



# Dossier Technique

## Projet Hub-Entreprendre

ROMATET Mathieu

CDA8 Session 2022.



Tuteur

LEOBON Anne

Formateur

MENDOUZE David



# Avertissement

La société Inetum demeure unique propriétaire des documents et données liés aux réalisations sur l'outil HUB Entreprendre. Aucun transfert des composants de l'outil tels que le code, les données ou la documentation technique ne peut être réalisé sans accord préalable écrit.

# Remerciements

Tout d'abord, je souhaite adresser mes remerciements à ma tutrice de stage, Anne LEOBON, pour sa disponibilité, son professionnalisme, la confiance qu'elle m'a témoignée et l'autonomie qu'elle m'a donnée dans la réalisation de ma mission.

Un grand merci à toute la Squad 3 d'Inetum en charge du Conseil Régional d'Occitanie pour leur accueil chaleureux et leur soutien, et notamment à Samuel GRANDBARBE, chef de projet coté Inetum, pour le temps passé à partager patiemment ses connaissances et son expérience.

Je voudrais ensuite adresser mes remerciements à toute l'équipe de l'AFPA de Balma pour son accompagnement au cours de cette année de formation, et notamment à Pascal DANGU, Lana POPLAVSKAYA et Philippe VIGUIER, formateurs, pour leur implication dans la transmission de leurs savoirs.

Merci également à tout le groupe de stagiaires CDA 8 de l'AFPA pour l'entraide sans faille, les encouragements et les moments de détentes partagés qui se prolongeront après la fin de notre formation.

Enfin, je tiens à remercier de tout mon cœur ma femme Sarah sur qui je peux toujours compter. Elle a su me soutenir dans les moments de doute et me porter, parfois littéralement, pour que je puisse atteindre l'objectif que je m'étais fixé.

# Abstract

Mon entreprise d'accueil, Inetum, est une ESN agile qui fournit des services et des solutions numériques, mais aussi un groupe mondial qui aide les entreprises et les institutions à tirer le meilleur parti du digital flow

Le client pour qui j'ai travaillé durant ma période d'application en entreprise est le conseil régional d'Occitanie.

L'application sur laquelle j'ai travaillé, Hub Entreprendre, permet de centraliser toutes les aides de la Région et de ses partenaires sur une seule plateforme (financements, appels à projets, ressources utiles, contacts) pour bénéficier d'un accompagnement 360° de son projet.

Son backend est développé en java avec le framework Spring (Spring boot, Spring data, Spring Security). Elle est reliée à une base de données Postgres et une base de données Elasticsearch.

Hub Entreprendre a 3 portails web développés avec le framework Angular :

- Le Front-Office qui est le portail d'accueil des entreprises pour s'inscrire, échanger et se renseigner
  - Le Front-Agent qui permet aux partenaires /agents d'accompagner les entreprises et les créateurs/repreneurs.
  - Le Back-Office qui donne la possibilité d'administrer les éléments du front entreprise et du front agent.
- 

My host company, Inetum, is an agile ESN that provides digital services and solutions, but also a global group that helps companies and institutions make the most of digital flow.

The client I worked for during my internship period was the regional council of "Occitanie".

The application I worked on, Hub Entreprendre, allows to centralize all the aids of the Region and its partners on a single platform (financing, calls for projects, useful resources, contacts) to benefit from a 360° support of its project.

His backend is developed in Java with the Spring framework (Spring boot, Spring data, Spring Security). It is connected to a Postgres database and an Elasticsearch database.

Hub Entreprendre has 3 web portals developed with Angular framework:

- The Front-Office which is the portal for companies to register, exchange and get information
- The Front-Agent which allows partners/agents to accompany companies and entrepreneurs.
- The Back-Office which gives the possibility to manage the elements of the front office and the front agent.

# Table des matières

<b>1. Introduction</b>	<b>7</b>
<b>2. Présentation de l'entreprise</b>	<b>8</b>
2.1. Historique et Présentation du groupe	8
2.2. Implantations en France	9
2.3. Présentation de l'agence	10
2.4. Organisation des services	10
2.5. Présentation de l'équipe SQUAD 3	11
<b>3. Présentation du Projet</b>	<b>12</b>
3.1. Les besoins et enjeux stratégiques pour le Conseil Régional d'Occitanie	12
3.2. Présentation de l'écosystème de la plateforme Entreprise :	12
3.3. Présentation du projet HUB ENTREPRENDRE	13
3.4. Cahier des charges	13
3.5. Planification du projet	15
A. Organisation	15
B. Processus de développement et de test	17
C. Le Suivi	18
D. Le Versioning	18
E. Echéances et répartition des tâches	19
3.6. Productions attendues	19
3.7. Environnement et Contraintes techniques	19
A. Environnement Technique	19
B. Contraintes du projet	22
<b>4. Missions annexes</b>	<b>22</b>
<b>5. Conception fonctionnelle du projet HUB</b>	<b>24</b>
5.1. Analyse du besoin	24
A. Contexte et Objectifs	24
B. Analyse de l'existant	25
C. Fonctionnalités de l'application	27

D.	Spécification des besoins fonctionnels	27
5.2.	Ma mission	30
5.3.	Besoins non fonctionnels	35
<b>6.</b>	<b>Conception technique du projet Hub</b>	<b>35</b>
6.1.	Présentation des différents composants	35
A.	Le Backend Hub Entreprendre	35
B.	Front Back-Office	38
C.	Fonctionnement général de Hub Entreprendre	41
D.	Plateforme technique	42
6.2.	Les évolutions	42
A.	Diagramme de classe	42
B.	Diagramme de séquence	44
<b>7.</b>	<b>Implémentation</b>	<b>45</b>
7.1.	PR1 : Modification url associée au bouton "contacter un expert de proximité"	45
7.2.	PR8 : Modification url du flux BPI	45
7.3.	PR7 : Extraction des dispositifs	46
A.	Le front Back-Office	46
B.	Le Backend	49
7.4.	PR5 : Extraction des Utilisateurs	54
A.	Le front Back-Office	54
B.	Le Backend	56
<b>8.</b>	<b>Sécurité</b>	<b>57</b>
8.1.	Prévention des XSS dans Angular	57
8.2.	Prévention des XSS avec SPRING	57
<b>9.</b>	<b>Déploiement</b>	<b>58</b>
<b>10.</b>	<b>Tests</b>	<b>60</b>
<b>11.</b>	<b>Bilan du projet</b>	<b>62</b>
<b>12.</b>	<b>Bilan personnel et perspectives</b>	<b>62</b>

<b>13. Annexes</b>	<b>63</b>
13.1. Tableau Kanban	63
13.2. Requêtes SQL pour résolution de ticket sur CARL	64
13.3. Dictionnaire des données	65
13.4. Modèle conceptuel des données - MCD	66
13.5. Modèle logique des données - MLD	67
13.6. Schéma d'architecture détaillé de Hub-Entreprendre	68
13.7. Fichier fr.json du Front-Entreprise	69
13.8. Requête de création de la Vue pour les Extraction des Dispositifs	69
13.9. IHM Extraction de usagers	70
13.10. fichier Jenkins pour déploiement sur plateforme azure	70

# 1. Introduction

Après 7 ans dans le domaine de l'IT, j'ai choisi de me reconvertis professionnellement dans le domaine du développement d'applications et j'ai ainsi suivi la formation de Concepteur Développeur d'Applications à l'AFPA de Balma (31) de septembre 2021 à août 2022.

