien que M. GALLARD soit très informé sur le sujet et soit également un utilisateur final, avoir l'avis d'un utilisateur principal côté métier m'aurait sans doute permis d'obtenir une perspective supplémentaire. Cela aurait également pu aider à identifier des technologies de licences plus pertinentes à implémenter parmi les trois que je devais réaliser. Cependant le fait de ne pas voir directement avec l'utilisateur côté métier est volontaire. Cela est dû à un emploi du temps ne lui permettant pas de participer à la conception.

Chapitre 4

Extraction des données des outils de gestion

4.1 Spécifications

4.1.1 Objectif de l'extraction

L'objectif de cette extraction de données va être de récupérer toutes les données nécessaires pour répondre aux demandes effectuées dans la partie de planification. Ces données sont les suivantes :

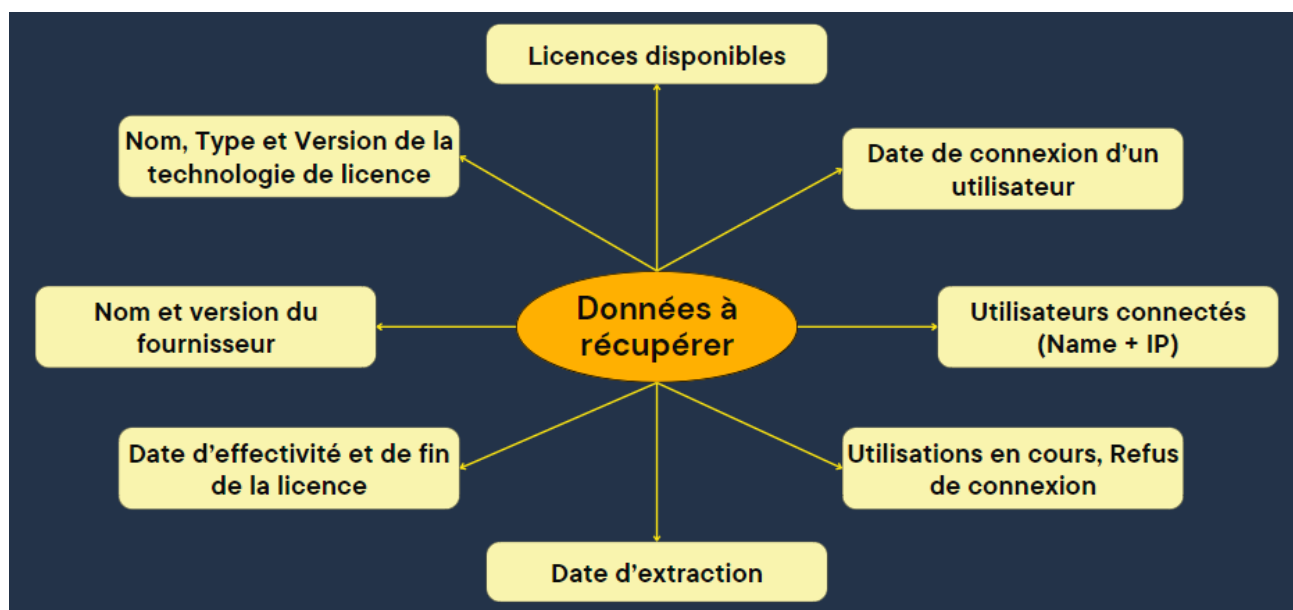


FIGURE 4.1 – Données à récupérer

4.1.2 Contexte

Pour cette extraction de données, il faut, en lançant simplement un script, aller récolter les données de tous les outils de gestion des licences. Il faut savoir que chaque outil ne fonctionne pas de la même manière. Certains vont renvoyer des données à l'exécution d'une commande, d'autres mettent à disposition un rapport web. Certains fournisseurs vont renvoyer les données concernant les utilisateurs séparément des données qui concernent les licences et d'autres ne renvoient pas toutes les données que nous souhaitons. L'enjeu ici va donc être de trouver un

moyen d'extraire toutes ces données sous un format uniforme tout en nécessitant le moins de paramétrage possible.

4.1.3 Contrat d'échange

Le contrat d'échange ici désigne la définition du moyen de communication entre deux applications. Dans notre projet, cela revient à dire : "comment doit-on renvoyer les données pour qu'elles puissent être utilisées par la technologie permettant d'intégrer ces données en base". Dans le contrat d'échange, on va retrouver également le format des données ainsi que les données renvoyées. Les données à renvoyer sont précisées dans la partie 4.1.1 - **Objectif de l'extraction**. En ce qui concerne le format, pour cette extraction de données, on souhaite renvoyer un fichier `.csv`. Ce format va être privilégié car il est compatible avec beaucoup de logiciels d'intégrations de données et notamment Talend.

```
id,nom,prenom,age  
1,millet,arthur,21  
2,dupont,martin,44
```

FIGURE 4.2 – exemple de Csv

4.1.4 Architecture de l'application

L'objectif est d'avoir une vue uniforme pour chaque outil. Pour faciliter ceci, l'architecture de l'application a été découpée en plusieurs parties :

- Un dossier `conf` qui va contenir pour chaque outil un fichier `nom_outil.env` qui contiendra les informations susceptibles de varier en cas de changement d'environnement.
- Un dossier par outil qui lui même contiendra un dossier `fonctions` et/ou un dossier `enums` et/ou un dossier `classes` pour les fonctions, classes et énumérations propres à cet outil.
- Un dossier `Commun` qui contiendra les classes, fonctions et énumérations communes à tous les outils de licences.
- Un fichier `imports.ps1` qui contiendra tous les imports utilisés dans plusieurs scripts.
- Un fichier `MALicenceTool.ps1` qui va être le script à lancer pour exécuter tous les scripts.

4.2 Processus d'extraction des données

Pour récupérer les données nécessaires, il est indispensable d'interroger le serveur du fournisseur afin qu'il nous renvoie les informations requises. La méthode d'extraction des données varie en fonction des outils utilisés. Pour certains outils, il est nécessaire d'exécuter une commande spécifique qui renvoie les valeurs souhaitées, tandis que pour d'autres, il suffit d'accéder directement au fichier source de licence contenant les informations recherchées.

Il existe une contrainte importante à considérer dans ce processus : les serveurs fonctionnent dans un environnement de production, ce qui empêche tout travail direct sur cet environnement. Pour surmonter cette contrainte, nous devons exécuter la commande et rediriger sa sortie vers un fichier `.txt`, permettant ainsi d'importer ces données dans l'environnement de développement pour effectuer des tests et des travaux ultérieurs.

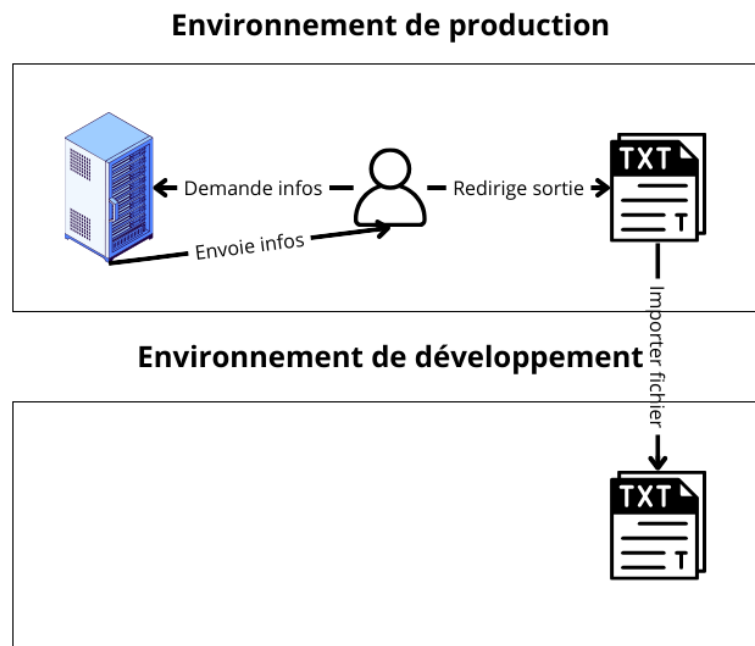


FIGURE 4.3 – Schéma d’une extraction de données

4.2.1 Transformation des données

Uniformisation des données :

Une fois les données récupérées, il est nécessaire de les structurer de manière uniforme pour chaque outil. Pour ce faire, nous allons organiser nos données en plusieurs classes distinctes, facilitant ainsi leur compréhension et leur uniformisation. Voici un diagramme de classe qui illustre la répartition des données :

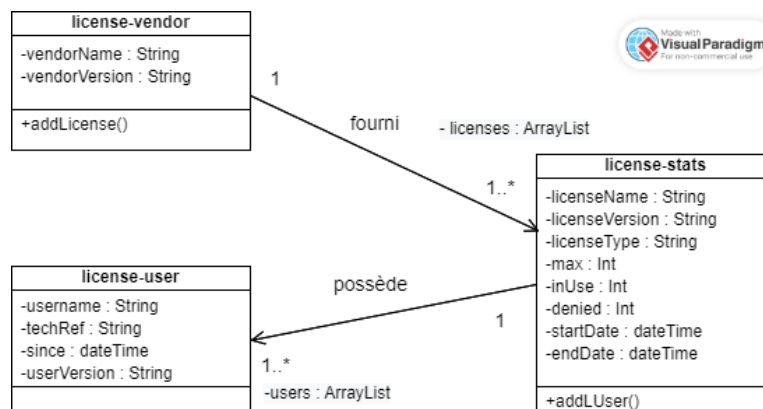


FIGURE 4.4 – Diagramme de classe de l’application

Nous pouvons également le résumer de cette manière : Un fournisseur possède une ou plusieurs licences, et une licence possède un ou plusieurs utilisateurs.

Récupération des informations :

Maintenant que nous avons défini comment uniformiser et hiérarchiser nos données il nous faut remplir nos classes avec les valeurs pour chaque outils.

Les données sont déformées par soucis de confidentialité mais le format des informations est le suivant :


```
# Fournisseur1
LICENSE un_nom_de_licence une_version une_date_d'expiration un_nombre_max_de_licence_disponibles share=i
min_timeout=900 max_roam=32 start=une_date_de_debut issuer="Fournisseur 1" customer="Licensed to: MANITOU BF"
issued=une_date replace_ck=1234567 sig="un_uuid"
# Fournisseur2
LICENSE un_nom_de_licence une_version une_date_d'expiration un_nombre_max_de_licence_disponibles share=i
min_timeout=900 max_roam=32 start=une_date_de_debut issuer="Fournisseur 1" customer="Licensed to: MANITOU BF"
issued=une_date replace_ck=1234567 sig="un_uuid"
# Fournisseur2
LICENSE un_nom_de_licence une_version une_date_d'expiration un_nombre_max_de_licence_disponibles share=i
min_timeout=900 max_roam=32 start=une_date_de_debut issuer="Fournisseur 1" customer="Licensed to: MANITOU BF"
issued=une_date replace_ck=1234567 sig="un_uuid"
# Fournisseur2
LICENSE un_nom_de_licence une_version une_date_d'expiration un_nombre_max_de_licence_disponibles share=i
min_timeout=900 max_roam=32 start=une_date_de_debut issuer="Fournisseur 1" customer="Licensed to: MANITOU BF"
issued=une_date replace_ck=1234567 sig="un_uuid"
```

FIGURE 4.5 – Exemple de fichier le licence

Utilisation des REGEX pour la manipulation de chaînes de caractères

Une **REGEX** (Expression régulière) est une séquence de caractères qui forme un modèle de recherche. Ce modèle est utilisé pour correspondre, rechercher et manipuler des chaînes de caractères basées sur un motif précis. L'expression régulière est un outil extrêmement puissant car il va permettre de définir des règles complexes.

Exemple de REGEX en PowerShell

Pour illustrer l'utilisation des REGEX, considérons l'exemple suivant

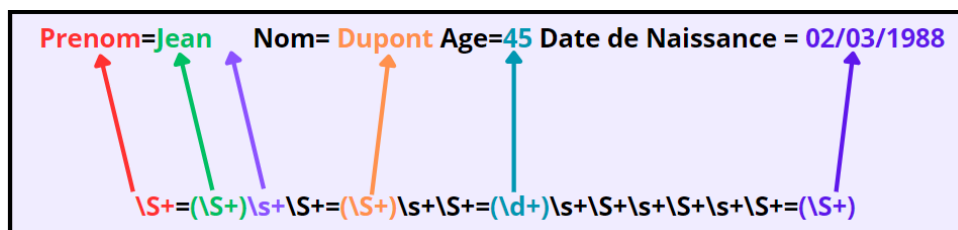


FIGURE 4.6 – Exemple d'utilisation d'une REGEX

Explication de la REGEX

- \d+ : Correspond à un ou plusieurs chiffres.
- \s+ : Correspond à un ou plusieurs espaces.
- \S+ : Correspond à un ou plusieurs caractères non-espaces.
- () : Correspond à ce qu'on veut capturer

La REGEX va donc découper les lignes qui correspondent au motif défini et va capturer sur ces lignes tous les motifs qui se trouvent entre les parenthèses de la REGEX. En l'occurrence, sur la **Figure 4.6**, les premières parenthèses devront capturer "Jean", les deuxièmes "Dupont", puis "45" et enfin "02/03/1988".

Pour chaque fichier renvoyé par les outils de gestion des licences je vais effectuer une opération similaire.

4.2.2 Mise à disposition des données

Comme défini dans le contexte d'échange, maintenant que toutes les données sont récupérées, il faut les mettre à disposition pour qu'elles puissent être insérées en base. Pour cela nous al-

lons générer 2 fichiers `.csv` par exécution des scripts. Le premier prendra toutes les données concernant le vendor et les licences, et le deuxième celles concernant les utilisateurs.

Pour cela, il faut utiliser la fonction déjà présente dans PowerShell, appelée `Add-Content`. Cette fonction permet d'ajouter du contenu à un fichier spécifié. Si le fichier n'existe pas, elle le crée automatiquement.

Exemple d'export PowerShell en csv

```
Add-Content -Path "D:\arthur\unFichier.txt" -Value "Toto mange du pain"
```

4.3 Problèmes et solutions

4.3.1 Problèmes rencontrés et solutions apportées

Données réparties en plusieurs fichiers

Lors de l'extraction des données, le problème rencontré pour certains outils de gestion était que l'ensemble des données n'étaient pas répertoriées dans un seul fichier. Les données concernant les vendor et les technologies de licences étaient dans un fichier tandis que les données concernant les utilisateurs étaient dans un autre fichier.

Pour pallier ce problème j'ai donc du faire en sorte de récupérer les données dans ces différents fichiers via plusieurs REGEX. J'ai ensuite dû comparer les données pour relier les utilisateurs d'une licence aux données de la licence correspondante. L'inconvénient de cette méthode est la multiplication du nombre de boucles et de comparaisons dans le code. Il y aura donc une légère perte de performance.

Données chiffrées

Le deuxième problème rencontré : les données dans le fichier de licence étaient chiffrées ce qui rendait impossible la récupération sans la fonction de déchiffrement.

Pour résoudre ce problème, il a fallu utiliser le fichier `.jar`* fourni par le serveur. Pour exploiter les fonctions du `.jar`, un projet Java a été créé afin d'utiliser la fonction de déchiffrement contenue dans le `.jar`. Une fois le fichier de licence déchiffré, un fichier `.txt` avec les informations pertinentes a été généré pour être transformé dans un script PowerShell de la même manière que pour les autres outils de gestion. Ainsi, les scripts PowerShell devaient également exécuter le projet Java lors de la récupération des données pour accéder au fichier généré.

Ajout de l'identifiant

Le problème avec l'identifiant est que les données sont réparties dans plusieurs csv. La question s'est donc posée de savoir comment obtenir un identifiant unique par ligne alors que les données sont réparties dans plusieurs fichiers.

Pour résoudre cela, j'ai généré un UUID* aléatoire. Cela solutionnerait notre problème de fichiers différents, mais est-ce possible de générer 2 UUID identiques ? Auquel cas cette méthode ne fonctionnerait pas. Pour générer 2 UUID identiques il faudrait générer 1 million d'UUID par secondes pendant 85 années, ce qui est extrêmement peu probable pour la quantité de données générées.

Données inexistantes

Enfin le dernier problème rencontré concerne certains fournisseurs qui ne renvoient pas toutes les données dont nous avons besoin ce qui complique, parfois, l'uniformisation.

Ce problème n'a malheureusement pas pu être corrigé sauf dans certains cas spécifiques où les informations se trouvaient dans des fichiers annexes. La question à se poser est de savoir ce qu'il faut faire pour les données inexistantes. Faut-il mettre une valeur par défaut ou alors, laisser vide les données inexistantes. Sur ce point M.LHERIAUD a préféré laissé vide les données inexistantes car une valeur par défaut pourrait fausser les statistiques par la suite. De plus une absence d'informations est une information.

4.4 Tests

Pendant cette phase, aucun framework de test n'a été utilisé. En effet, comme expliqué dans la section 1.4.1 - **Infrastructure informatique** et la section 4.2 - **Processus d'extraction des données**, l'entreprise possède trois environnements distincts. Le problème pour les tests est qu'il n'est pas possible d'extraire des données sur l'environnement de développement, car les serveurs des outils de licences fonctionnent dans l'environnement de production. De plus, pour ce projet, il n'y a pas d'environnement de qualif permettant de tester dans des conditions proches de la production.

Par conséquent, les seuls tests réalisables sont des tests avec un fichier d'entrée contenant des données spécifiques, pour vérifier si la sortie était celle attendue, ou alors quelques tests structurels afin de s'assurer que le résultat des fonctions était cohérent.

4.4.1 Exemple de test structurel d'une fonction

La figure ci-dessous illustre la démarche à suivre pour déterminer les tests structurels de la fonction `getTodayMetrics`. Cette fonction vérifie la présence d'un fichier de Metrics dans un dossier donné.

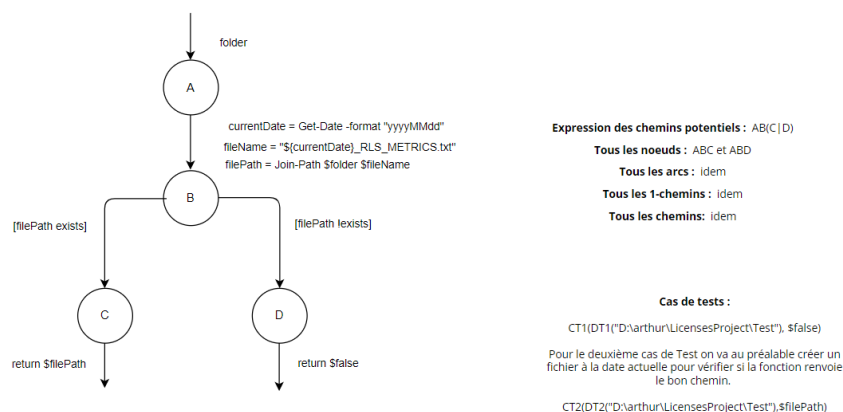


FIGURE 4.7 – Test structurel `getTodayMetrics`

4.5 Conclusion et réflexion personnelle

Conclusion

En conclusion, on se retrouve avec deux `.csv` par outil de gestion, un pour les utilisateurs et un pour les licences et fournisseurs. Chaque `.csv` est uniforme et contient les données qui vont nous permettre de répondre aux besoins définis dans la phase de planification.

Réflexion personnelle

Le projet a été lancé il y a un an et M. LHERIAUD a déjà établi un début d'architecture d'application. Il a également initié l'extraction des données pour un des outils, ce qui m'a servi de référence pour commencer. Cette préexistence m'a offert une base solide et des repères précieux quant à l'utilisation de PowerShell et à l'uniformisation des données. Cependant, en débutant l'extraction des autres outils, il est devenu évident que la structure existante nécessitait des ajustements pour répondre à l'ensemble des besoins. J'ai donc effectué des modifications importantes sur le travail déjà réalisé par M. LHERIAUD.

L'utilisation de PowerShell s'est avérée à la fois intuitive et pratique. La richesse des fonctionnalités déjà implémentées a grandement facilité le développement et amélioré la lisibilité du code. Malgré cela, le processus d'apprentissage a été un peu difficile au départ, étant donné que je n'étais pas familiarisé avec l'environnement de l'entreprise et que mes connaissances en PowerShell étaient limitées. Je m'attendais initialement à ce que les extractions soient assez répétitives, mais j'ai rapidement réalisé que chaque outil de gestion des licences renvoyait des données avec des structures très différentes, nécessitant ainsi une adaptation constante de mon approche pour atteindre mes objectifs.

Cette partie du projet est celle qui m'a le plus exposé à des concepts techniques. J'ai pu en découvrir de nouveaux comme les REGEX mais j'ai également eu l'opportunité d'appliquer des connaissances existantes, comme le langage objet appris en cours de Kotlin et les concepts client-serveur étudiés en cours de Programmation Système.

*Pour plus d'informations sur cette partie, vous pouvez consulter l' **Annexe B - Informations supplémentaires sur l'extraction des données***

Chapitre 5

Intégration des données

5.1 Objectifs et contexte

5.1.1 Objectif de l'intégration

L'objectif de l'intégration est de récupérer les données depuis une ou plusieurs sources et de les pousser dans une base de données.

5.1.2 Contexte

Dans le contexte du projet, il faut récupérer les données extraites pour les envoyer en base de données. En revanche un problème se pose c'est qu'il n'est pas possible de récupérer les données extraites car elles sont situées dans un dossier du projet. Pour résoudre cela, il faut envoyer ces données sur un serveur FTP. La deuxième question qui se pose c'est de choisir le dépôt des données qui n'a pas été défini en phase de conception du projet. Une fois ce dépôt déterminé il faut réaliser l'interface Talend permettant de récupérer les fichiers sur le FTP puis de les pousser sur dépôt.

5.2 Transfert des données vers le serveur FTP

5.2.1 Objectif du transfert

L'objectif est de réaliser le transfert des fichiers sur le serveur FTP de l'entreprise pour ensuite pouvoir, grâce à Talend, récupérer les fichiers déposés sur le serveur et cela peu importe l'environnement de travail.

5.2.2 Processus de transfert

Configuration de l'architecture du projet

Dans un premier temps il faut créer l'architecture du projet. M.LHERIAUD a donc créé un dossier avec le nom du projet sur le FTP n'ayant pas, moi même, l'autorisation de le faire. Une fois fait, j'ajoute à l'architecture initiale du projet PowerShell un fichier `ftp.ps1` qui contient toutes mes fonctions relatives au serveur FTP.

Transfert des fichiers

Voici un schéma qui illustre le déroulement du transfert des fichiers vers le serveur FTP une fois l'extraction des données terminée :

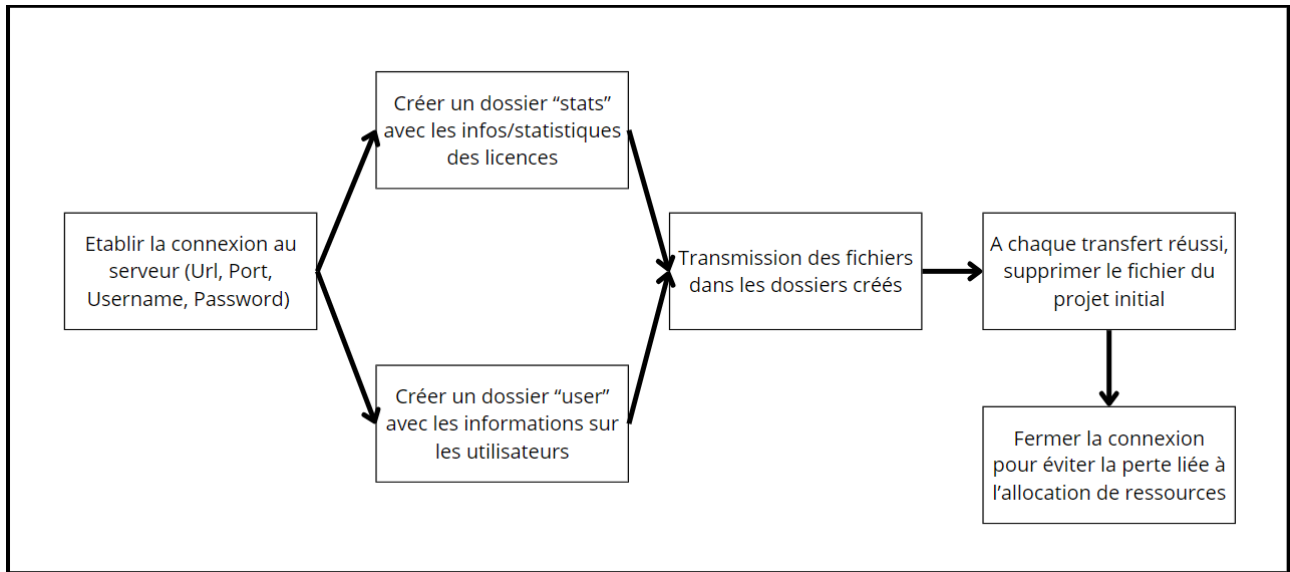


FIGURE 5.1 – Processus de transfert vers le serveur FTP

Les Web Requests

Pour transférer des fichiers vers le serveur FTP, j’ai utilisé les Web Requests de PowerShell. Ces requêtes permettent diverses opérations réseau telles que le transfert de fichiers, l’envoi de données à des serveurs web ou la récupération de données depuis ces derniers. Grâce à cette fonctionnalité, il devient possible d’effectuer des actions telles qu’envoyer des fichiers, supprimer des fichiers ou ajouter des répertoires directement sur le serveur FTP.

Voici un schéma expliquant le fonctionnement d’une web request vers le serveur FTP en PowerShell :

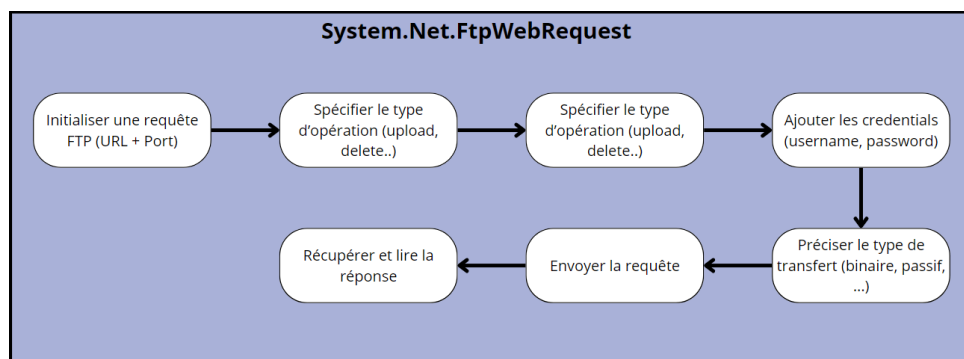


FIGURE 5.2 – Fonctionnement d’une Web Request en PowerShell

Pour plus d’informations sur le serveur ftp, vous pouvez consulter l’Annexe C

5.3 Processus d’intégration

5.3.1 Choix du dépôt des données

Au début de la phase de conception, le point de dépôt des données n’avait pas encore été déterminé. Nous savions que l’ETL serait Talend, mais l’emplacement restait indécis. À

mi-parcours du projet, une réunion avec l'équipe Data a été organisée pour répondre à cette question. Il a été convenu que les données seraient hébergées sur BigQuery un entrepôt de données (datawarehouse) proposé par Google Cloud Platform.

5.3.2 Google Cloud Platform

Présentation de GCP

Google Cloud Platform (GCP) est une suite de services cloud publics proposés par Google, conçue pour répondre à divers besoins techniques des entreprises. GCP va permettre aux développeurs de déployer et de gérer des applications sur une infrastructure évolutive.

Il faut savoir que actuellement au sein de l'entreprise, GCP est encore en phase de test, ce n'est pas la technologie utilisée par les applications actuellement en fonctionnement.

La technologie actuel est un Blob Storage*. Un inconvénient notable du Blob Storage réside dans son manque de structure et de capacité d'analyse intégrée. Étant conçu principalement pour le stockage de données brutes non structurées, il est difficile d'effectuer des analyses avancées ou des requêtes complexes.

L'intégration de BigQuery et de GCP chez Manitou favoriserait non seulement la structuration des données, mais également une interopérabilité accrue entre les applications. Cela se traduirait par une architecture de données cohérente et indépendante des technologies sous-jacentes.

Présentation de BigQuery

BigQuery est un service de stockage et d'analyse de données massives proposé par GCP. Il offre une capacité de traitement rapide des requêtes SQL sur de grands ensembles de données, permettant des analyses détaillées et en temps réel.

Avantages de BigQuery ?

BigQuery offre de nombreux avantages pour Manitou. En utilisant du SQL, il permet aux employés, déjà familier avec le langage, de l'adopter rapidement sans nécessiter de formation supplémentaire.

BigQuery permet également de maintenir des performances élevées garantissant des analyses rapides, même à grande échelle.

De plus, son intégration avec Google Workspace facilite le transfert et l'analyse des données entre les services que l'entreprise utilise déjà, comme Google Sheets et Google Analytics.

Avec des fonctionnalités de sécurité avancées et un modèle de facturation basé sur l'utilisation, BigQuery assure une gestion efficace des coûts et la protection des données, tout en offrant des capacités d'analyse enrichies grâce à ses fonctionnalités avancées. Cela permet également de s'aligner sur la stratégie de l'entreprise et les objectifs de New Horizon 2025.

5.3.3 Les Jobs Talend

Dans Talend, un job est un processus ETL qui automatise l'importation des données. La création des jobs comprend la définition des sources de données, l'application des transformations nécessaires, et la spécification des destinations de données.

Un job Talend est une suite de composants reliés par des connecteurs, chacun réalisant une tâche spécifique dans le processus ETL, comme la lecture, la transformation, ou l'écriture

de données. Les connecteurs dirigent le flux de données et orchestrent l'ordre d'exécution des tâches.

Un job est toujours composé de 3 parties :

- **PréJob** : Le PréJob est un sous-job qui s'exécute avant le job principal. Il est utilisé pour effectuer des tâches préliminaires nécessaires au bon déroulement du job principal. Dans le cadre du projet je m'en sers uniquement pour ouvrir les connexions.
- **PostJob** : Le PostJob est un sous-job qui s'exécute après l'exécution du job principal. Il est utilisé pour effectuer des actions finales après que le traitement principal a été effectué. Dans le cadre du projet le post-job sert uniquement à fermer les connexions.
- **Job principal** : Le job principal dans Talend exécute les principales opérations ETL.

5.3.4 Présentation des jobs réalisés

Un des avantages de Talend est sa simplicité de compréhension du fonctionnement des jobs. Cependant, il est important de respecter certaines normes de développement pour maintenir cette simplicité.

L'une de ces normes consiste à diviser un projet en au moins deux jobs distincts. Le premier job exécute les actions souhaitées, tandis que le second orchestre l'exécution de ce premier job. Chaque composant devra aussi être nommé en fonction de l'action qu'il fait.

Voici un exemple de job Talend :

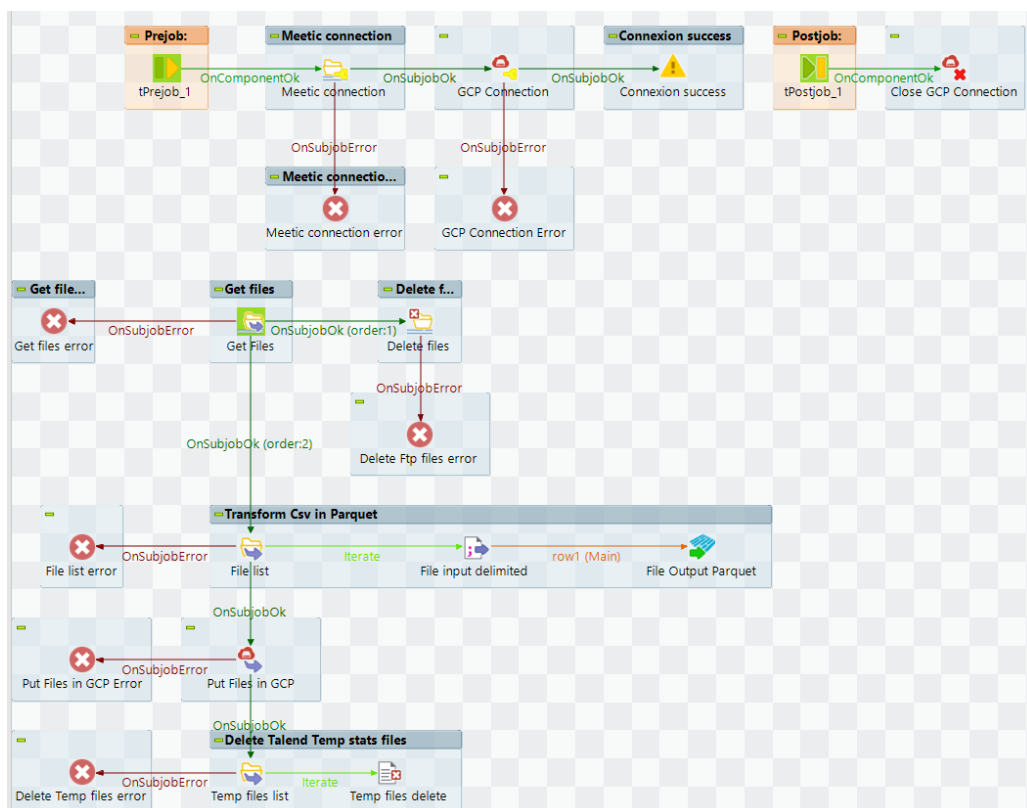


FIGURE 5.3 – Jd_licenses_datas_to_datawarehouse_010_push_ftp_stats

Dans le cadre du projet j'ai 3 jobs :

- **Js_Licenses_datas_to_datawarehouse_01** : Ce job sera le principal, c'est celui qui lancera l'exécution des deux autres.
- **Jd_licenses_datas_to_datawarehouse_010_push_ftp_stats** : Dans ce job, on établit la connexion au FTP et à GCP. Les fichiers sont récupérés du FTP, leur contenu est lu, transformé en fichiers parquet, puis supprimé du FTP. Les fichiers parquet sont stockés temporairement dans un dossier Talend avant d'être envoyés vers GCP et supprimés du dossier temporaire. Enfin, les connexions sont fermées. La connexion au serveur FTP se ferme automatiquement après quelques minutes. Pour chaque composant, un composant d'erreur est ajouté pour indiquer où le job est bloqué et pourquoi.
- **Jd_licenses_datas_to_datawarehouse_020_push_transco** : Ce job a pour objectif de transformer la table de transco en un fichier **parquet**. Pour cela, il se connecte d'abord à la base de données, lit le contenu, puis le transforme en fichier parquet. Ensuite, il envoie ce fichier sur GCP. Pour ce faire, il ouvre une connexion vers GCP, lit le dossier temporaire où le fichier parquet de la table de transco est stocké, envoie le fichier sur GCP, puis supprime le fichier parquet du dossier temporaire.