Dans ce cadre, j'ai effectué une Période d'Application en Entreprise (PAE) au sein de la société Inetum à Toulouse dans l'équipe dédiée aux projets du Conseil Régional d'Occitanie, du 16 mai au 12 août 2022, pour valider les compétences acquises et les mettre en application.

Plus largement, cette PAE a été l'opportunité pour moi d'intégrer une équipe de professionnels qui m'ont permis, non seulement de mettre en pratique mes connaissances mais également d'obtenir de nouvelles compétences à travers le développement d'une application en production.

Tout au long de cette période, l'objectif général était l'élaboration de nouvelles fonctionnalités commandées par le Conseil Régional d'Occitanie. La mission a été extrêmement formatrice et m'a permis d'atteindre les objectifs suivants :

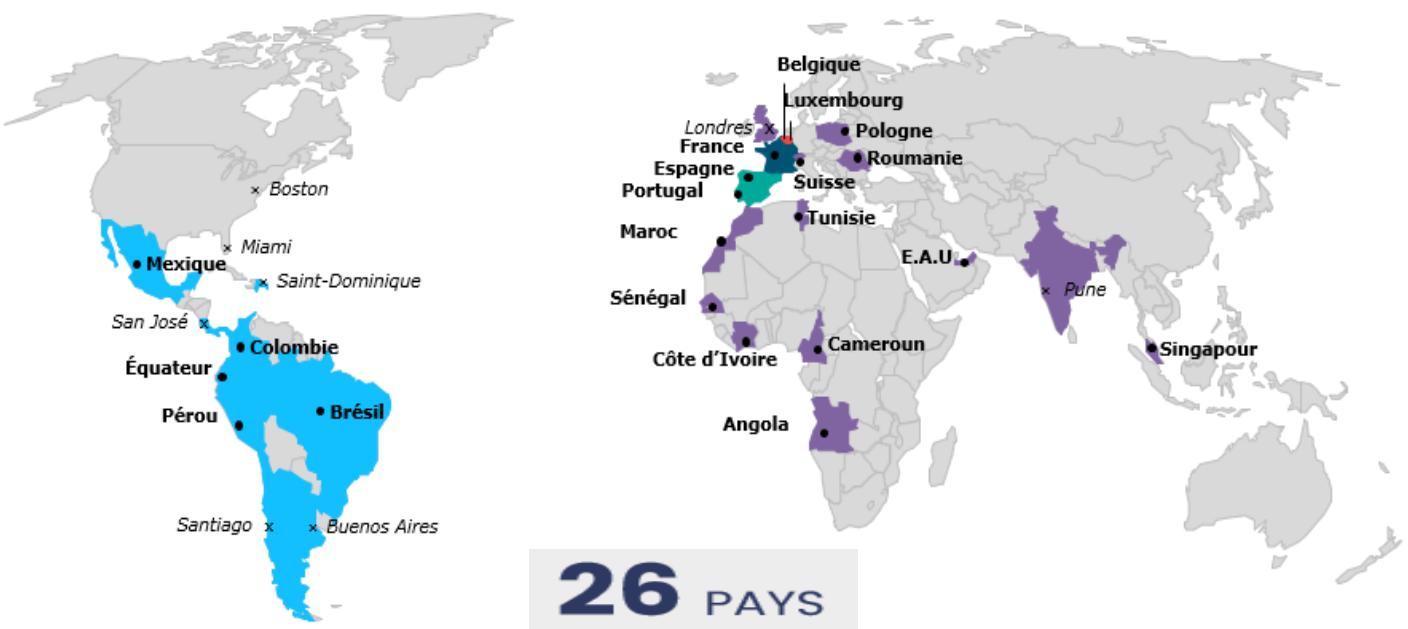
- ✓ Découvrir le cœur de métier de l'application et intégrer les différents environnements de développement.
- ✓ Enrichir ma connaissance dans la gestion de projet logiciel.
- ✓ Monter en compétence sur la partie fonctionnelle.
- ✓ Planifier et réaliser les tâches qui me sont affectées.
- ✓ Préparer et exécuter des tests.
- ✓ Préparer et exécuter le déploiement sur les serveurs d'intégration.

## 2. Présentation de l'entreprise

### 2.1. Historique et Présentation du groupe

Créée en France en 1970 sous le nom de « Groupe Français d'Informatique » (Gfi), Inetum est aujourd'hui une **entreprise de service numérique** (ESN) inscrite dans une dimension internationale.

La volonté d'Inetum - explicitée dans son slogan « Positive digital flow » - est de soutenir la transformation digitale des entreprises. Sa particularité est d'accompagner sur la durée et de façon évolutive et personnalisée, grâce à une maîtrise agile qui permet de s'adapter aux besoins de leurs clients.