5.4 Problématiques et solutions

5.4.1 Problèmes rencontrés et solutions apportées

Modification du périmètre

Lors d'une réunion avec l'équipe Data, il a été déterminé que l'intégration toutes les 15 minutes pour récupérer les utilisateurs connectés sur les licences, était considérée comme du temps réel. Cela nécessitait potentiellement l'utilisation de technologies et d'autorisations différentes. Dans le cadre du stage, cela ne pouvait pas être réalisé en raison des contraintes de temps. Par conséquent, nous avons retiré les utilisateurs du cadre du projet et opté pour une intégration 1 à 2 fois par jour afin de voir combien de données seraient renvoyées et d'adapter progressivement l'intervalle d'intégration.

Cependant, même si l'intégration des données ne se fait pas toutes les 15 minutes, il est toujours possible de récupérer les données toutes les 15 minutes afin d'avoir des données cohérentes lors de l'intégration 1 à 2 fois par jour.

5.5 Conclusion et réflexion personnelle

Conclusion

En conclusion, nous avons trois jobs distincts pour faciliter la compréhension du fonctionnement de l'intégration. Le dépôt de données initialement défini a été modifié après des réunions et réflexions avec l'équipe Data, passant d'une base de données classique à Google Cloud Platform. La fréquence d'intégration a également évolué depuis la phase de planification, passant de toutes les 15 minutes à une ou deux fois par jour. Cette décision implique qu'il ne sera plus possible de savoir qui est connecté sur quelle licence à un instant t .

Réflexion Personnelle

La planification de cette phase était assez floue pour moi car il s'agissait d'une partie du projet sur laquelle l'équipe s'était très peu attardée dans la phase de planification/conception.

Je connaissais déjà le principe d'intégration des données mais je ne connaissais pas Talend ni son fonctionnement.

Une fois la planification terminée, j'ai commencé à mieux cerner l'attente et le travail qui m'incombait. M.LHERIAUD m'a présenté Talend, j'ai trouvé l'ETL très complet et assez simple d'utilisation au premier abord. La difficulté résidait dans le fait qu'il fallait paramétrer correctement les paramètres d'environnement pour que les jobs fonctionnent. De plus Manitou souhaite rendre paramétrable le plus de facteurs possibles pour facilement modifier les jobs dans le futur. C'est certes un avantage mais cela demande un peu plus de temps au départ pour consigner les paramètres dans des fichiers et faire en sorte qu'ils soient utilisable par Talend.

Manitou a également mis en place beaucoup de normes de développement comme par exemple, les noms de jobs qui doivent être d'une certaine forme, les commit doivent également respecter une certaine forme etc. Une fois habitué aux normes, tout devient parfaitement lisible et compréhensible mais ça reste difficile de penser à tout dès le début.

En finalité, le processus reste relativement commun pour une intégration. Ce qui a rendu la tâche intéressante mais difficile en même temps c'est de penser que le projet est commun à d'autres personnes et qu'il faut donc rendre la chose le plus lisible et maintenable possible.

*Pour plus d'informations sur cette partie, vous pouvez consulter l' **Annexe D - Informations supplémentaires sur l'intégration des données***

Chapitre 6

Consolidation des données

6.1 Objectifs et contexte

6.1.1 Objectif de la consolidation

L'objectif de la consolidation des données est d'unifier et de structurer les données extraites afin de faciliter les analyses futures. Cela permet d'assurer une meilleure lisibilité et cohérence des données, ce qui rend ainsi les analyses plus efficaces et fiables. La consolidation vise également à préparer une architecture adaptable et évolutive pour répondre aux besoins futurs du projet.

6.1.2 Contexte

Lors de la conception et de la planification du projet, la consolidation des données n'avait pas été initialement envisagée, en raison de la difficulté à anticiper la nature des données renvoyées. Il était donc compliqué de se projeter sur une éventuelle consolidation. Une fois la phase d'extraction terminée, la quantité de données extraite ne nécessitait pas de consolidation. Cependant, il a finalement été décidé de procéder à une consolidation des données pour les rendre plus lisibles. De plus, le projet étant destiné à évoluer, il était préférable d'adopter une architecture facilement modifiable et efficace. Pour ce faire, l'utilisation du framework DBT a été retenue.

6.2 Framework Dbt

6.2.1 Présentation de Dbt

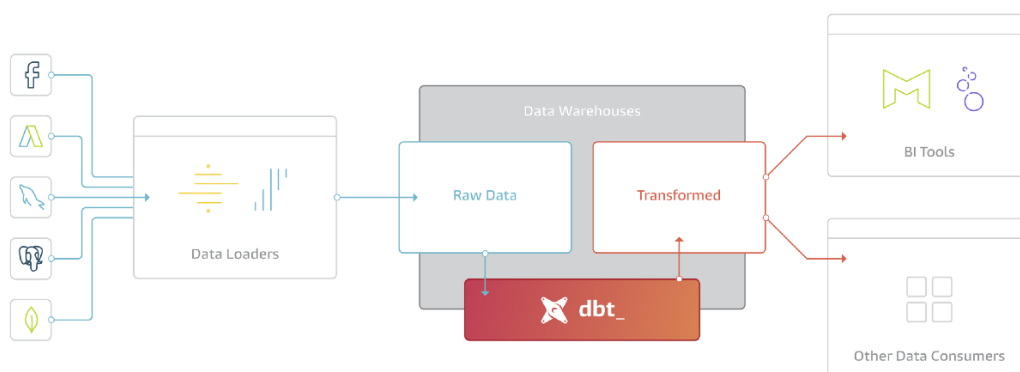


FIGURE 6.1 – Fonctionnement de dbt

DBT (Data Build Tool) est un outil open source dédié à la transformation des données de type ELT*. DBT permet d'effectuer des transformations directement dans l'entrepôt de données, que ce soit sous forme de tables ou de vues.

Il s'intègre généralement dans une stratégie ELT. D'abord, les données sont chargées dans un data warehouse, puis elles sont transformées par dbt. Ensuite, ces données transformées sont mises à la disposition des équipes décisionnelles.

6.2.2 Justification du choix de Dbt

Le choix de DBT offre plusieurs avantages significatifs. Tout d'abord, DBT utilise le langage SQL, largement documenté et couramment utilisé en base de données, facilitant ainsi l'apprentissage. De plus, en étant open source, il offre une transparence au niveau du code source et réduit les coûts associés à la transformation des données par rapport aux solutions payantes.

DBT simplifie la gestion des dépendances entre les transformations et automatise la documentation des changements, ce qui facilite le suivi de l'évolution des données.

DBT permet également de tester la compatibilité avec les solutions technologiques existantes, telles que Google Cloud Platform (GCP) et les data lakes.

Enfin, travailler avec DBT en tant que stagiaire offre l'opportunité de découvrir une technologie prisée dans le domaine de la gestion des données, contribuant ainsi à mon développement professionnel.

6.3 Processus de consolidation

6.3.1 Structuration des données

Afin de rendre l'architecture lisible et cohérente, les données sont découpées en plusieurs tables distinctes. Voici un modèle conceptuel de données qui illustre le découpage :

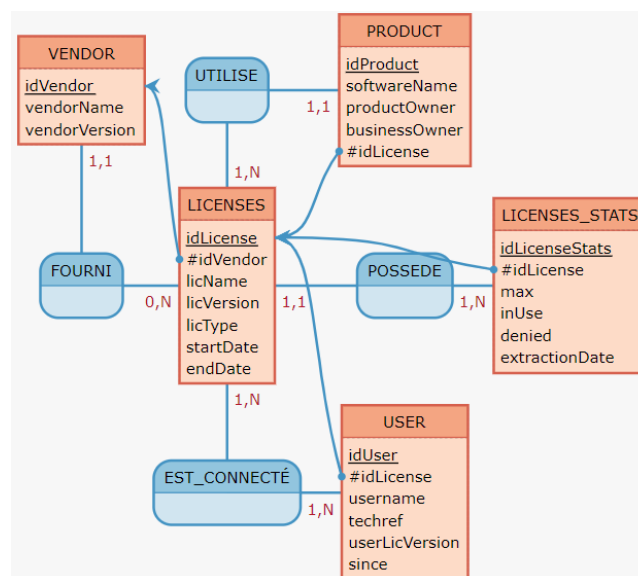


FIGURE 6.2 – Modèle conceptuel de données

6.3.2 Utilisation de dbt

6.3.3 Architecture du framework

- **Modèles (Models)** : Les modèles sont des fichiers SQL contenant la logique de transformation des données.
- **Configurations** : Les fichiers de configurations sont définies via des fichiers **YAML** (comme `dbt_project.yml`), sur lesquels il est possible de spécifier des paramètres d'environnements.
- **Macros** : Les macros sont des blocs de code réutilisables écrits en Jinja*, permettant d'écrire du SQL dynamique et de réutiliser des morceaux de code à travers plusieurs modèles.
- **Tests** : Les tests dans dbt valident les données en vérifiant des contraintes comme la non-nullité, l'unicité, etc.
- **Documentation** : dbt génère une documentation interactive et visuelle du projet.
- **Snapshots** : Les snapshots capturent l'état des données à un moment donné. Utile pour le suivi des changements et les analyses historiques.
- **Packages** : dbt permet d'installer des packages externes.

6.3.4 Configuration du projet

Avant d'utiliser dbt, il est nécessaire de le configurer. Cela inclut la création d'un environnement virtuel Python, l'installation des packages dbt requis, et la configuration du fichier `profiles.yml` avec les détails de connexion à l'entrepôt de données, tels que l'URL du serveur, le port, les identifiants de connexion, le nom de la base de données et le schéma.

L'installation de dbt peut se faire de trois manières : manuellement, via une image Docker, ou en utilisant Python avec `pip install`. Parmi ces options, l'installation avec Python est la plus simple, la plus sûre et la plus rapide car elle se fait en seulement quelques commandes :

```
python -m venv dbt-env
dbt-env\Scripts\activate
python -m pip install dbt-core dbt-bigquery
dbt init
```

6.3.5 Fonctionnement global :

Il est important de noter que Dbt ne permet pas de créer des bases de données comme on peut le faire avec SQL et `CREATE TABLE`. Dbt se contente de se connecter à une base de données existante pour y effectuer des transformations. Dans ce cas, je vais configurer Dbt pour qu'il se connecte à une base de données créée sur BigQuery*, un data warehouse proposé par GCP. Cette base sera créée à partir des fichiers transférés sur GCP par Talend. C'est sur cette base que je réaliserai les transformations.

La création de la base ne fait pas partie du projet car elle est gérée par la partie Data de l'IT&D. En ce qui concerne le fonctionnement, la fonctionnalité utilisée est celle des **tables externes**. Cette fonctionnalité permet de pointer vers un emplacement spécifique, tel qu'un fichier ou un dossier, dans le stockage de données de GCP. Ainsi, dès qu'un fichier **parquet** est ajouté à l'emplacement spécifié, BigQuery l'intègre automatiquement dans sa base de données.

6.3.6 Fonctionnalités utilisées

Modèles

Dans le framework, j'ai utilisé les modèles pour créer des tables de manière automatique via des requêtes SQL SELECT. Cela m'a permis de construire l'architecture décrite dans la section 6.3.1 - Structuration des données.

Contraintes et tests

Je me suis servi du fichier `schema.yml` pour déclarer des contraintes et des tests comme les clés primaires, secondaires, l'unicité d'une colonne ou encore sa non-nullité. Voici un exemple de fichier `schema.yml` :

```
models:
  - name: vendors
    columns:
      - name: idVendor
        tests:
          - not_null
          - unique
        constraints:
          - type: not_null
          - type: unique
          - type: primary_key
        description: Identifiant unique de la table vendors
```

FIGURE 6.3 – Exemple de fichier `schema.yml`

Le lancement des tests se fait avec la commande : `dbt test`

Documentation

Dbt donne la possibilité de générer une documentation via ces commandes : `dbt docs generate` et `dbt docs serve`. Cela va générer héberger un site web en local permettant de consulter nos modèles, les descriptions ajoutées, les tests, les tables, les contraintes, etc. Pour encore plus de lisibilité j'ai également commenté le code.

6.3.7 Fonctionnalités utilisables dans le futur

Dans le futur, il sera possible d'utiliser des sources pour référencer les fichiers d'origine des données et ainsi faciliter la traçabilité. Les snapshots permettraient de suivre facilement les données sur le dernier mois ou les dernières années. Enfin, on pourrait envisager d'utiliser des seeds à l'avenir. Les seeds, en dbt, sont des fichiers CSV qui peuvent être chargés directement dans l'entrepôt de données sous forme de tables. Cette fonctionnalité permettra de fournir des données statiques ou de référence, qui ne changeront presque pas, directement dans l'entrepôt de données, facilitant ainsi leur accès et leur gestion.

6.4 Conclusion et réflexion personnelle

Conclusion générale

En conclusion, la consolidation des données à l'aide de dbt a prouvé son efficacité en optimisant l'architecture des données et en rendant les analyses plus fiables et accessibles. Cette démarche permet d'anticiper les besoins futurs du projet et d'adopter une solution flexible et performante. Le choix de dbt, s'est donc révélé bénéfique, dans le développement et l'évolution du projet. De plus, cela s'accorde parfaitement avec les objectifs de New Horizon 2025 et avec l'environnement de l'entreprise.

Réflexion personnelle

La découverte de dbt a été particulièrement intéressante et enrichissante. Apprendre à utiliser un outil moderne et couramment utilisé m'a permis de mieux comprendre les défis liés à la transformation des données.

Cependant, j'ai trouvé dommage de ne pas pouvoir exploiter pleinement tous les avantages de ce framework, mais néanmoins, cette expérience m'a donné un aperçu de son potentiel et m'a montré comment utiliser cette technologie dans mes futurs projets, tant professionnels que personnels.

Par ailleurs, lors de la mise en place du projet dbt sur Git, cela nécessitait des autorisations spécifiques ralentissant légèrement le rythme de progression. Dans les grandes entreprises comme Manitou, obtenir ces autorisations peut prendre du temps, illustrant un des inconvénients des structures très hiérarchisées.

Enfin, cette partie m'a permis d'appliquer des concepts vus en cours sur les modèles relationnels, la structuration des données ou encore le SQL.

*Pour plus d'informations sur cette partie, vous pouvez consulter l' **Annexe E - Informations supplémentaires sur la consolidation des données***

Chapitre 7

Création du rapport

7.1 Objectifs et contexte

7.1.1 Objectif du rapport Power BI

L'objectif est de créer un rapport permettant aux utilisateurs finaux du projet, d'avoir une visualisation des statistiques qui concernent les licences logiciels, leurs fournisseurs et les utilisateurs de ces licences.

7.1.2 Contexte

Dans le cadre du projet, des maquettes devront être fournies pour valider avec les utilisateurs métiers si les données correspondent à leurs attentes pour identifier les éventuels manques ou pour repérer les données inutiles. Bien que les utilisateurs ne soient plus dans le périmètre actuel du projet, les maquettes doivent tout de même les intégrer. Ainsi, une fois l'autorisation obtenue du côté de la Data, il suffira de recopier la maquette.

7.2 Processus de création du rapport

7.2.1 Conception des visualisations

Afin de répondre au besoin initial, le rapport à été réparti en 6 pages :

- **Sommaire des licences et fournisseurs** : La première page du rapport recensera les informations des fournisseurs et de leurs licences dans un tableau de synthèse.
- **Licences Maximum** : Cette page va répertorier dans un tableau toutes les technologies de licences où toutes les licences sont utilisées.
- **Statistiques** : Ici on retrouvera sous la forme d'un diagramme en baton, les statistiques de chaque technologie de licences à savoir le nombre maximum disponible, la moyenne de l'utilisation depuis la première extraction, le pic d'utilisation depuis la première extraction, le nombre de refus.
- **Evolution sur une période** : On retrouve les même informations que la page précédente mais avec un graphique en courbe permettant d'avoir une vue par heure/jour/mois/année.
- **Utilisateurs connectés** : Sur cette page on retrouve toutes les informations concernant les utilisateurs connectés sur une licence.
- **Tous les utilisateurs** : Enfin sur la dernière page on retrouve tous les utilisateurs qui se sont un jour connecté sur une licence.

7.2.2 Personnalisation et interactivité

Des filtres ont été ajoutés pour répondre aux besoins spécifiques, tels que le contrôle de la période de visualisation des données et la clarté des visualisations. Sur chaque page, il est possible de trier les données par fournisseur ou par technologie de licence. De plus, les utilisateurs peuvent manipuler la période d'observation des données, par exemple, pour afficher les données des trois derniers mois. Enfin, ils ont également la possibilité de trier les données par nom d'utilisateur, afin de voir quelles licences ont été utilisées par chaque utilisateur.

Concernant la charte graphique, elle est conforme à celle du site de Manitou-Group, tout en incluant les logos du groupe ainsi que de l'équipe R&D IS.

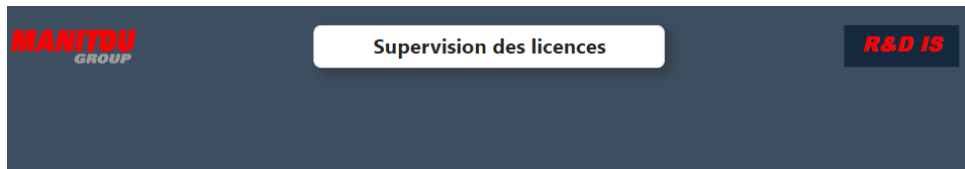


FIGURE 7.1 – Design d'une page du rapport

7.3 Problèmes rencontrés et solutions apportées

7.3.1 Données Ambiguës

Problème

Lors de la présentation de la maquette au métier, le contenu associé aux termes "fournisseur" et "licence" s'avère peu compréhensible pour ce dernier. Il s'agit en réalité de données techniques qui échappent au domaine de compréhension du métier.

Solution

Pour remédier à cette situation, il est envisagé de créer une table de transcoding agissant comme un lexique, associant les données techniques à leur signification respective. Cette table, créée manuellement, sera intégrée sur GCP à l'aide de Talend. Ensuite, DBT sera utilisé pour créer une nouvelle table avec les traductions. Cela permettra, sur PowerBI, de faire le lien entre la licence et les informations de la table de transcodage.

Modifications

Sur PowerBi, toutes les mentions des fournisseurs ont été remplacées par les noms des logiciels. Les informations sur les product et business owner ont également été intégrées aux vues, lorsque cela s'est avéré nécessaire. De plus, une nouvelle page a été ajoutée spécifiquement pour afficher les détails des fournisseurs. Cette page sera utilisée pour trouver un fournisseur en cas de besoin de support ou de problème.

7.4 Documentation

Enfin, une documentation a été élaborée pour accompagner le rapport PowerBi afin d'offrir une explication détaillée de chaque page, ainsi que pour décrire le fonctionnement des filtres.

Consulter l'Annexe F pour avoir une vue des pages du rapport

Chapitre 8

Déploiement de la solution

8.1 Mise en Qualif

Avant la mise en production, la solution doit être mise en qualif, c'est-à-dire déployée sur un environnement similaire à la production pour tester son fonctionnement, comme expliqué dans la section 1.4.1 **Infrastructure Informatique**.

La première étape consiste à déposer la solution sur Nexus*. Ensuite, il faut tester la solution dans des conditions réelles : les scripts doivent s'exécuter toutes les 15 minutes pour récupérer les données, et Talend doit les pousser sur GCP une à deux fois par jour. Il faut vérifier que tout fonctionne correctement, y compris le temps de traitement des fichiers.

Pour exécuter la solution, on utilise Opcon. Dans Opcon, deux paramètres sont importants :

- **Job Master** : Définit les options d'exécution, comme le chemin des jobs Talend dans Azure, la fréquence d'exécution des jobs et des scripts SQL.
- **Schedules** : Instances du job master. Pour exécuter un job master, il faut créer une instance. Il faut donc créer un job master pour chaque type d'exécution : scripts PowerShell, jobs Talend, et dbt.

Pour PowerBi, il suffit de publier le rapport et de définir une fréquence d'actualisation d'une à deux fois par jour, en synchronisation avec la fréquence d'intégration sur GCP.

Pour ce qui concerne DBT, pour l'instant il n'y a aucune certitude que ce framework sera utilisé d'ici 1 an. Pour cette raison la manière de déployer et le fonctionnement du framework en production n'a pas encore été étudié.

8.2 Mise en production

Après validation et tests en qualification, la solution est prête pour la mise en production, gérée par l'équipe CoreOps, et non par IT&D. Il faut créer des tickets de déploiement et fournir une documentation détaillée de déploiement. Le ticket signale l'intention de mettre en production, tandis que la documentation explique les étapes de déploiement, y compris le serveur cible, la version, et les scripts de déploiement nécessaires.

Il est possible de déployer plusieurs instances des jobs master selon les régions géographiques, avec des horaires différents par exemple pour les États-Unis et la France. Chaque mardi, une réunion (mep meeting) est tenue avec CoreOps pour valider les déploiements.

Je n'ai pas pu réaliser la mise en qualification et la mise en production en raison du manque de temps disponible. Cependant, j'ai préparé la documentation de déploiement et les scripts nécessaires pour que tout soit prêt pour la validation.

8.2.1 Scripts de déploiement

Les scripts de déploiement ont pour but de préparer le déploiement pour l'équipe CoreOps. L'objectif du projet est de copier la solution développée depuis l'environnement de développement vers l'environnement de production.

Pour cela, nous utiliserons deux scripts PowerShell :

- **packager_livraison.ps1** : Ce script copiera les dossiers et fichiers nécessaires vers l'environnement de production. Tous les éléments copiés seront placés dans un dossier nommé avec la version. Ce dossier ne doit pas être modifié, il sert à versionner chaque mise en production.
- **appliquer_livraison.ps1** : Ce script copiera l'intégralité du dossier de version à l'endroit où l'ordonnanceur pourra lancer les scripts.

Le premier script sera lancé depuis l'environnement de développement par M. LHERIAUD, tandis que le deuxième script sera exécuté par CoreOps. Les chemins des dossiers où la solution doit être copiée sont configurables dans les fichiers d'environnement.

Voici un schéma qui illustre le fonctionnement des deux scripts :

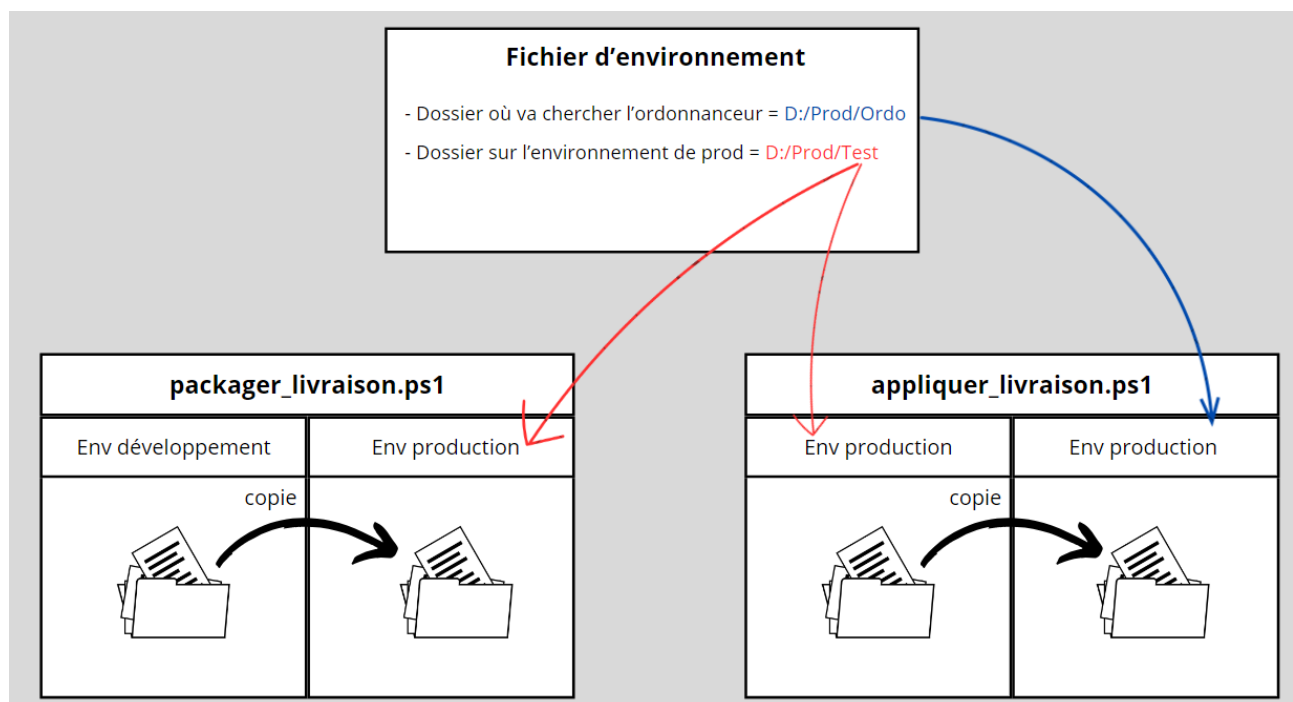


FIGURE 8.1 – Explication des scripts de déploiement

Consultez l' **Annexe G - Déploiement** pour avoir plus d'informations concernant les scripts.

Chapitre 9

Conclusion et Réflexion Personnelle sur l'ensemble du stage

9.1 Résultats

En résumé, les exigences initiales des lots 1 à 4 ont été remplies. L'extraction est opérationnelle, testée et documentée. L'intégration sur GCP fonctionne correctement et est maintenable grâce à la mise en variables des aspects susceptibles de changer au fil du temps. La consolidation avec DBT est fonctionnelle, testée, documentée et permet d'avoir des données structurées pour maintenir la cohérence et la lisibilité. Enfin, les maquettes répondent aux besoins métier et couvrent toutes les données nécessaires à la supervision, à l'exception des utilisateurs qui ont dû être exclus du périmètre en cours de projet.

9.2 Améliorations futures

État Actuel et Futurs Développements

L'architecture est désormais opérationnelle de bout en bout. Cependant, de nombreux outils doivent encore être ajoutés à la solution et les utilisateurs doivent être ajoutés.

Améliorations de Performance

Il est probable que des améliorations de performance soient nécessaires lors de l'intégration d'outils supplémentaires.

Prise en Compte des Besoins du Lot 5

Il restera à prendre en compte les besoins spécifiques du lot 5 dans le processus à savoir les alertes en cas d'utilisation de toutes les licences pour une technologie.

Automatisation de la Table de Transcoding

Actuellement, Manitou est en test d'un projet intitulé **EAS**. Ce logiciel, développé par l'entreprise du même nom, a pour objectif de gérer les couches d'architecture de l'entreprise (métier, applications, technologies, etc.). Il permet de répertorier toutes les composantes d'une application et les sujets sur lesquels elle a une influence.

L'objectif est de créer un lien entre ce logiciel et la table de transcoding afin de la remplir automatiquement. EAS pourrait permettre de récupérer les noms techniques de chaque licence

ainsi que les noms des logiciels, le *product owner* et le *business owner* associés. Cela permettra d'éviter ainsi les erreurs de saisie liées au remplissage manuel.

Planifier l'utilisation

Une amériolation possible, maintenant que les statistiques sont disponibles, pourrait être une planification de l'utilisation des licences. Par exemple pour un logiciel disposant de peu de licences et n'ayant pas besoin d'être utilisé régulièrement, il serait possible de définir des jours d'utilisations. Cela va permettre d'éviter de racheter des nouvelles licences qui seront utilisées une fois par semaine. L'objectif étant toujours d'améliorer les coûts de l'entreprise tout en gardant un confort de travail pour les employés.

9.3 Conclusion personnelle sur l'ensemble du stage

Mon expérience chez Manitou Group a été extrêmement satisfaisante, marquée par un accueil chaleureux de la part des employés qui m'ont permis de m'intégrer facilement dans l'équipe. Le sujet de mon stage était bien défini, facilitant ainsi mon adaptation à l'entreprise. L'environnement de travail agréable, enrichi par de nombreuses infrastructures, a cependant présenté une difficulté : le langage spécifique à Manitou, avec ses multiples termes pour désigner une même chose.

L'écoute et la disponibilité des employés ont été particulièrement appréciables, m'offrant l'opportunité d'apprendre et d'échanger régulièrement sur leurs métiers et missions. Ces interactions ont renforcé ma compréhension de l'importance du travail d'équipe en milieu professionnel. La participation à un séminaire de deux jours organisé par l'équipe Product Management a été une expérience enrichissante, améliorant la cohésion d'équipe et consolidant mes perspectives pour l'avenir.

À la fin du projet, bien que je n'ai pas pu aller jusqu'à sa mise en production, j'ai pris conscience des défis constants et des imprévus liés à la mise en place d'un projet en entreprise. Cette expérience a souligné l'importance de la flexibilité, de la gestion des risques et de la communication avec toutes les équipes impliquées. Malgré les défis, je suis fier d'avoir donné mon maximum pour mener le projet à terme, en réalisant un travail utile à l'entreprise.

9.4 Compétences Acquises

Les compétences que j'ai acquises durant ce stage sont principalement techniques, comme la maîtrise de PowerShell, PowerBi, ainsi que des notions de base sur Talend et DBT. Par ailleurs, ce stage m'a permis de faire évoluer certaines compétences existantes. En effet, il m'a aidé à améliorer mes compétences en communication, notamment grâce aux nombreuses réunions où je devais présenter l'avancement de mon projet de manière compréhensible pour tous. J'ai également développé une vision plus globale, que ce soit dans le domaine de l'informatique, de l'entreprise ou de la gestion de projet.

Ce stage m'a permis de découvrir de nombreux aspects de la vie en entreprise, passant de l'interaction avec mes collègues, à la gestion de projet dans une grande hiérarchie. Un des points les plus importants que j'ai retenu est l'importance de toujours penser à l'utilisateur final. Dans le cas de Manitou, il est crucial de se rappeler que cette entreprise fabrique des engins de manutention. L'objectif n'est donc pas de développer une solution extrêmement technique,

mais de réaliser un projet abouti et intuitif pour la suite de la chaîne. Pour le projet que j'avais à mener, j'ai constamment gardé à l'esprit qu'il devait évoluer et avoir pour but de faire gagner du temps aux employés.

9.5 Perspective de carrière

Ce stage ne m'a pas fait découvrir une nouvelle perspective de carrière, mais il m'a grandement conforté dans mon intérêt pour le monde de l'entreprise et de l'informatique. Il m'a également renforcé dans l'idée que ce qui me plaît, au-delà du développement, c'est de comprendre le côté fonctionnel d'un projet. J'apprécie de savoir ce que je développe et dans quel but, cela me permet non seulement de m'intéresser davantage à mon travail, mais aussi d'apporter un regard technique aux Business Analysts et d'adapter mon code aux besoins de l'utilisateur final.

Je ne trouve pas forcément intéressant de recevoir une spécification, d'écrire le code, et de passer ensuite à un autre projet. Au sein de Manitou, j'ai pu constater que les Business Analysts et les développeurs travaillent très régulièrement ensemble pour pallier leurs défauts respectifs et offrir à l'utilisateur le meilleur produit possible.

Aussi, avec l'arrivée de l'intelligence artificielle, je pense que le fait d'être plus axé sur le côté fonctionnel sera grandement bénéfique pour un développeur. Cet aspect étant propre à chaque entreprise et chaque projet, il est donc difficilement remplaçable par une intelligence artificielle.