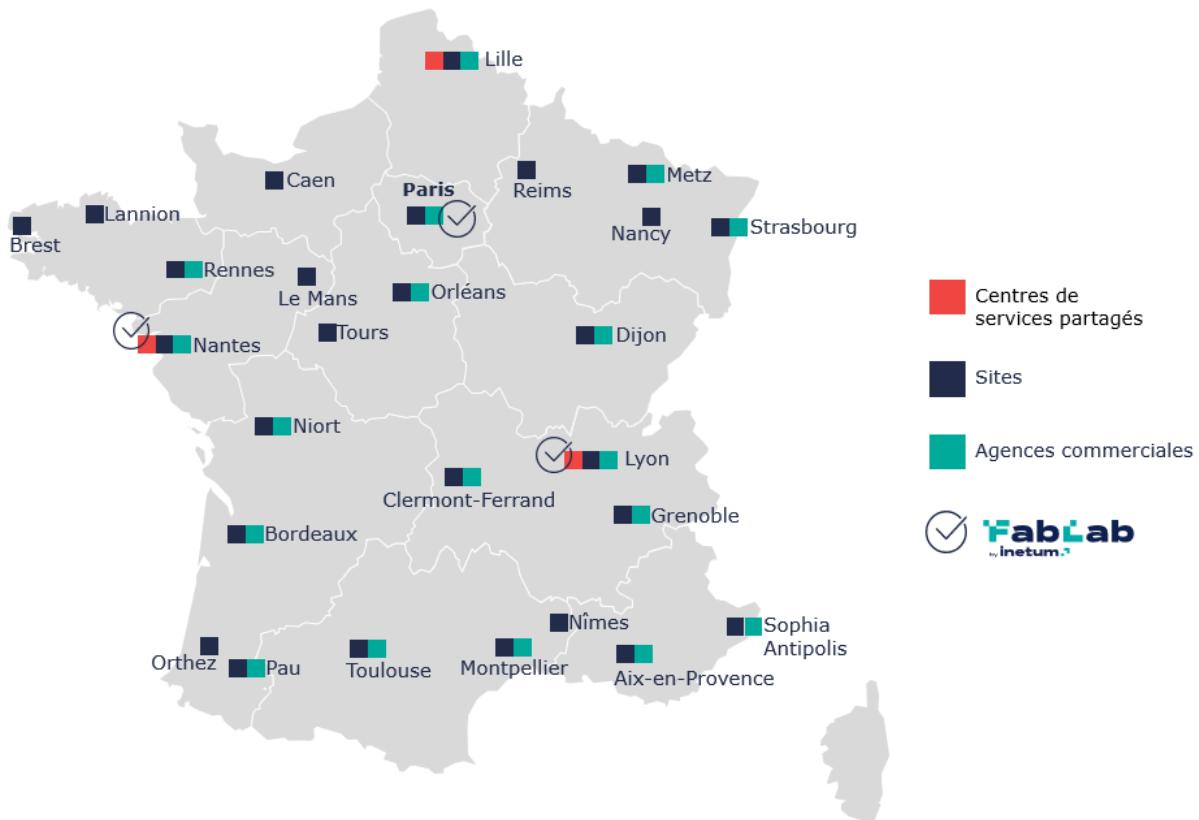
En 2020, la société choisit de se rebaptiser avec un nom plus moderne qui puise ses racines latines dans le mot « *incrementum* » signifiant croissance. Elle s'est ainsi donné une nouvelle image pour incarner son **ADN**, un nom qui revendique **l'esprit d'entreprenariat**, une **ambition** et un **esprit de conquête**.



## 2.2. Implantations en France

Avec une forte implantation sur le territoire français en raison de son histoire, la France génère **39% du chiffre d'affaires** avec un effectif de plus de **11.000 consultants**.

**45 sites** y compris  
**18 agences commerciales**



## 2.3. Présentation de l'agence

S'inscrivant dans cette dynamique de « Positive digital flow », la Branche Sud-Ouest constitue un pôle important sur le territoire. Avec 940 consultants, un chiffre d'affaires de 108 millions d'euros et cinq centres de compétences (ALM/PLM, SIG, MOBILITE, SOLUTIONS TELECOM et PERFORMANCE ET TRANSFORMATION), elle est présente dans le Sud-Ouest avec quatre sites : Toulouse, Bordeaux, Pau et Orthez.

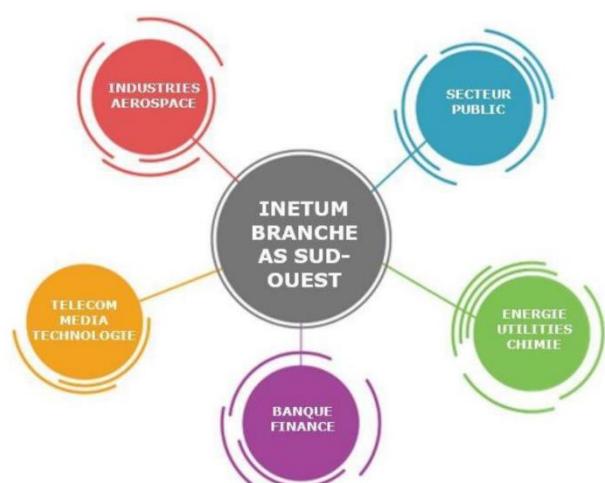
L'agence de Toulouse dans laquelle j'ai fait mon stage appartient à la Branche Sud-Ouest et est la plus importante.



## 2.4. Organisation des services

Inetum AS Sud-Ouest (AS pour Application Services) est organisée en plusieurs pôles représentant des secteurs d'activités diverses.

Ces pôles entretiennent des partenariats avec différents entreprises et acteurs publics tels que le Ministère de la Justice français, Tigo, Orange, Rolls Royce en Europe, Auchan, la Ville de Madrid, EDF, le Gouvernement du Mexique, etc.