Chapitre 10

Sources

10.1 Sources publiques

- <https://www.manitou.com/fr-FR#1> : Utilisé pour obtenir des informations sur l'entreprise.
- <https://learn.microsoft.com/en-us/power-bi/> : Documentation PowerBi pour la création du rapport
- <https://learn.microsoft.com/en-us/powershell/> : Documentation PowerShell pour la réalisation des scripts
- <https://docs.oracle.com/javase/8/docs/technotes/tools/windows/javadoc.html> : Documentation Java pour la création du projet
- <https://docs.getdbt.com/> : Documentation Dbt pour la consolidation des données
- <https://learn.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver16> : Documentation SQL Server
- <https://www.overleaf.com/learn> : Documentation Overleaf pour le rapport en LaTeX
- <https://stackoverflow.com/> : StackOverflow pour tous les apports et questions relatives au développement.
- <https://chatgpt.com/> : Utilisé pour des apports techniques lors d'un manque de documentation

10.2 Sources Internes

- **Site IT&D** : Site de l'IT&D sur l'intranet de l'entreprise
- **Nouveaux Arrivants** : Site pour les nouveaux arrivants dans l'entreprise
- **Glossaire** : Glossaire interne à l'entreprise
- **GAL** : Document répertoriant les projets déjà effectués
- **Codes** : Des exemples de code réalisés par M.LHERIAUD pour la partie PowerShell ainsi que par M.THIEBAUT-GEORGE pour DBT.
- **TalendDoc** : Documentation Talend avec des exemples de jobs et de composants
- **Templates** : Des templates de rapport PowerBI

Glossaire

2FA : Le 2FA, ou double authentification, est un mécanisme de sécurité qui requiert deux méthodes distinctes pour vérifier l'identité d'un utilisateur. Cela implique l'utilisation d'un mot de passe ainsi que d'une deuxième méthode, comme un code unique envoyé par SMS.

Artefact Logiciel : Un artefact logiciel est un fichier produit ou utilisé lors du processus de développement logiciel. Cela peut inclure des binaires compilés, des bibliothèques de code, des fichiers de configuration, etc..

BigQuery : BigQuery est un entrepôt de données, proposé par Google Cloud Platform (GCP), basé sur le cloud. Il permet de stocker, gérer et analyser de grandes quantités de données structurées et semi-structurées.

Certificat SSL : Un certificat SSL, ou Secure Sockets Layer, sécurise les échanges entre un serveur web et un navigateur en liant les détails de l'organisation à une clé cryptographique. Il protège de l'interception et la manipulation des données tout en garantissant une connexion sécurisée.

CSV : Un CSV (Comma-Separated Values) est un format de fichier utilisé pour stocker des données tabulaires, telles que celles d'une feuille de calcul ou d'une base de données. Dans un fichier CSV, chaque ligne correspond à un enregistrement ou une ligne de données, et chaque champ de l'enregistrement est séparé par une virgule ou un autre caractère.

ELT : Un ELT, ou Extraction, Load, Transformation, est un processus où les données sont d'abord extraites de différentes sources, chargées dans un entrepôt de données, puis transformées directement dans cet entrepôt. Contrairement à l'ETL (Extraction, Transformation, Load) comme Talend, où les données sont extraites, transformées sur un serveur intermédiaire, puis chargées dans l'entrepôt de données. ELT tire parti de la puissance de traitement du data warehouse pour effectuer les transformations.

ERP : Système logiciel qui permet à une organisation de gérer et d'intégrer l'ensemble de ses processus et de ses fonctions commerciales. Chez Manitou l'ERP s'appelle M3.

ETL : ETL est l'acronyme de Extract, Transform, Load, qui désigne un processus utilisé dans le domaine de la gestion de données. Ce processus consiste à extraire des données depuis différentes sources hétérogènes. Ensuite, ces données sont transformées pour les adapter aux besoins spécifiques de l'application Enfin, les données transformées sont chargées dans le système de destination.

Google Sheets : Google Sheets est une application en ligne de tableur, qui fait partie de Google Workspace. Elle permet de créer, de modifier et de partager des feuilles de calcul. Elle donne la possibilité d'entrer et organiser des données, effectuer des calculs avec des formules, créer des graphiques et collaborer en temps réel avec d'autres utilisateurs. Les documents sont automatiquement sauvegardés dans le cloud, ce qui facilite l'accès et la sécurité des données..

Industrie 4.0 : Intégration de technologies digitales intelligentes dans la production et les processus industriels. (IA, Robotique...).

ISIT : Information System and Information Technology.

JAR : Un fichier `.jar` est une archive Java, utilisée pour regrouper plusieurs fichiers en une seule archive, souvent pour distribuer des applications Java.

Jinja : Jinja est un moteur de template pour Python, permettant d'incorporer des variables, des structures de contrôle et des boucles dans des fichiers de modèle.

Nexus : Nexus Repository est un gestionnaire de dépôts développé par Sonatype, utilisé pour stocker, gérer et distribuer des artefacts* logiciels tels que des bibliothèques de code et des dépendances..

OPCON : Opcon est un ordonnanceur informatique conçu pour automatiser et orchestrer les workflows. Il permet la planification, l'exécution et la surveillance efficaces des tâches informatiques, offrant ainsi une optimisation des opérations.

PHISHING : Le phishing est une méthode d'attaque informatique où les fraudeurs utilisent des emails ou des messages frauduleux, souvent très semblables à ceux d'organisations légitimes, pour inciter les utilisateurs à divulguer des informations sensibles telles que des identifiants de connexion ou des données bancaires.

PLM : Un PLM, ou Product Lifecycle Management, est un système de gestion qui permet de gérer l'ensemble du cycle de vie d'un produit, depuis sa conception jusqu'à sa fin de vie. Il centralise toutes les informations relatives au produit, y compris les données techniques, les spécifications, les documents de conception, les processus de fabrication, les données de qualité, etc. Chez Manitou le PLM s'appelle Model Manager.

SpringBatch : Spring Batch est un framework Java open-source conçu pour le développement et l'exécution de traitements en batch.

SQL : Le SQL (Structured Query Language) est un langage de programmation utilisé pour communiquer avec les bases de données relationnelles. Il permet de manipuler et de gérer les données stockées dans une base de données, en effectuant des opérations telles que l'insertion, la mise à jour, la suppression et la récupération des données.

Suite CAO : Une Conception Assistée par Ordinateur est un ensemble de logiciels permettant de créer, modifier, analyser et documenter des modèles virtuels en 2D ou 3D de produits ou de composants..

UUID : Un Universally Unique Identifier est un identifiant unique universellement utilisé pour identifier de manière unique une ressource ou une entité dans un système informatique. Il est généralement représenté par 32 caractères hexadécimaux.

Annexe A

Cahier des charges

A.1 Contexte et présentation de la mission

Quotidiennement, les employés utilisent des licences pour avoir accès à des logiciels. Les licences sont fournies grâce à des fournisseurs et leurs serveurs. L'entreprise souhaite se servir des services fournis par les outils de gestion des licences pour faire des statistiques sur les licences comme le taux d'utilisation ou le maximum d'utilisation. Les données sont uniquement consultables manuellement. La consultation prend du temps et entraîne donc une perte financière. La mission du stagiaire est d'automatiser la récupération des données en provenance des serveurs de licences. Il devra ensuite uniformiser les données avant de les intégrer dans un entrepôt. Enfin il mettra ces données à disposition des utilisateurs finaux.

Ce projet ayant été initié il y a 1 an, il existe une base de code sur laquelle pourra s'appuyer le stagiaire. La planification des besoins sera effectuée avec le stagiaire afin qu'il puisse cerner les attentes de l'entreprise vis à vis du projet.

L'entreprise prévoit d'effectuer le suivi du projet avec les éventuelles corrections de bugs et de mises à jour. Le stagiaire n'a aucune obligation de résultat, en revanche tout le travail effectué se doit d'être documenté.

A.2 Acteurs Impliqués

Le projet étant initié par un besoin de l'entreprise ainsi que des employés, la MOA est donc Manitou Group. Le product owner du projet est M.GALLARD, manager de l'équipe R&D IS. Le business owner du projet côté métier est M.RENOU.

La maîtrise d'œuvre pour ce projet est composée de l'équipe R&D IS ainsi que du stagiaire. Des intervenants d'autres équipes seront amenés à interagir avec le projet.

A.3 Confidentialité

Toutes les données concernant les licences logicielles sont confidentielles et ne doivent pas apparaître dans le rapport ou dans la soutenance. En revanche le code est utilisable tant qu'il ne comporte pas d'informations susceptibles de nuire à la sécurité de l'entreprise.

A.4 Livrables

Comme précisé dans la présentation de la mission, il n'y a pas d'obligations de résultats et par conséquent pas de livrables attendus. Cependant, tout le travail effectué par le stagiaire (code, documentation, ...) devra être accessible à la fin de la période de stage. Une copie du rapport de stage sera également conservée par l'entreprise.

Annexe B

Informations supplémentaires sur l'extraction des données

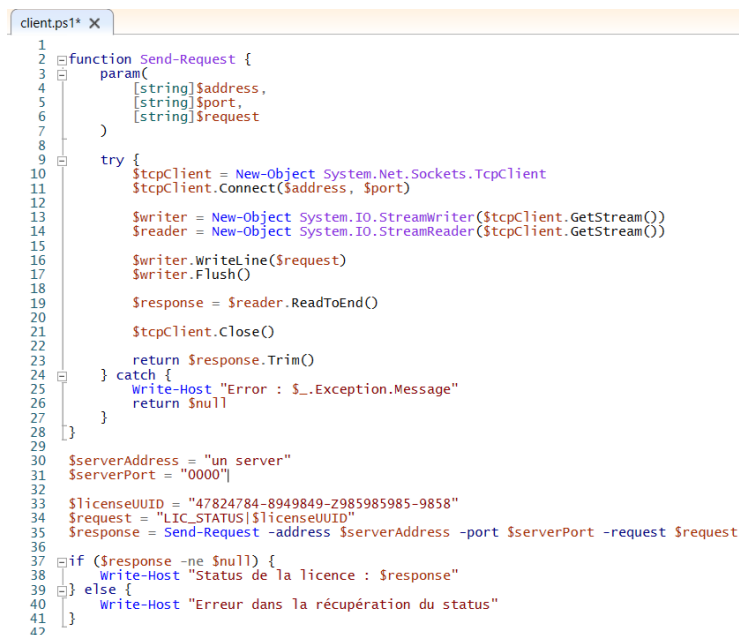
B.1 Présentation de l'extraction pour un outil :

B.1.1 Contexte

Comme expliqué dans le rapport, l'objectif était de récupérer les données nécessaires à la supervision des outils de licence. Toutefois, pour l'extraction concernant l'outil que je vais vous présenter, nous avons rencontré un problème : le fichier de licence, qui contenait habituellement les données répondant au besoin, était cette fois-ci chiffré. Il a donc été nécessaire de trouver un moyen de déchiffrer ces données ou de les récupérer autrement. Avant de chercher une solution alternative, j'ai voulu vérifier si nous pouvions obtenir les informations non chiffrées en interrogeant directement le serveur.

Pour ce faire, j'ai créé un client pour interroger le serveur :

.txt :



```
1 function Send-Request {
2     param(
3         [string]$address,
4         [string]$port,
5         [string]$request
6     )
7
8
9     try {
10         $tcpClient = New-Object System.Net.Sockets.TcpClient
11         $tcpClient.Connect($address, $port)
12
13         $writer = New-Object System.IO.StreamWriter($tcpClient.GetStream())
14         $reader = New-Object System.IO.StreamReader($tcpClient.GetStream())
15
16         $writer.WriteLine($request)
17         $writer.Flush()
18
19         $response = $reader.ReadToEnd()
20
21         $tcpClient.Close()
22
23         return $response.Trim()
24     } catch {
25         Write-Host "Error : $_.Exception.Message"
26         return $null
27     }
28 }
29
30 $serverAddress = "un server"
31 $serverPort = "0000"
32
33 $licenseUUID = "47824784-8949849-Z985985985-9858"
34 $request = "LIC_STATUS|$licenseUUID"
35 $response = Send-Request -address $serverAddress -port $serverPort -request $request
36
37 if ($response -ne $null) {
38     Write-Host "Status de la licence : $response"
39 } else {
40     Write-Host "Erreur dans la récupération du status"
41 }
42 }
```

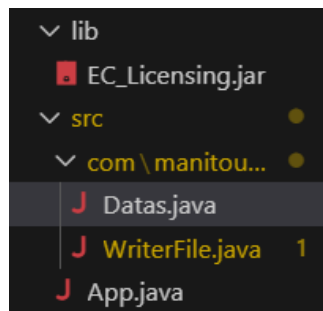
FIGURE B.1 – Client

En observant la réponse renvoyée par le serveur, j'ai compris que je devrais procéder différemment pour récupérer les données souhaitées initialement, car le serveur renvoyait quand

[illegible]

B.1.2 Contexte du projet Java

B.1.3 Architecture du projet Java :



L'architecture était organisée de la manière suivante :

- 52

B.1.4 Illustrations du code

Voici un exemple de fonctions permettant de récupérer les données en interrogeant le serveur :

```
/**
 * Function to retrieve all the license data from the server using the JAR file.
 * @param address - Server (REKNOW) address
 * @param port - Server port
 * @return licences - Response from the server with an ArrayList of LicenseData
 */
public ArrayList<REKNOW_LicenseData> getLicences(String address, String port){
    return REKNOW_LicenseServerConnector.getLicences(address, port);
}

/**
 * Returns the status of the license based on an integer parameter.
 * @param status - Response from the server for the status of the license,
 * @return status|defaultStatus - Returns the string that corresponds to the status or returns the default status.
 */
public String getStatusString(int status){
    return statusMap.getOrDefault(status, defaultStatus);
}
```

FIGURE B.4 – Exemple de récupération de données via le serveur

Ensuite, il faut écrire le fichier avec les données récupérées en appelant les fonctions illustrées ci-dessus.

```
/**
 * Creates the list of strings containing the license data and writes it to a file.
 * @param filePath The destination file path.
 * @param address The server address (REKNOW).
 * @param port The server port.
 */
public void write(String filePath, String address, String port) {
    ArrayList<REKNOW_LicenseData> licences = Datas.getInstance().getLicences(address, port);
    List<String> lines = new ArrayList<>();
    lines.add(e:"productName type status startDate endDate user ip");

    licences.forEach(licence -> {
        String licenceType = Datas.getInstance().getTypeString(licence.type);
        String licenceStatus = Datas.getInstance().getStatusString(licence.status);
        String startDate = Datas.getInstance().getDateString(licence.start_date);
        String endDate = Datas.getInstance().getDateString(licence.end_date);

        String user = licence.user.isEmpty() ? "NO_USER" : licence.user;
        String ipAddressClient = licence.ip_address_client.isEmpty() ? "NO_IP" : licence.ip_address_client;

        lines.add(String.format(format:"%s\t%s\t%s\t%s\t%s\t%s\t%s",
            licence.product_name, licenceType, licenceStatus, startDate, endDate, user, ipAddressClient));
    });

    writeTextFile(lines, filePath);
}
```

FIGURE B.5 – Ecriture du fichier

Enfin, il faudra lancer le projet Java au moment de l'exécution des scripts.