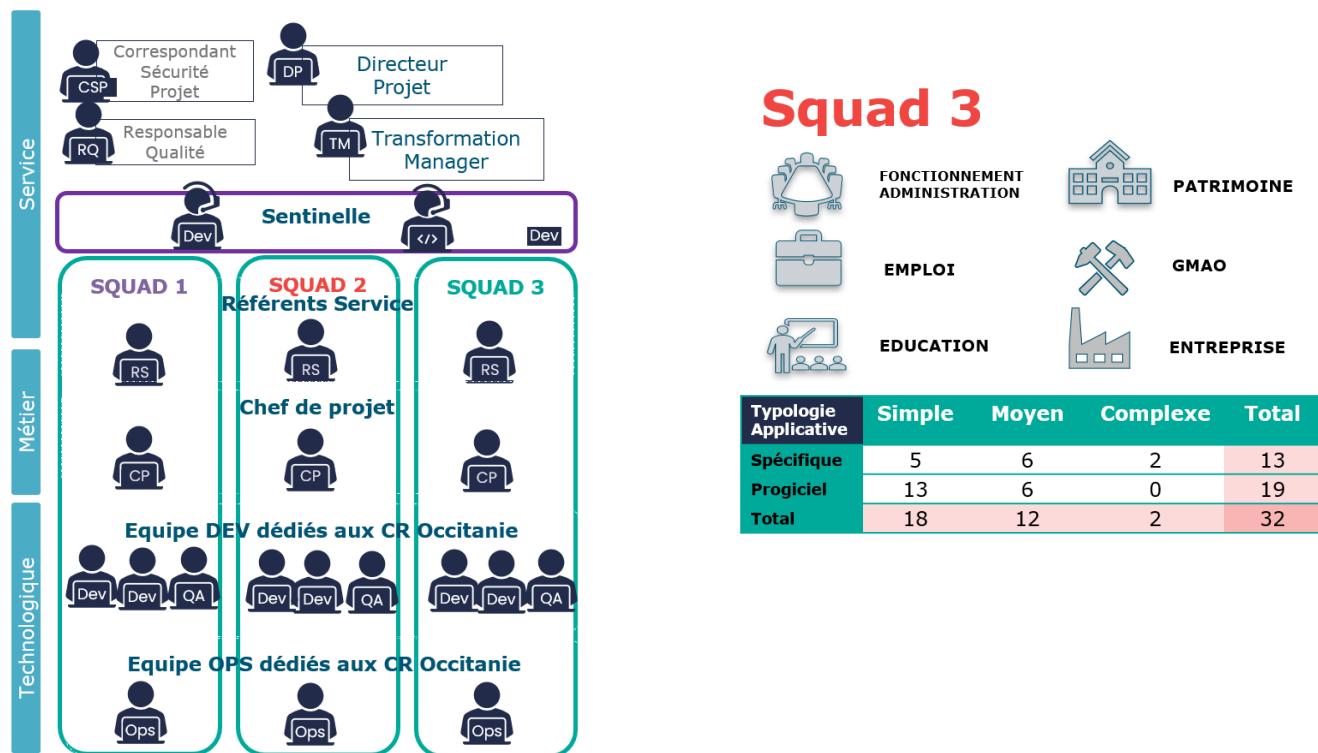


## 2.5. Présentation de l'équipe SQUAD 3

En 2021, Inetum a gagné le contrat du Conseil Régional d'Occitanie pour se charger de la Tiers Maintenance Applicative (**TMA**) et de la mise en conformité opérationnelle (**MCO**) pour 80 applications.

La TMA regroupe la conception de nouvelles applications ou la maintenance et l'amélioration de l'existant. La MCO consiste à s'assurer que l'outil est pleinement opérationnel à tout moment et en adéquation avec son environnement technique.

Les applications ont été réparties entre 3 Squad, j'ai intégré la troisième Squad positionnée à Toulouse et responsable de la TMA et la MCO de 32 applications.



Mon équipe est constituée actuellement de 8 développeurs, 1 testeur, 1 support, 1 référent service/technique et 1 chef de projet.

SQUAD 3	
Gouvernance	
Directrice de Projet	Anne LEOBON
Référents	
Référent Service	Fabrice NERON
Squad 3	
CP Squad #3	Samuel GRANDBARBE
Equipe SUPPORT	Serigne DIADHIOU
Equipe DEV	
Développeur	Anass MASAOUIDI
Développeur	Laurent NAVARRO
Développeur	Frederic DUARTE-GIL
Développeur	Sébastien FOURES
Développeur	Nicolas RICHOU
Développeur	Ahmed LASSOUED
Développeur	Yassine EL-GHONNAJI
Développeur	Mohammed LACHKAR
Testeur	Mickael DE CONINCK

### 3. Présentation du Projet

#### 3.1. Les besoins et enjeux stratégiques pour le Conseil Régional d'Occitanie

Le Conseil Régional a eu besoin de mettre en place un écosystème métier, applicatif et technique visant à répondre aux besoins des entreprises. Cette **plateforme Entreprise** permet :

- **Le développement économique** : Soutenir le développement économique de la Région, être au service de la politique régionale, rendre attractif la Région, offrir des services en ligne dédiés aux entrepreneurs.
- **La souveraineté à l'échelle de la Région** : Maîtrise de l'ensemble des technologies, gestion de la connaissance des logs, meilleure exploitation, suivi de l'audience et de l'usage.
- **L'animation de réseaux** : Permettre à la collectivité d'animer les acteurs économiques de la Région, proposer un parcours d'accompagnement structuré.
- **L'efficacité opérationnelle entre la région et Ad'OCC** (agence de développement économique de la Région Occitanie) : Pouvoir mesurer la performance interne.

#### 3.2. Présentation de l'écosystème de la plateforme Entreprise :

La « plateforme Entreprises » est un système composé de plusieurs applications rendant les services suivants :

- La gestion de la confiance : **Mon Compte Entreprise** (solution de gestion des identités) et l'**API Entreprise**, dont les enjeux sont la simplification du parcours usager avec un compte unique, une seule connexion pour tous les services, et la gouvernance des données d'identité des entreprises et porteurs de projet.
- Le portail d'information sur les subventions régionales, de structuration de l'accompagnement et de l'animation du réseau des développeurs économiques d'Occitanie : **le Hub Entreprendre Occitanie**.
- Le dépôt de demande d'aides, de suivi des instructions et paiements : **Mes Aides en Ligne**, dont les ambitions consistent à dématérialiser les dossiers de subvention, à suivre en ligne les dossiers et simplifier les échanges avec les déposants.
- L'organisation du travail d'accompagnement en vision transversale (prospection, communication, suivi de dossiers d'accompagnement, suivi des réalisations) : **le CRM OPERA**.
- L'exploitation des données issues du SI Entreprise (production et visualisation d'indicateurs d'activité en lien avec la dimension financière et budgétaire) : **l'Univers Entreprises**.

Gestion de la confiance	Identité numérique des entreprises et des porteurs de projet	 MON COMPTE ENTREPRISE
Portail de services	Portail d'information sur les subventions régionales, de structuration de l'accompagnement et de l'animation du réseau des dévelopeurs économiques	
Gestion d'interventions	Gestion d'interventions partagées AD'OCC – Région Occitanie : organisation du travail d'accompagnement en vision transversale – prospection, communication, suivi de dossiers d'accompagnement, suivi des réalisations, etc.	 Gestion Relation Entreprises
Dépôt de dossiers	Dépôt de demande d'aides et suivi des instructions, Simplification des échanges avec les déposants	 Mes aides en ligne
Gestion des paiements	Gestion des paiements des subventions et suivi	 Paiement
Suivi des métriques	Production et visualisation d'indicateurs d'activités en lien avec la dimension financière et budgétaire. <i>Exemple à venir : délai moyen d'instruction.</i>	 Métriques

3

### 3.3. Présentation du projet HUB ENTREPRENDRE

L'application Hub Entreprendre joue un rôle majeur dans le système qu'utilise la Région. C'est sur cette application que mes missions portaient pendant ma PAE.

Elle permet de centraliser toutes les aides de la Région et de ses partenaires sur une seule plateforme : financements, appels à projets, ressources utiles, contacts... pour bénéficier d'un accompagnement 360° de son projet.

En quelques clics, un entrepreneur peut facilement identifier les financements adaptés à son projet : création, croissance, innovation, export... et bénéficier de l'appui d'un référent professionnel de l'accompagnement et membre du Réseau des Développeurs Economiques Occitanie qui sera son contact privilégié.

Sur son espace personnalisé, l'entrepreneur peut également suivre sa demande de financement et consulter les événements et actualités adaptés à sa filière.

Le Hub Entreprendre Occitanie est le fruit d'un travail partenarial avec les membres du Réseau des Développeurs Economiques Occitanie. Ce réseau, animé par la Région Occitanie, est composé de 300 structures issues du territoire régional : Etablissements Publics de Coopération Intercommunale, Réseau des Chambres de Commerce et d'Industrie, Réseau des Chambres des Métiers et de l'Artisanat, Agence de développement économique Ad'OCC, réseaux spécialisés, etc.

Il a pour objectif de coordonner et renforcer l'action des professionnels de l'accompagnement pour plus de lisibilité et efficacité auprès des entrepreneurs d'Occitanie.

### 3.4. Cahier des charges

Le périmètre du projet sur lequel j'ai travaillé comportait cinq évolutions. Toutefois, la demande du client a évolué au cours de ma période en entreprise, en ajoutant et modifiant des évolutions. Voici la version finale du cahier des charges :

**Jira PR1 - [BO HUB] Modification url associée au clic-bouton "contacter un expert de proximité"**

**Compréhension du besoin** : Sur l'application Front Hub Entreprendre, il est demandé de faire une mise à jour de l'URL associée au lien "Contacter un expert de proximité".

**Tâches identifiées sur le développement :** Dans le code source du frontend Hub Entreprendre : modification d'un lien url dans un fichier de paramétrage de l'application.

#### Jira PR5 - [BO HUB] Extraction des usagers du Hub

**Compréhension du besoin :** Sur l'application Front Back-Office Hub Entreprendre, il est demandé d'ajouter une nouvelle carte (non cliquable) « Usagers » avec un bouton « extraire » permettant d'extraire en csv la liste des usagers du hub.

**Tâches identifiées sur le développement :**

- Dans le code source du frontend Back-Office :
  - Création d'un nouveau bouton raccordé à un service de création d'un fichier export ;
  - Création d'une nouvelle carte (affichage de données) sur l'accueil du Back-Office.
- Dans le code source du backend Hub Entreprendre :
  - Ajout de code pour construction de la structure des données à exporter en tenant compte de la complexité du fichier Excel, et des règles de transcodage à appliquer entre valeurs techniques vers des valeurs fonctionnelles.

#### Jira PR6 - [BO HUB] ajout des porteurs de projets dans la fiche référents

**Compréhension du besoin :** Sur l'application Front Back-Office Hub Entreprendre, il est demandé de modifier le tableau portefeuille de la fiche interlocuteur référent, afin d'y faire apparaître les « porteurs de projets » suivis par l'interlocuteur de la fiche concernée. L'ajout d'une fonction « affecter à un autre référent » sur la fiche interlocuteur doit aussi permettre de changer l'interlocuteur référent d'un porteur de projet.

**Tâches identifiées sur le développement :** Dans le code source du frontend Back-Office : intervention sur le composant web « page.interlocuteur » (plusieurs fichiers html, Typescript) pour compléter les données « Entreprises (Company) » par des données « Porteur de projet » dans le tableau portefeuille de l'interlocuteur. Il s'agit de modifier le code source existant pour gérer aussi les porteurs de projets sans impacter la gestion déjà réalisée pour remonter les entreprises (companies).

#### Jira PR7 - [BO HUB] Fonctionnalité d'extraction des dispositifs

**Compréhension du besoin :** Sur l'application Front Back-Office Hub Entreprendre, il est demandé dans le menu « Dispositifs » d'ajouter un picto cliquable permettant l'extraction de l'ensemble des dispositifs en format CSV.

**Tâches identifiées sur le développement :**

- Dans le code source du frontend Back-Office : Création d'un nouveau bouton raccordé à un service de création d'un fichier export.
- Dans le code source du backend Hub Entreprendre : Ajout de code pour construction de la structure des données à exporter en tenant compte de la complexité du fichier Excel, et des règles de transcodage à appliquer entre valeurs techniques vers des valeurs fonctionnelles.

#### Jira PR8 - [BO HUB] Flux BPI

**Compréhension du besoin :** Sur l'application Front agent Hub Entreprendre, il est demandé de faire une mise à jour de l'URL du flux BPI des dossiers projecteurs.

**Tâches identifiées sur le développement :** Dans le code source du backend Hub Entreprendre, modification d'un lien url dans un fichier de paramétrage de l'application.

## 3.5. Planification du projet

### A. Organisation

- **La Squad 3 :**

Dans la Squad 3, nous basculons progressivement sur les méthodes agiles, plus précisément sur la méthode SCRUM.

Une mêlée quotidienne (daily-scrum) est planifiée tous les matins qui permet aux développeurs de faire un point de coordination sur les tâches en cours et sur les difficultés rencontrées.

Ainsi, aucun membre de l'équipe ne reste bloqué plus de 24h sans qu'un autre l'accompagne dans l'exécution de sa tâche. L'objectif est de permettre à chaque membre de l'équipe de s'exprimer sur ses difficultés, ce qui permet aux membres de s'aider à trouver des solutions.

On utilise souvent les trois questions suggérées par Scrum pour structurer la réunion :

- Qu'ai-je fait hier qui rapproche le produit de l'objectif du sprint ?
- Que vais-je faire aujourd'hui qui rapproche le produit de l'objectif du sprint ?
- Ai-je une ou plusieurs difficultés qui m'empêchent d'avancer ?

Tout ceci doit rentrer dans une « timebox », un temps maximum imposé, afin que la cérémonie se finisse rapidement.

- **Le projet HUB ENTREPRENDRE :**

Après réception du cahier des charges, la première étape a été une session de chiffrage. Cet exercice m'a permis d'estimer le nombre de jours nécessaires au développement des évolutions. Après analyse des fonctionnalités, j'ai réfléchi au meilleur découpage des tâches et défini un nombre de jours pour la réalisation de chaque tâche.

PR1 :

Action	Chiffrage (jours)	Description
Développement/TU	0,25	Mise à jour d'une URL dans un fichier de paramétrage
Validation interne	0,25	Mise en Dev (Azure) INETUM et test
Rebond	0,25	Prise en compte correctif et retour suite à recette client
<b>TOTAL</b>	<b>0,75</b>	

PR5

Action	Chiffrage (jours)	Description
Développement/TU	1,5	Création du bouton servant à faire l'export et d'une nouvelle carte sur l'accueil du Back-Office
	2,5	Structuration de la donnée à exporter dont le besoin a été exprimé
Documentation	0,5	Documentation technique et fonctionnel
Validation interne	1	Mise en Dev (Azure) INETUM et test
Rebond	0,5	Prise en compte correctif et retour suite à recette client
<b>TOTAL</b>	<b>6</b>	

## PR6

Action	Chiffrage (jours)	Description
Développement/TU	3	Frontend : modification du component page interlocuteur pour compléter les données « Entreprises » par des données « Porteur de projet » dans le tableau portefeuille.
Documentation	0,5	Documentation technique et fonctionnel
Validation interne	1	Mise en Dev (Azure) INETUM et test
Rebond	0,5	Prise en compte correctif et retour suite à recette client
TOTAL	5	