B.1.5 Présentation des scripts PowerShell

Fichier de configuration

Voici un exemple de fichier de configuration :

```
malt.app.code=Code pour identifier le fichier de configuration
malt.app.manager=Nom de l'outil
malt.app.licencefile=chemin/vers/un/fichier
malt.app.serverAdress=Adresse du serveur de l'outil
malt.app.serverPort=Port sur lequel répond le serveur
malt.app.filePath=Chemin vers du fichier renvoyé par java
```



```

##
.SYNOPSIS
    Function to fill the licenceUser, max, and inUse.
.DESCRIPTION
    Function to fill the licenceUser in the licenceStats, fill the max in the licenceStats, and fill the inUse in the licenceStats.
.PARAMETER vendors
    An array containing vendors.
.PARAMETER FileContent
    A System.Object[] containing the content of the license file.
.PARAMETER pattern
    A string regex that facilitates the extraction of elements from the FileContent.
.PARAMETER MaxCount
    A hashtable containing the maximum count for each license.
.PARAMETER UsageCount
    A hashtable containing the usage count for each license.
.EXAMPLE
    FillLicenceUserMaxAndUsageCount -vendors $vendors -FileContent $FileContent -pattern $pattern -maxCount $maxCount -usageCount $usageCount
#>
function FillLicenceUserMaxAndUsageCount{
    [CmdletBinding()]
    param (
        [Parameter(Mandatory)]
        [System.Collections.ArrayList]$vendors,
        [Parameter(Mandatory)]
        [System.Object[]]$FileContent,
        [Parameter(Mandatory)]
        [string]$pattern,
        [Parameter(Mandatory)]
        [hashtable]$maxCount,
        [Parameter(Mandatory)]
        [hashtable]$usageCount
    )
}

```

FIGURE B.9 – Exemple d’une fonction de remplissage des données

```

$currentUser = @()
$licenceNameAndUsername = @()
foreach ($line in $FileContent) {
    if ($line -match $pattern) {
        $matches = $line | Select-String $pattern
        foreach ($match in $matches) {
            # Declare attributes necessary to fill licence stats
            $currentUser = [LicenceUser]::new()
            $currentUser.username = $match.Matches.Groups[6].Value
            $currentUser.techref = $match.Matches.Groups[7].Value
            $currentUser.version = $toolsEnv['malt.app.version']
            $name = $match.Matches.Groups[1].Value
            $use = $match.Matches.Groups[3].Value
            # Retrieve the license in the vendors array corresponding to the loop's license
            $desiredLicence = $vendors | ForEach-Object { $_.licences } | Where-Object { $_.name -eq $name }
            if ($use -eq "IN_USE") {
                $usageCount[$name]++
                $currentUser += $name + "`t" + $match.Matches.Groups[6].Value
                if ($desiredLicence) {
                    $desiredLicence.inUse = $usageCount[$name]
                    # Add the current users to the current license
                    $desiredLicence.AddUser($currentUser)
                }
            }
            # Add to the retrieved license the maximum number of available licenses and the current usage count
            $desiredLicence.max = $maxCount[$name]
        }
    }
}
return $currentUser
}

```

FIGURE B.10 – Exemple d’une fonction de remplissage des données suite

Il faut maintenant transformer les données récupérées en **.csv**. Les csv porteront le nom de l’outil de gestion en cours d’extraction ainsi que la date à laquelle les données sont extraites. Comme nous souhaitons lancer un seul csv pour tous les outils il faudra utiliser le code du fichier d’environnement pour différencier les outils. Pour cela on va utiliser la méthode **Add-Content** de powershell est utilisée. Cette méthode va permettre de spécifier un chemin correspondant à l’endroit où l’on veut créer le fichier. Il faudra également lui fournir le contenu à ajouter dans ce fichier :


```

<#
SYNOPSIS
    Fonction de consigner les statistiques des utilisateurs dans un fichier.
DESCRIPTION
    Fonction de consigner les statistiques des utilisateurs de licences dans un fichier.
OUTPUTS
    Fichier csv contenant les stats des utilisateurs
EXAMPLE
    UsersToFile -vendor $vendor -fileName "D:\Temp\toto.csv"
#>
Function UsersToFile {
    param(
        [Parameter(Mandatory)]
        [LicenceVendor]$vendor,
        [Parameter(Mandatory)]
        [string]$fileName
    )

    # La partie du vendor propre au vendor est fixe, il n'y a besoin de la definir qu'une seule fois
    $vendorPrefix = (StringToCsv -str $vendor.licFile) + ',' + (StringToCsv -str $vendor.name) + ',' + (StringToCsv -str $vendor.version) + ',';
    $extractionDate = (Get-Date -Format "yyyyMMdd HHmmss");
    # Parcours des differentes licences geres par ce vendor
    $vendor.licences | ForEach-Object {
        $lic = $_;
        # La partie propre a la licence est fixe pour tous les utilisateurs de la licence
        $licPrefix = $vendorPrefix + (StringToCsv -str $lic.name) + ',' + (StringToCsv -str $lic.version) + ',';
        $lic.users | ForEach-Object {
            $user = $_;
            #Version uuid : 4
            $uuid = [guid]::NewGuid().ToString();
            $line = (StringToCsv -str $uuid) + ',' + (StringToCsv -str $extractionDate) + ',' + $licPrefix + (StringToCsv -str $user.version) + ','
            Add-Content -Value $line -Path $fileName;
        }
    }
}

```

FIGURE B.11 – Exemple d’une fonction création d’un CSV

Exécution de tous les scripts

Toutes les fonctions et fichiers présentés ici ne représentent pas l’intégralité de la méthode d’extraction. De nombreuses autres fonctions sont nécessaires pour remplir correctement les objets de classes.

Annexe C

Informations supplémentaires sur le serveur FTP

Dans cette partie je vais détailler un peu plus précisément comment fonctionne les requêtes vers le serveur FTP en PowerShell.

C.1 Les Web Requests en PowerShell

Les Web Requests en PowerShell permettent d'effectuer des opérations de réseau telles que le transfert de fichiers, l'envoi de données à des serveurs web, ou la récupération de données depuis des serveurs. Dans le cadre des opérations FTP, PowerShell utilise principalement la classe `System.Net.FtpWebRequest` du `.NET Framework` pour interagir avec les serveurs FTP.

C.1.1 La Classe `System.Net.FtpWebRequest`

La classe `System.Net.FtpWebRequest` fournit des fonctionnalités pour interagir avec un serveur FTP. Elle permet de réaliser des opérations qui nous intéressent dans le cadre du projet comme transférer des fichiers.

C.1.2 Principales Méthodes et Propriétés

Create

Create est une méthode permettant d'initialiser une requête FTP pour une URL de serveur et un port donnés.

Method

Permet de spécifier le type d'opération FTP à effectuer. Voici une liste des options utilisées dans le projet :

- `UploadFile` : Pour transférer un fichier.
- `DeleteFile` : Pour supprimer un fichier. Utilisé uniquement pour tester le fonctionnement.
- `MakeDirectory` : Pour créer un répertoire.
- `RemoveDirectory` : Pour supprimer un répertoire. Utilisé uniquement pour tester le fonctionnement.
- `Ftp.ListDirectory` : Pour lister les fichiers dans un répertoire.

Credentials

Permet de gérer les informations d'authentification (nom d'utilisateur et mot de passe) nécessaires pour se connecter au serveur FTP.

UseBinary et UsePassive

Indique si le transfert de données doit se faire en mode binaire. En mode binaire, les fichiers sont transférés sans aucune modification. C'est très important surtout pour des fichiers **image** ou **vidéos** car une simple modification pourrait les rendre totalement inutilisable.

Indique si le transfert de données doit se faire en mode passif. En mode passif, c'est le client qui va établir la connexion vers le serveur.

GetResponse

Envoie la requête au serveur FTP et récupère la réponse.

C.1.3 Processus de Transfert de Fichiers FTP en PowerShell

Voici les étapes pour transférer un fichier vers le serveur FTP en PowerShell

Utiliser la méthode Upload File

Lire le Contenu du Fichier

Pour cela je vais utiliser la méthode `ReadAllBytes` de la classe `System.IO.File` pour lire le contenu du fichier local.

Définir la Longueur du Contenu

Il faut ensuite définir la longueur du fichier à envoyer. Cela permet au serveur FTP de savoir combien de données il va recevoir.

Obtenir le Flux de Requête

Un "Flux" fait référence à une séquence continue de données. Obtenir le flux de requête signifie obtenir un chemin pour envoyer ou recevoir des données via la connexion FTP.

Écrire les données et fermer le flux

Enfin, il faut écrire les données dans le flux et le fermer pour libérer les ressources allouées au transfert.

C.1.4 Exemples d'utilisations

Voici des exemples d'utilisation de toutes ces méthodes :

```

try {
    $ftpUrl = $ftpConnection.FtpUrl + $remoteFilePath
    $ftp = [System.Net.FtpWebRequest]::Create($ftpUrl)
    $ftp.Method = [System.Net.WebRequestMethods+Ftp]::UploadFile
    $ftp.ConnectionGroupName = $ftpConnection.Username

    $ftp.Credentials = New-Object System.Net.NetworkCredential($ftpConnection.Username, $ftpConnection.Password)
    $ftp.UseBinary = $true
    $ftp.UsePassive = $true

    # Read the content of the file to upload
    $fileContent = [System.IO.File]::ReadAllBytes($localFilePath)
    $ftp.ContentLength = $fileContent.Length

    $requestStream = $ftp.GetRequestStream()
    $requestStream.Write($fileContent, 0, $fileContent.Length)
    $requestStream.Close()
    $requestStream.Dispose()

    $response = $ftp.GetResponse()
    $response.Close()

    Write-Output "File successfully uploaded to $ftpUrl"
    # Remove the file that was just uploaded to avoid double upload in case of an issue
    Remove-Item -Path $localFilePath
}

```

FIGURE C.1 – Transférer un fichier sur le serveur FTP

```

function New-FtpDirectory {
    [CmdletBinding()]
    param (
        # FTP connection information
        [Parameter(Mandatory)]
        [hashtable]$ftpConnection,

        # Remote path to create the directory at
        [Parameter(Mandatory)]
        [string]$DirectoryPath
    )

    try {
        # Build the full FTP URL for the directory to create
        $ftpUrl = $ftpConnection.FtpUrl + $DirectoryPath

        # Create the FtpWebRequest object
        $ftp = [System.Net.FtpWebRequest]::Create($ftpUrl)
        $ftp.Method = [System.Net.WebRequestMethods+Ftp]::MakeDirectory
        $ftp.Credentials = New-Object System.Net.NetworkCredential($ftpConnection.Username, $ftpConnection.Password)
        $ftp.UseBinary = $true
        $ftp.UsePassive = $true

        # Get the FTP server response
        $response = $ftp.GetResponse()
        $response.Close()

        Write-Output "Directory successfully created at $ftpurl"
    }
    catch {
        Write-Error "An error occurred while creating the directory (the directory may already exist): $_"
    }
}

```

FIGURE C.2 – Créer un nouveau répertoire sur le serveur FTP

Annexe D

Informations supplémentaires sur Talend

D.1 Normes de développement

Dans cette partie je voulais apporter un peu plus de détails sur les normes de développement. En effet dans la partie 5.3.4 **Exécution des jobs** j'avais seulement abordé les jobs et leur noms mais à l'intérieur des jobs il y a également d'autres normes de développement.

Au delà des jobs tous les composants utilisés doivent avoir un nom clair qui permet de comprendre ce qu'il fait sans même regarder ses paramètres.

Il ne faut pas qu'il y ait de chevauchement entre les flux où les composants. Les lignes de type flux de données et itération doivent aller de gauche à droite tandis que les lignes de type déclencheur vont de haut en bas.

Une règle obligatoire est que chaque nouveau projet doit faire partie d'une nouvelle branche et il ne doit jamais être ajouté sur la branche principale.

Enfin, une des normes les plus importantes est de tout variabiliser. Cela permet d'éviter d'avoir à changer tous les endroits où la donnée est présente en plus d'éviter les erreurs liés aux différents passages entre les environnements de développement et de production

Il existe d'autres normes de développement mais celles que j'ai citées sont celles que j'ai dû respecter.

D.2 Talend Context

Dans l'optique de respecter la mise en variable de tous les dossiers, noms apparaissant dans les composants, il existe sur Talend une fonctionnalité appelée **Context**. Le contexte d'un projet sur talend va contenir des variables avec des valeurs qui lui sont associées :

	Name	Type	Comment	Default		Deploy
				Value	Enable prompt	Value
1	vTalendJs	String		jobName	<input type="checkbox"/>	Js_licenses_datas_to_datawarehouse_01

FIGURE D.1 – Exemple d'une variable représentant le nom d'un job

Cela permet de changer qu'à un seul endroit le nom du job pour changer toutes les occurrences de celui ci.

Il est également possible de faire autre chose avec le contexte. Ce dernier peut aller chercher directement dans un fichier de propriété. Il suffit donc dans ce fichier de préciser le nom

de la variable et la valeur pour que le context connaisse la valeur de la variable peu importe l'environnement sur lequel on exécute le job.

```
fichier_rep_temp==D:/Talend/Temp/
```

FIGURE D.2 – Exemple de ligne du fichier de propriété



FIGURE D.3 – Ajouter la variable dans le contexte

Pour ajouter la variable dans le contexte il faut configurer Talend pour lui dire d'aller chercher dans le fichier où il y a les variables et en cliquant à l'endroit où pointe la flèche rouge sur la figure C.3 – Ajouter la variable dans le contexte. Il est ensuite possible d'aller chercher le nom de la variable dans le fichier pour que la valeur se remplisse automatiquement.

D.3 Composants Talend

Pour que les composants fonctionnent il faut les paramétrer correctement avec les bonnes informations par exemple pour se connecter à GCP le composant à besoin des valeurs suivantes :

FIGURE D.4 – Exemple de paramétrage de composant Talend

Ces composants sont générés avec du code Java par Talend. Cependant il est également possible d'écrire directement du code java pour déclarer des variables, effectuer des calculs, etc... Dans le cadre de ce projet je n'ai pas eu besoin d'utiliser ce composant car ce que j'avais besoin de faire n'était pas assez complexe pour avoir besoin d'écrire du java supplémentaire. En revanche ça reste une option très pratique lorsque les composants mis à disposition ne suffisent pas.

D.4 Autres jobs développés

En respectant toutes ces normes, voici le reste des jobs développés pour répondre au besoin initial.



FIGURE D.5 – Js_Licenses_datas_to_datawarehouse_01

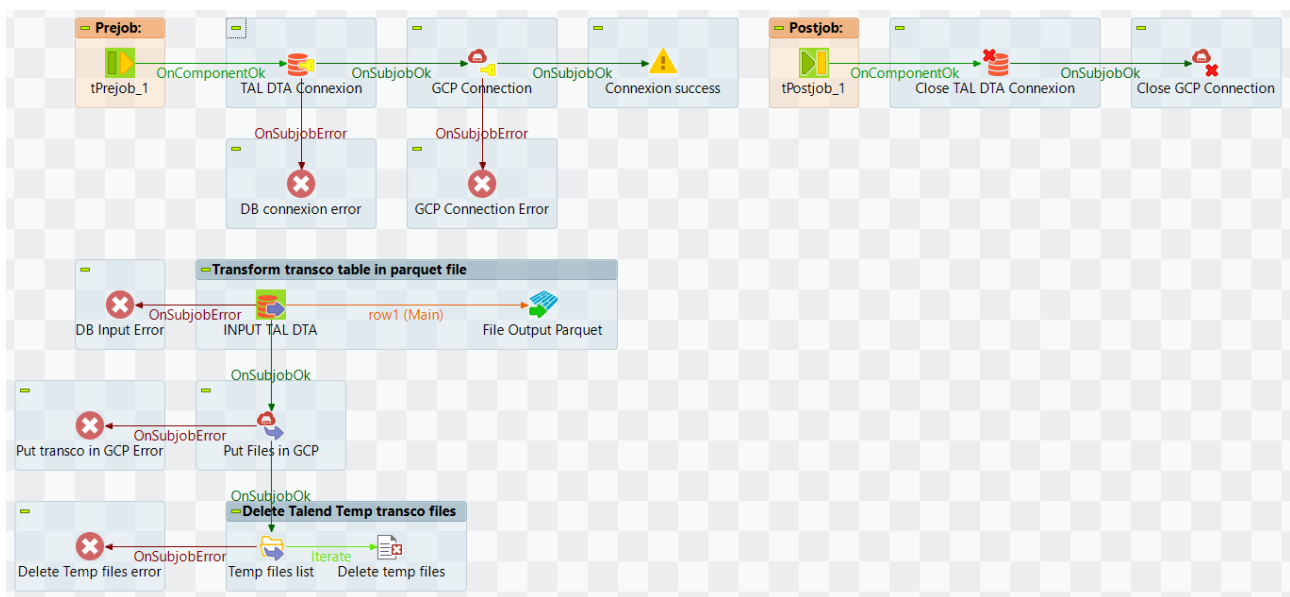


FIGURE D.6 – Jd_Licenses_datas_to_datawarehouse_020_push_transco

La description des jobs se trouve dans la partie 5.3.4 – Présentation des jobs réalisés

Annexe E

Informations supplémentaires sur dbt

E.0.1 Fichier de configuration

```
licensesMonitoring :
  target: dev
  outputs:
    dev:
      type: sqlserver
      driver: 'ODBC Driver 18 for SQL Server' # (The ODBC Driver installed on your system)
      server: unServer
      port: 1433
      database: uneDataBase
      schema: unSchema
      user: arthur
      password: millet
      trust_server_certificate: true
```

FIGURE E.1 – Exemple de fichier de configuration

E.0.2 Exemple de modèle

```
-- Creating a table in the final select using dbt.
{{ config(materialized='table') }}
```

```
-- Retrieve data from the transco table created in GCP
WITH unique_vendor_data AS (
  SELECT DISTINCT
    vendorName,
    vendorVersion
  FROM
    -- Replace with the name of the table on GCP
    transcoAndLicensesInfos AS t

  -- Construct a table with the desired attributes and adding a unique id for each insertion
), vendor_data_with_id AS (
  SELECT
    ROW_NUMBER() OVER (ORDER BY vendorName) AS idVendor,
    vendorName,
    vendorVersion
  FROM
    unique_vendor_data
)

SELECT
  *
FROM
  vendor_data_with_id;
```

FIGURE E.2 – Modèle de la table Vendors

E.0.3 Documentation

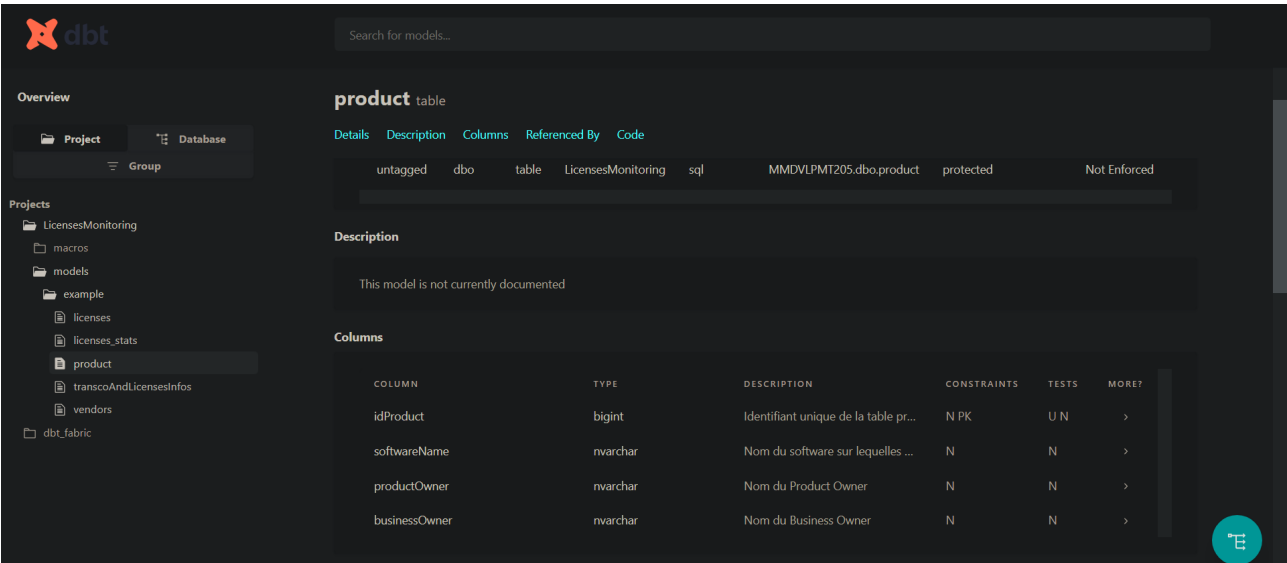


FIGURE E.3 – Page d’accueil de la documentation

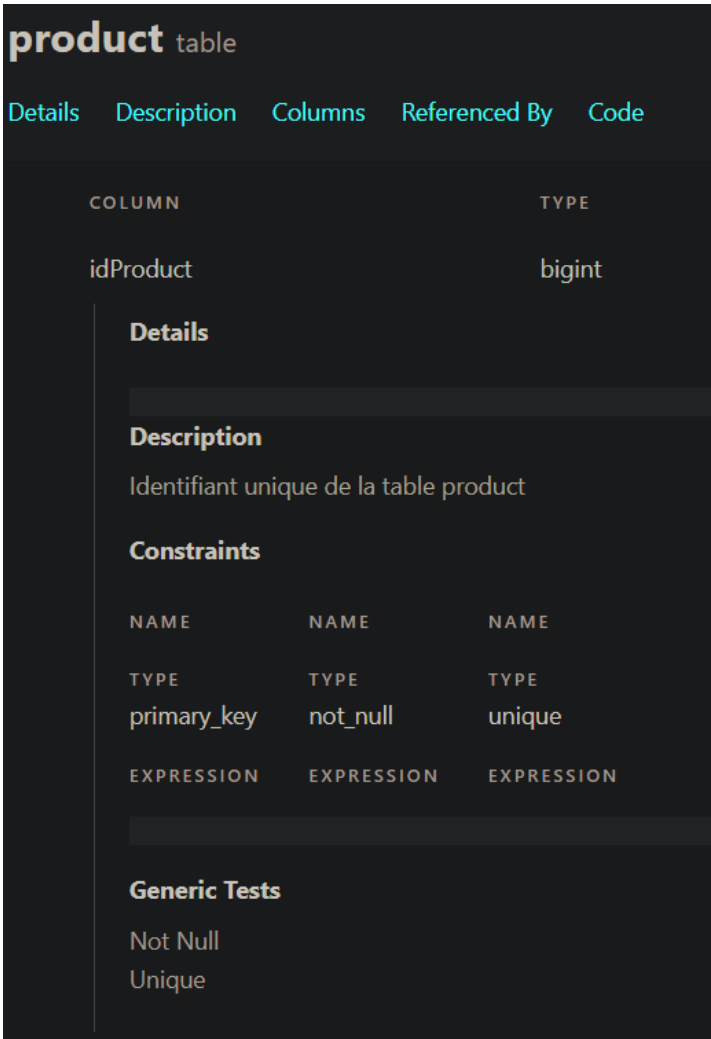


FIGURE E.4 – Description d’une colonne (contraintes, tests, description)

E.0.4 Tests

```
12:19:07 19 of 23 START test not_null_vendors_vendorVersion ..... [RUN]
12:19:07 19 of 23 PASS not_null_vendors_vendorVersion ..... [PASS in 0.07s]
12:19:07 20 of 23 START test unique_licenses_idlicense ..... [RUN]
12:19:08 20 of 23 PASS unique_licenses_idlicense ..... [PASS in 0.06s]
12:19:08 21 of 23 START test unique_licenses_stats_idLicenseStats ..... [RUN]
12:19:08 21 of 23 PASS unique_licenses_stats_idLicenseStats ..... [PASS in 0.05s]
12:19:08 22 of 23 START test unique_product_idProduct ..... [RUN]
12:19:08 22 of 23 PASS unique_product_idProduct ..... [PASS in 0.06s]
12:19:08 23 of 23 START test unique_vendors_idVendor ..... [RUN]
12:19:08 23 of 23 PASS unique_vendors_idVendor ..... [PASS in 0.05s]
12:19:08
12:19:08 Finished running 23 tests in 0 hours 0 minutes and 1.98 seconds (1.98s).
12:19:08
12:19:08 completed successfully
```

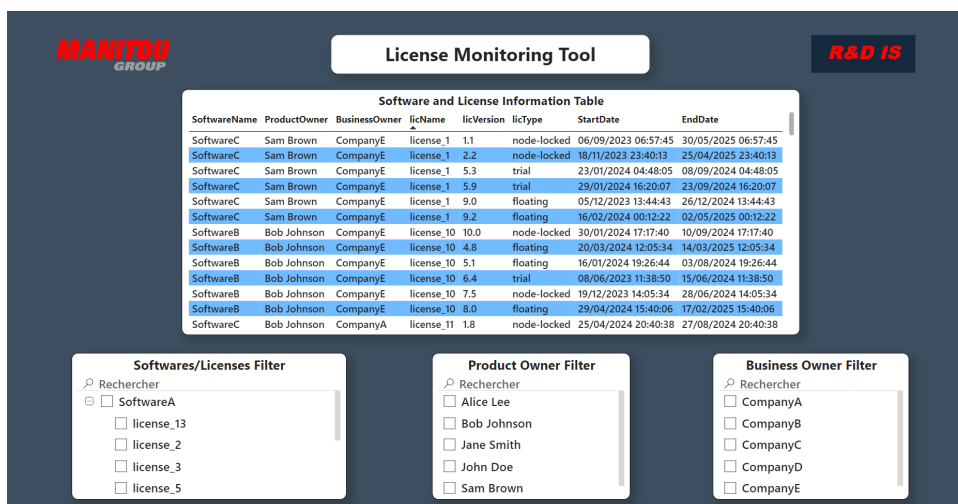
FIGURE E.5 – Tests Validés

Annexe F

Présentation et rendu des rapports Power BI

Certaines sections seront floutées pour des raisons de confidentialité et de sécurité.

F.1 Softwares et informations sur la licence (Page 1)



MANITOU GROUP License Monitoring Tool **R&D IS**

Software and License Information Table

SoftwareName	ProductOwner	BusinessOwner	licName	licVersion	licType	StartDate	EndDate
SoftwareC	Sam Brown	CompanyE	license_1	1.1	node-locked	06/09/2023 06:57:45	30/05/2025 06:57:45
SoftwareC	Sam Brown	CompanyE	license_1	2.2	node-locked	18/11/2023 23:40:13	25/04/2025 23:40:13
SoftwareC	Sam Brown	CompanyE	license_1	5.3	trial	23/01/2024 04:48:05	08/09/2024 04:48:05
SoftwareC	Sam Brown	CompanyE	license_1	5.9	trial	29/01/2024 16:20:07	23/09/2024 16:20:07
SoftwareC	Sam Brown	CompanyE	license_1	9.0	floating	05/12/2023 13:44:43	26/12/2024 13:44:43
SoftwareC	Sam Brown	CompanyE	license_1	9.2	floating	16/02/2024 00:12:22	02/05/2025 00:12:22
SoftwareB	Bob Johnson	CompanyE	license_10	10.0	node-locked	30/01/2024 17:17:40	10/09/2024 17:17:40
SoftwareB	Bob Johnson	CompanyE	license_10	4.8	floating	20/03/2024 12:05:34	14/03/2025 12:05:34
SoftwareB	Bob Johnson	CompanyE	license_10	5.1	floating	16/01/2024 19:26:44	03/08/2024 19:26:44
SoftwareB	Bob Johnson	CompanyE	license_10	6.4	trial	08/06/2023 11:38:50	15/06/2024 11:38:50
SoftwareB	Bob Johnson	CompanyE	license_10	7.5	node-locked	19/12/2023 14:05:34	28/06/2024 14:05:34
SoftwareB	Bob Johnson	CompanyE	license_10	8.0	floating	29/04/2024 15:40:06	17/02/2025 15:40:06
SoftwareC	Bob Johnson	CompanyA	license_11	1.8	node-locked	25/04/2024 20:40:38	27/08/2024 20:40:38

Software/Licenses Filter

Rechercher

☐ SoftwareA

☐ license_13

☐ license_2

☐ license_3

☐ license_5

Product Owner Filter

Rechercher

☐ Alice Lee

☐ Bob Johnson

☐ Jane Smith

☐ John Doe

☐ Sam Brown

Business Owner Filter

Rechercher

☐ CompanyA

☐ CompanyB

☐ CompanyC

☐ CompanyD

☐ CompanyE

FIGURE F.1 – Page numéro 1

Contenu

L'objectif de cette première page est de présenter des informations sur les licences, les logiciels et les administrateurs. Les détails sont répertoriés dans un tableau.

Filtres

Des filtres sont également disponibles pour vous aider à voir les informations plus clairement :

- Filtrer par nom de logiciel et nom de licence
- Filtrer par propriétaire du produit
- Filtrer par propriétaire commercial

Advice

Vous pouvez rechercher directement le nom que vous cherchez dans les filtres :

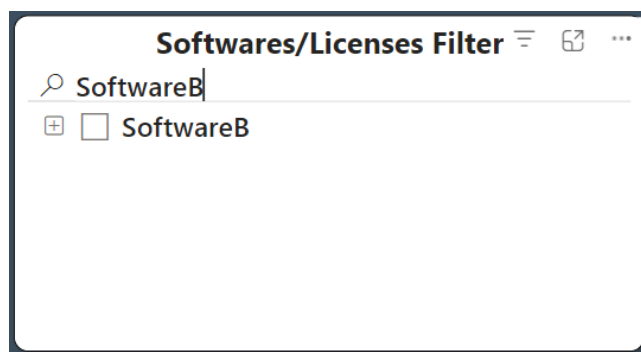


FIGURE F.2 – Recherche dans les filtres de la première page

Enfin, vous pouvez sélectionner plusieurs noms dans les filtres en maintenant la touche **ctrl** enfoncée et en cliquant.

F.2 Licences utilisées au maximum (Page 2)

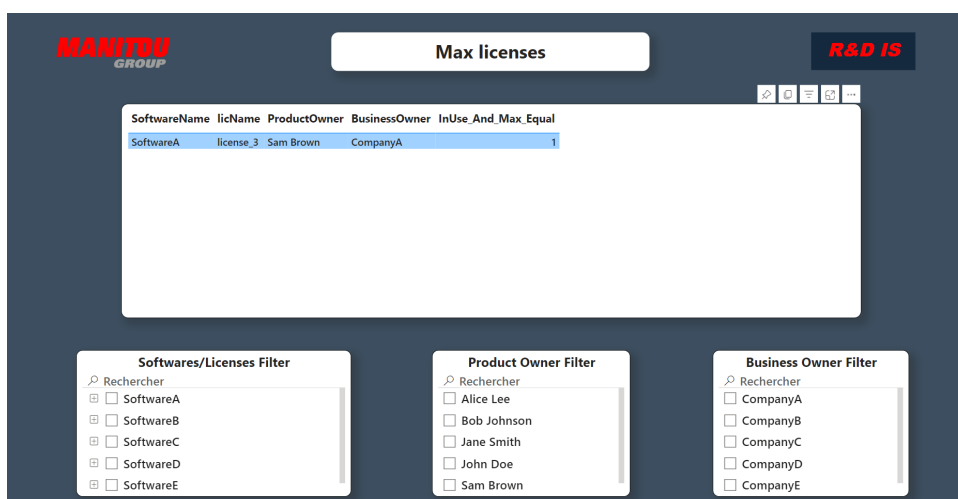


FIGURE F.3 – Page numéro 2

Content

Cette page permet de visualiser quelles technologies de licence ont toutes les licences en cours d'utilisation. Ces données seront répertoriées dans un tableau contenant :

- Noms des logiciels
- Noms des licences
- Propriétaire du produit
- Propriétaire commercial
- Utilisation maximale : Un paramètre actuellement sans information mais utilisé pour le filtrage.

Filters

Pour les filtres, les mêmes que sur la première page seront disponibles.

F.3 Statistiques de licences (Page 3)

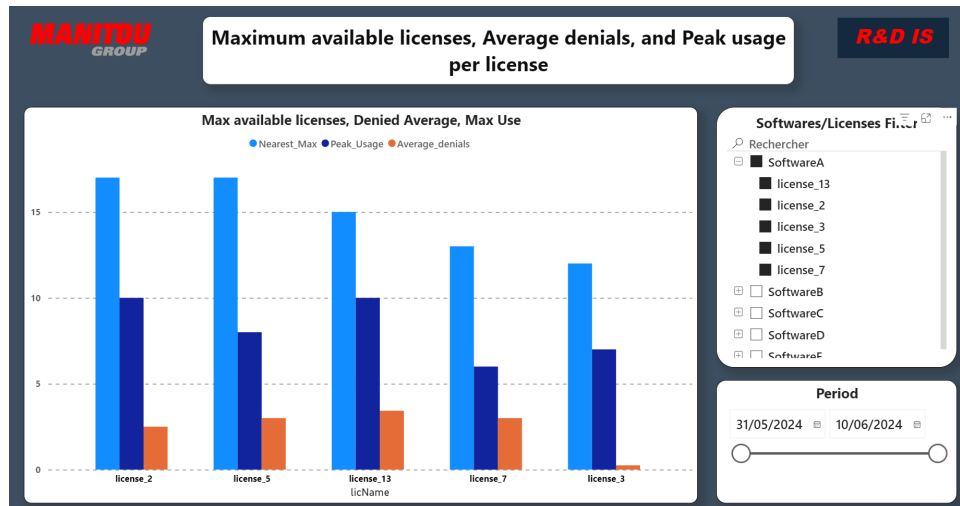


FIGURE F.4 – Page numéro 3

Content

Sur cette page, vous trouverez un graphique à barres listant des statistiques liées aux licences.