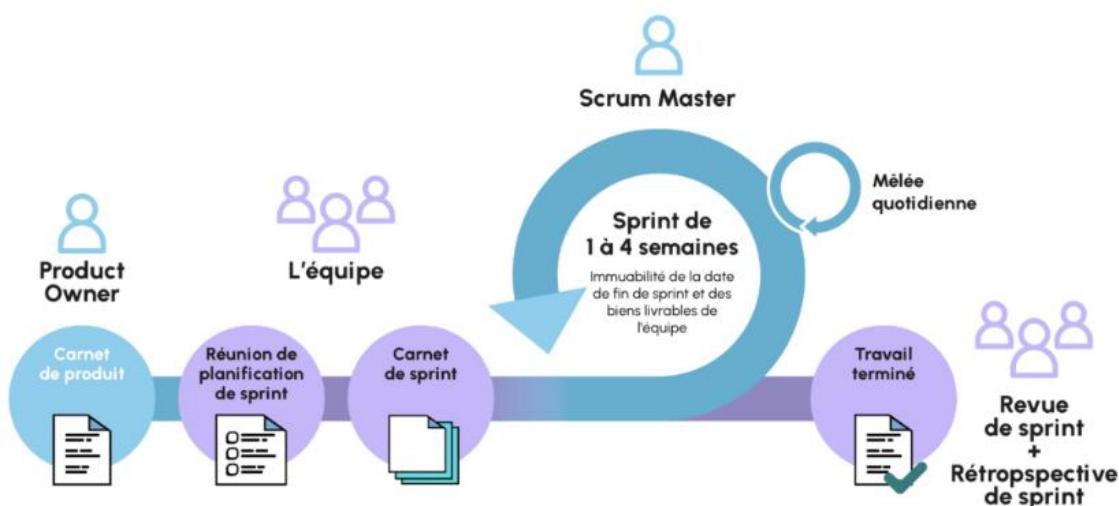
## PR7

Action	Chiffrage (jours)	Description
Développement/TU	1,5	Création du bouton servant à faire l'export dans la liste des dispositifs du Back-Office
	2,5	Structuration de la donnée à exporter dont le besoin a été exprimé
Documentation	0,5	Documentation technique et fonctionnel
Validation interne	1	Mise en Dev (Azure) INETUM et test
Rebond	0,5	Prise en compte correctif et retour suite à recette client
TOTAL	6	

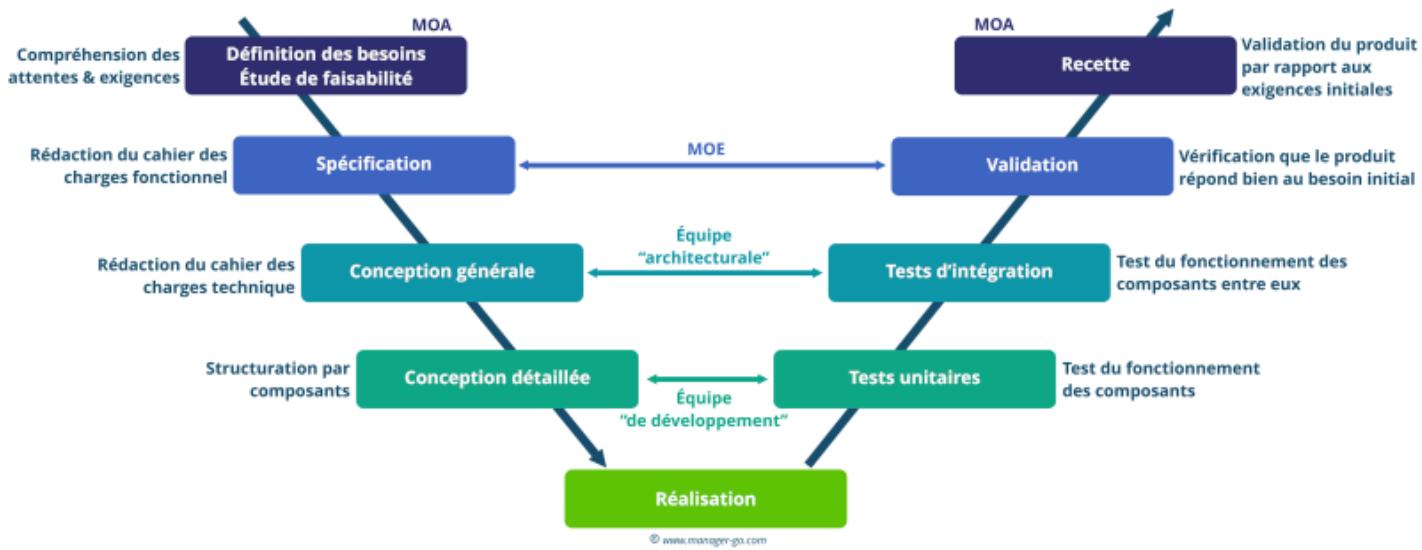
## PR8

Action	Chiffrage (jours)	Description
Développement/TU	0,5	Mise à jour d'une URL dans un fichier de paramétrage
Documentation	0,25	Documentation technique et fonctionnel
Validation interne	0,25	Mise en Dev (Azure) INETUM et test
Rebond	0,25	Prise en compte correctif et retour suite à recette client
TOTAL	1,25	

L'organisation du projet a initialement été proposée en suivant la méthode agile/itératrice dans le but d'appréhender les besoins des utilisateurs et du client est de permettre à ces derniers de **tester le produit au fur et à mesure**, en situation réelle.



Cependant, celle-ci demandant une plus forte implication du client dans le processus de développement accompagné de contraintes techniques, la gestion de projet s'est finalement déroulée en cycle en V. Voici une illustration de celui-ci :