Filters

Deux filtres seront disponibles sur la page :

- Filtrer par logiciel puis par licences
- Filtrer pour sélectionner une période et visualiser les données entre deux dates

F.4 Vue par période (Page 4)

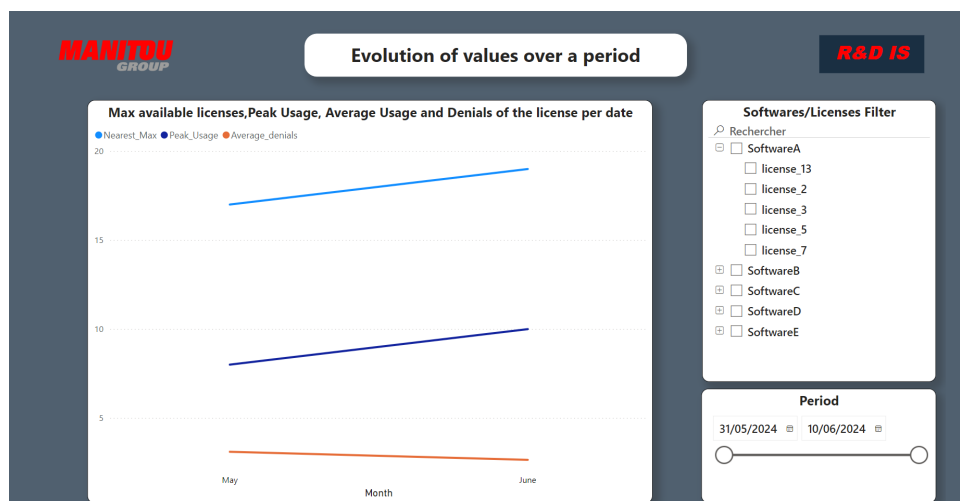


FIGURE F.5 – Page numéro 4

Content

Sur cette vue, il sera possible de visualiser les mêmes informations que sur la page précédente avec l'ajout de l'utilisation moyenne. La différence est que cette fois, ce sera via un graphique en ligne et la vue sera manipulable. Il est possible d'avoir une vue par Heure/Jour/Mois/Année pour la période définie par le filtre.

Les flèches en haut de l'image ci-dessous permettent de passer de l'heure au jour, du jour au mois, et ainsi de suite :

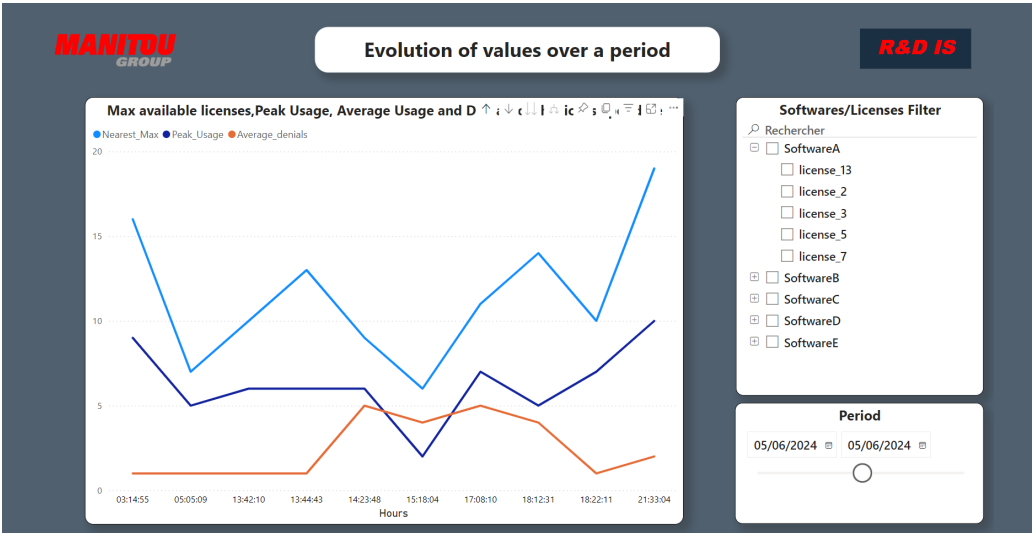


FIGURE F.6 – Vue par heures de la quatrième page

Filters

Les mêmes filtres que sur la page 3 seront disponibles.

F.5 Utilisateurs connectés (Page 5)

The dashboard shows a table titled 'Summary table of connected users for each license'. The table has the following columns: ExtractionDate, SoftwareName, licName, username, techRef, and ConnexionDate. The table contains 15 rows of data. To the right of the table is a 'Software/Licenses/Username Filter' section with a search bar and checkboxes for SoftwareA, SoftwareB, SoftwareC, SoftwareD, and SoftwareE.

ExtractionDate	SoftwareName	licName	username	techRef	ConnexionDate
01/06/2024 04:04:07	SoftwareA	license_13	user_25	tech_3	07/04/2024 04:04:07
01/06/2024 04:52:49	SoftwareA	license_13	user_5	tech_65	24/05/2024 04:52:49
01/06/2024 11:59:58	SoftwareA	license_2	user_37	tech_94	07/04/2024 11:59:58
02/06/2024 10:29:21	SoftwareA	license_2	user_36	tech_24	01/05/2024 10:29:21
04/06/2024 07:07:38	SoftwareA	license_3	user_20	tech_87	18/03/2024 07:07:38
05/06/2024 14:23:48	SoftwareA	license_13	user_11	tech_77	29/02/2024 14:23:48
05/06/2024 18:12:31	SoftwareA	license_13	user_30	tech_20	21/03/2024 18:12:31
07/06/2024 06:27:56	SoftwareA	license_13	user_8	tech_97	03/04/2024 06:27:56
07/06/2024 20:24:19	SoftwareA	license_13	user_43	tech_74	07/06/2024 20:24:19
08/06/2024 09:22:45	SoftwareA	license_2	user_32	tech_86	09/05/2024 09:22:45
08/06/2024 17:26:07	SoftwareA	license_3	user_29	tech_73	02/06/2024 17:26:07
09/06/2024 04:35:25	SoftwareA	license_2	user_34	tech_43	22/04/2024 04:35:25
09/06/2024 17:33:40	SoftwareA	license_2	user_37	tech_21	25/04/2024 17:33:40
09/06/2024 18:08:26	SoftwareA	license_2	user_18	tech_97	26/04/2024 18:08:26
09/06/2024 20:09:16	SoftwareA	license_3	user_21	tech_48	30/04/2024 20:09:16
10/06/2024 10:38:49	SoftwareA	license_13	user_17	tech_14	12/05/2024 10:38:49
10/06/2024 20:16:17	SoftwareA	license_3	user_4	tech_10	06/05/2024 20:16:17

FIGURE F.7 – Page numéro 5

Content

Sur cette page, vous trouverez tous les utilisateurs connectés dans les 15 dernières minutes ainsi que des informations les concernant. Il y aura un tableau contenant :

- Date d'extraction des données
- Nom du logiciel, de la licence et de l'utilisateur
- IP ou numéro de machine de l'utilisateur
- Date de connexion de l'utilisateur

Filtres

Il y aura un filtre permettant de filtrer par nom de logiciel, nom de licence, et nom d'utilisateur :

F.6 Tous les utilisateurs (Page 6)

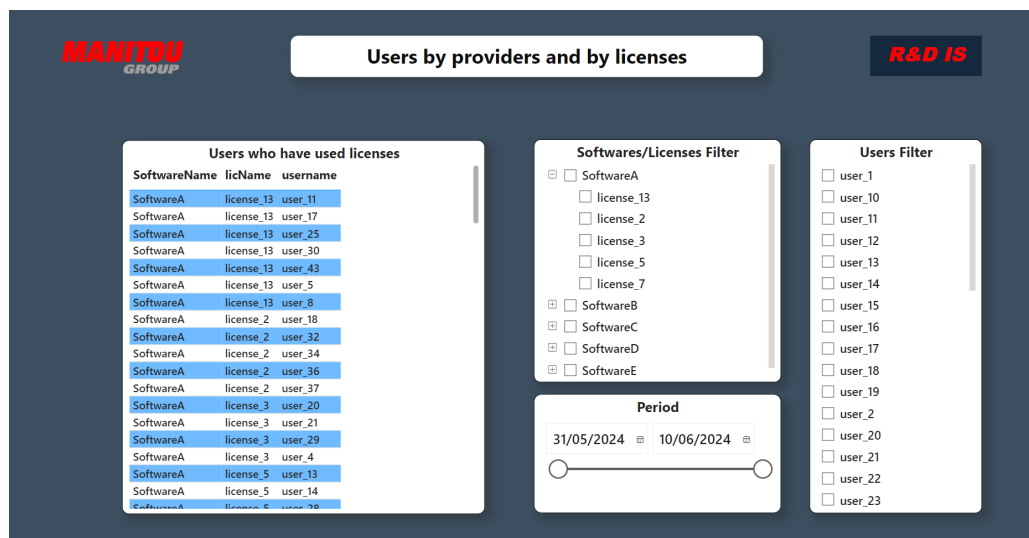


FIGURE F.8 – Page numéro 6

Content

Sur cette page, vous trouverez un tableau similaire à celui de la page précédente mais avec uniquement le nom de l'utilisateur et sans la date. Le but est d'avoir une vue de tous les utilisateurs ayant utilisé une licence sur une période définie.

Filters

Les filtres seront les mêmes que pour la page précédente mais avec la possibilité de modifier la période comme sur les pages 3 et 4.

F.7 Informations sur les fournisseurs (Page 7)

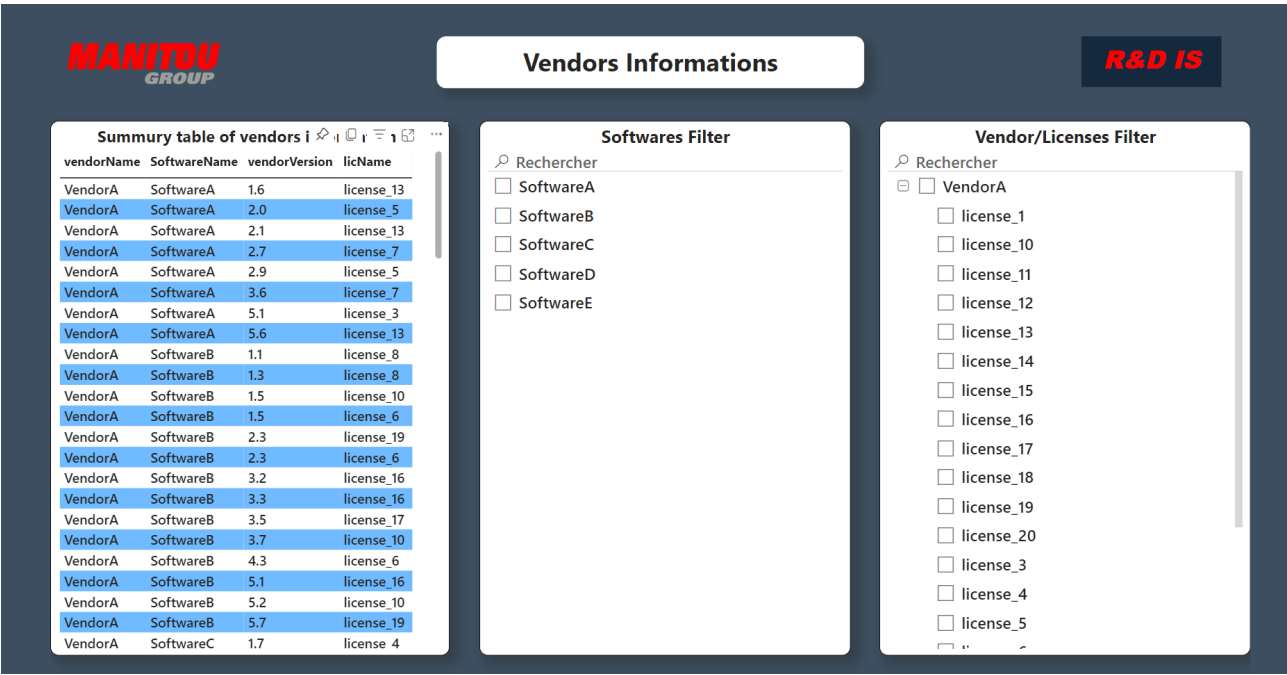


FIGURE F.9 – Page numéro 7

Content

La dernière page répertorie les informations sur les fournisseurs des outils de licence dans un tableau.

Filters

Il sera possible de filtrer par fournisseur, nom de licence, et nom de logiciel.

Annexe G

Déploiement

Pour mieux comprendre le fonctionnement des scripts de déploiement, voici quelques points clés à considérer :

Il est important de noter que l'objectif principal n'est pas simplement de copier tout ce qui se trouve sur l'environnement de développement. Il est essentiel d'effectuer une sélection rigoureuse des fichiers à déployer. En particulier, il ne faut pas inclure les tests ni les fichiers d'environnement spécifiques à chaque outil.

Les variables définies dans les fichiers d'environnement sont valides pour l'environnement de développement, mais ne le sont pas pour l'environnement de production. À la place, nous envoyons des modèles (templates) pour que l'équipe CoreOps sache quelles valeurs remplir. Cela permet d'adapter les configurations aux différents environnements de manière appropriée.

Ensuite, il est possible de déployer la solution sur plusieurs serveurs de production. Pour cela, il est nécessaire de pousser le fichier d'environnement global correspondant au serveur sur lequel on déploie la solution. Il est donc crucial que le nom du fichier d'environnement envoyé, afin de déployer la version appropriée en fonction du serveur.

Pour donner légèrement plus de détails, voici le code des deux scripts développés :

```
function PackagerApp {
    [CmdletBinding()]
    param (
        [Parameter(Mandatory)]
        #Nom de l'environnement sur lequel on veut envoyer les fichiers
        [string]$env,
        [Parameter(Mandatory)]
        #Nom de la version
        [string]$livName
    )

    #Récupération du fichier d'environnement
    $toolsEnv = @{};
    Get-ChildItem -Path "./env" -File -Filter "${env}.env" | ForEach-Object {
        $toolsEnv = ReadCompleteDotEnv -envFile $_.FullName;
    };

    #Déclaration de la sources des fichiers à copier ainsi que de la destination.
    $sourcePath = Get-Location
    $destinationPath = $toolsEnv['malt.global.livraison.dir']

    #Copie du dossier src ainsi que les templates des fichiers d'environnement
    CopyFilesToDestination -sourcePath "${sourcePath}/src" -destinationPath "${destinationPath}/${livName}/src"
    CopyFilesToDestination -sourcePath "${sourcePath}/conf/malt.*.template" -destinationPath "${destinationPath}/${livName}/conf"

    #Copie du fichier d'environnement, si ce dernier n'existe pas
    if (!(Test-Path -Path "${destinationPath}/${livName}/conf/malt.env")){
        CopyFilesToDestination -sourcePath "${sourcePath}/env/${env}.env" -destinationPath "${destinationPath}/${livName}/conf"
        Rename-Item -Path "${destinationPath}/${livName}/conf/${env}.env" -NewName "malt.env"
    }

    #Copie du script appliquer_livraison.ps1 permettant de copier la solution dans le dossier où ira chercher l'ordonnanceur.
    CopyFilesToDestination -sourcePath "${sourcePath}/appliquer_livraison.ps1" -destinationPath "${destinationPath}/${livName}"
}
```

FIGURE G.1 – packager.livraison.ps1

```

function AppliquerLivraison {
    param (
        [Parameter(Mandatory)]
        #Nom de la version
        [string]$livName
    )

    #Récupération des paramètres dans le fichier d'environnement
    $toolsEnv = @{};
    Get-ChildItem -Path "./conf" -File -Filter "malt.env" | ForEach-Object {
        $toolsEnv = ReadCompleteDotEnv -envFile $_.FullName;
    };

    #Déclaration du dossier où est la solution et du dossier où ira chercher l'ordonnanceur
    $sourcePath = $toolsEnv['malt.global.livraison.dir']
    $destinationPath = $toolsEnv['malt.global.deploy.dir']
    $completeSourcePath = "${sourcePath}/${livName}"

    # Suppression du fichier, s'il existe, indiquant quelle est la version actuellement utilisée
    if (Test-Path -Path "${destinationPath}/active_version.txt"){
        Remove-Item -Path "${destinationPath}/active_version.txt"
    }

    # Ajoute le dossier src
    CopyFilesToDestination -sourcePath "${completeSourcePath}/src" -destinationPath "${destinationPath}/src"

    #Ajoute du fichier indiquant quelle est la version actuellement utilisée
    New-Item -Path $destinationPath -Name "active_version.txt" -ItemType "file" -Value "${livName}"

    # Ajoute le fichier d'environnement dans le dossier conf
    if (!(Test-Path -Path "${destinationPath}/conf" -PathType Container)){
        New-Item -ItemType "directory" -Path "${destinationPath}/conf"
    }
    if (Test-Path -Path "${destinationPath}/conf/malt.env"){
        Remove-Item -Path "${destinationPath}/conf/malt.env"
    }
    CopyFilesToDestination -sourcePath "${completeSourcePath}/conf/malt.env" -destinationPath "${destinationPath}/conf" }

```

FIGURE G.2 – appliquer_livraison.ps1