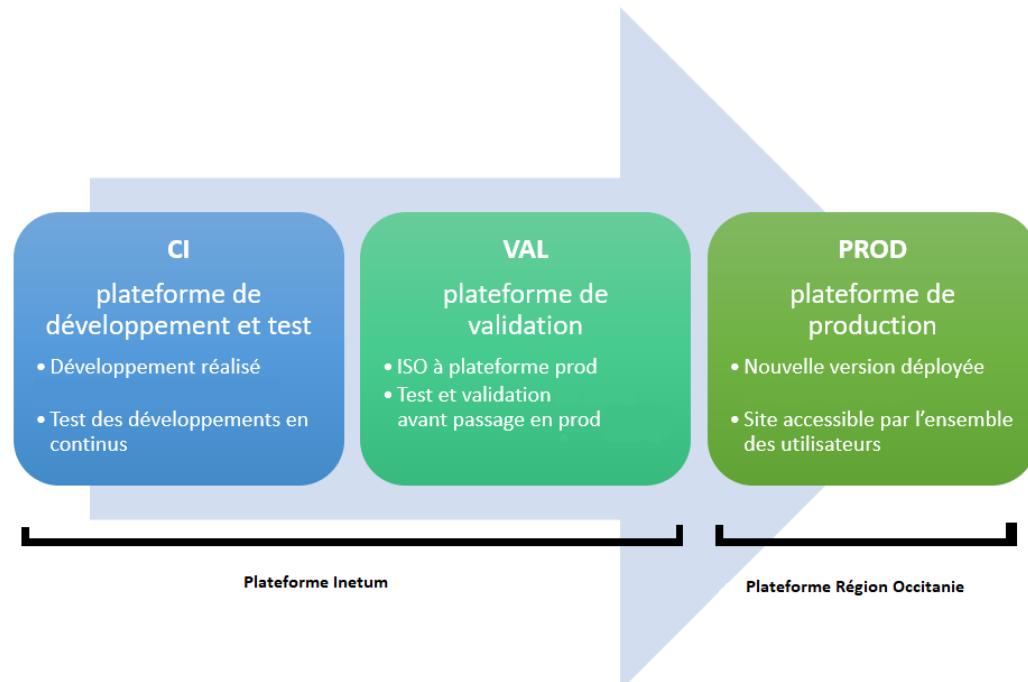


Le flux descendant permet de bien détailler le produit jusqu'à sa réalisation. Cela comprend l'expression des besoins, les spécifications fonctionnelles et les spécifications techniques. A l'issue de ce flux, on retrouve la réalisation qui sera ensuite suivie du flux montant.

C'est dans cette deuxième partie que tous les tests et la partie validation par le client seront réalisés.

## B. Processus de développement et de test

Le processus de développement et de test se déroule à travers 3 plateformes Microsoft Azure en ligne possédant chacune un rôle très précis comme on peut voir sur le schéma ci-dessous :



## C. Le Suivi

Le suivi avec le client a été effectué grâce à l'outil Jira Software. Dans Jira Software, le projet Kanban fournit un workflow prêt à l'emploi. Le responsable produit peut ajouter des tâches au « backlog » et les marquer comme « à analyser », « en cours de réalisation » ou « terminé » une fois la tâche ou la « user story » entièrement terminée.

Le client a accès au tableau Kanban et est toujours au courant de l'avancement du projet. Il pouvait aussi me contacter en laissant un commentaire.

Durant le processus de développement, j'ai pu sélectionner mes Jira et déplacer les éléments dans les colonnes « En cours de réalisation » ou « Terminé » (**annexe 1**). J'ai aussi contacté le client afin d'organiser une réunion avec le métier car j'avais besoin de précisions sur les besoins fonctionnels.

JIRA mère pour PR-1, 5, 6, 7 et 8 : lot pour recette commune

Joindre    Crée une sous-tâche    Associer un ticket    Couverture de Test

Généralités Qualification Ticket

Description  
PR-1, 5, 6, 7 et 8 : lot pour recette commune

Tickets associés +  
is tested by

- PR-12 TV-1 : Modification de l'url associée au clic-bouton "contacter un expert de proximité"    MC EN ATTENTE

se rapporte à

- PR-5 [BO HUB] Extraction des usagers du Hub    RM EN COURS DE RÉALISATION
- PR-8 [HUB] - mise à jour flux BPI "dossiers projecteurs"    RM EN COURS DE RÉALISATION
- PR-7 [BO HUB] Fonctionnalité d'extraction des dispositifs    RM À RÉALISER
- PR-6 [BO HUB] ajout des porteurs de projets dans la fiche référents    RM À RÉALISER
- PR-1 [HUB]Modification url associée au clic-bouton "contacter un expert de prox...    RM EN COURS DE RÉALISATION

A Réaliser

Détails

Responsable	DP
	Me l'assigner
Rapporteur	DP
Étiquettes	Aucun
Type Evolution	Evolution Planifiée
Date de livraison en qualification souhaitée	19 août 2022
Date de livraison en production souhaitée	29 août 2022
Composants	HUB ENTREPRENDRE

Plus de champs Estimation originale, Suivi temporel

Création 4 juillet 2022 à 08:47    Configurer  
Mis à jour il y a 4 heures

## D. Le Versioning

Pour le versioning, l'outil Git et la plateforme GitLab sont utilisés.

On peut voir sur le schéma à droite l'organisation des branches et des merges au cours du développement.

La branche développement correspond à la plateforme CI et la branche Master à la plateforme VAL.

A chaque merge un pipeline est lancé déployant la nouvelle version du projet sur la plateforme.



## E. Echéances et répartition des tâches

Après validation du chiffrage et proposition commerciale auprès du client, les dates de livraison prévues sont les suivantes :

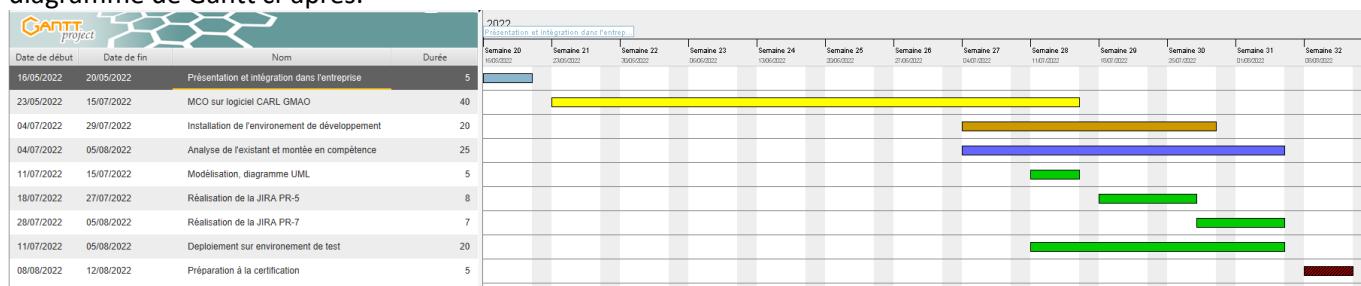
**Lancement** 11/07/2022

**Démonstration auprès du client** 16/08/2022

**Mise en recette** 19/08/2022

**Mise en production** 29/08/2022

Pour répondre à ces objectifs et atteindre le but fixé, j'ai planifié la gestion du projet Hub Entreprendre, et consigné l'organisation et la répartition des phases de développement et des tâches entreprises dans le diagramme de Gantt ci-après.



## 3.6. Productions attendues

Les productions attendues étaient initialement la livraison de la **nouvelle version de l'application** contenant les évolutifs demandés avec les **documents techniques associés** mais comme les demandes du client évoluaient et les contraintes techniques étaient importantes, l'objectif de la fin du stage a été fixé à la livraison d'une partie des évolutions.

## 3.7. Environnement et Contraintes techniques

### A. Environnement Technique

- **IntelliJ IDEA community :**

 Également appelé « IntelliJ », « IDEA » ou « IDJ », **IntelliJ IDEA community** est un environnement de développement intégré (en anglais Integrated Development Environment - IDE) destiné au développement de logiciels informatiques reposant sur la technologie Java. Il est développé par JetBrains (anciennement « IntelliJ ») et disponible en deux versions, l'une communautaire, open source, sous licence Apache 2 et l'autre propriétaire, protégée par une licence commerciale. Tous deux supportent les langages de programmation Java, Kotlin, Groovy et Scala.

- **Visual Studio Code :**

 **Visual Studio Code** est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré. Les utilisateurs peuvent modifier le thème, les raccourcis clavier, les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires.

- **Java :**



**Java** est un langage de programmation orienté objet qui reprend en grande partie la syntaxe du langage C++. Néanmoins, Java a été épuré des concepts les plus subtils du C++ et à la fois les plus déroutants, tels que les pointeurs et références, ou l'héritage multiple contourné par l'implémentation des interfaces. Les concepteurs ont privilégié l'approche orientée objet de sorte qu'en Java, tout est objet à l'exception des types primitifs (nombres entiers, nombres à virgule flottante, etc.) qui ont cependant leurs variantes qui héritent de l'objet Object (Integer, Float...).

- **Maven :**

**Maven™** Apache Maven (couramment appelé Maven) est un outil de gestion et d'automatisation de production des projets logiciels Java en général et Java EE en particulier. Il est utilisé pour automatiser l'intégration continue lors d'un développement de logiciel. L'objectif recherché est de produire un logiciel à partir de ses sources, en optimisant les tâches réalisées à cette fin et en garantissant le bon ordre de fabrication. Maven utilise un paradigme connu sous le nom de Project Object Model (POM) afin de décrire un projet logiciel, ses dépendances avec des modules externes et l'ordre à suivre pour sa production. Il est livré avec un grand nombre de tâches prédéfinies, comme la compilation de code Java ou encore sa modularisation.

- **Hibernate :**



**Hibernate** est un framework open source gérant la persistance des objets en base de données relationnelle. Hibernate est adaptable en termes d'architecture. Hibernate apporte une solution aux problèmes d'adaptation entre le paradigme objet et les SGBD en remplaçant les accès à la base de données par des appels à des méthodes objet de haut niveau.

- **Spring & Spring Boot :**



**Spring** est un framework open source pour construire et définir l'infrastructure d'une application Java, dont il facilite le développement et les tests. Spring s'appuie principalement sur l'intégration de trois concepts clés : L'inversion de contrôle, la programmation orientée aspect et une couche d'abstraction qui permet d'intégrer d'autres frameworks et bibliothèques avec une plus grande facilité.

**Spring Boot** facilite la création d'applications Spring autonomes de qualité production afin de les exécuter simplement. Il permet de simplifier le développement en Spring (configuration Maven, programme principal, etc.). Les éléments nécessaires pour une application standalone en SpringBoot sont un fichier pom.xml qui contient les informations nécessaires à la génération du projet (identification de l'artefact, déclaration des dépendances, définition d'informations relatives au projet, ...), un fichier Main.java qui initialise le conteneur Spring et des annotations dans les classes.

- **Angular :**



**Angular** est un framework côté client, open source, basé sur TypeScript, et codirigé par l'équipe du projet « Angular » à Google et par une communauté de particuliers et de sociétés. Il permet la création d'applications Web et plus particulièrement d'applications web single-page (SPA) : des applications web accessibles via une page web unique qui permet de fluidifier l'expérience utilisateur et d'éviter les chargements de pages à chaque nouvelle action.

- **Postgres :**



**PostgreSQL** est un système de gestion de base de données relationnelle orienté objet puissant et open source qui est capable de prendre en charge en toute sécurité les charges de travail de données les plus complexes.

- **DBeaver :**



**DBeaver** est un logiciel permettant l'administration et le requêtage de base de données.

Pour les bases de données relationnelles, il utilise un driver JDBC. Pour les autres bases de données (NoSQL), il utilise des pilotes de base de données propriétaire. Il fournit un éditeur qui prend en charge la complétion de code et coloration syntaxique.

- Jenkins :



**Jenkins** est un outil open source de serveur d'automatisation. Il aide à automatiser les parties du développement logiciel liées au build, aux tests et au déploiement, et facilite l'intégration continue et la livraison continue. Il s'interface avec des systèmes de gestion de versions tels que CVS, Git et Subversion, et exécute des projets basés sur Apache Ant et Apache Maven aussi bien que des scripts arbitraires en shell Unix ou batch Windows.

- Docker :



**Docker** est un outil qui peut empaqueter une application et ses dépendances dans un conteneur isolé, qui pourra être exécuté sur n'importe quel serveur. Il ne s'agit pas de virtualisation, mais de conteneurisation, une forme plus légère qui s'appuie sur certaines parties de la machine hôte pour son fonctionnement.

- Microsoft Azure



**Microsoft Azure** est une plateforme applicative qui comprend un service d'hébergement sur le cloud pour le stockage de données. Azure propose aussi des outils de développement pour la programmation d'applications et de machines virtuelles. Le système est compatible avec l'intégration dans d'autres logiciels grâce au principe d'émulation. Autrement dit, cela permet d'en disposer dans un environnement spécifique comme s'il était question d'un support physique.

- Lens



**Lens** est un client desktop pour gérer des clusters Kubernetes qui a été initié en 2020. Le projet est codé avec Electron. Il est maintenu par Mirantis, société qui traditionnellement faisait du openstack et qui maintenant a fait un virage vers Kubernetes. C'est aussi la société qui a racheté Docker Enterprise Edition.

- Gitlab :



Nous utilisons le logiciel Git avec la plateforme **Gitlab** pour la gestion de versions du projet. Son système de branche est particulièrement adapté dans le cadre d'un projet agile, il nous permet de toujours avoir une version déployable en production sur sa branche master, et nous permet de travailler sereinement nos fonctionnalités sur des branches annexes.

- Jira :



L'outil **Jira** est destiné aux sociétés désireuses de mettre en place un fonctionnement en méthode agile et facilite le travail des utilisateurs concernés dans leur organisation. Il permet notamment la création et la planification de tâches via un système de rédaction et de gestion des récits utilisateurs. Jira Software à destination des équipes de développeurs pour la mise en place de méthode Kanban et d'organisation de type scrum. Il permet le découpage des projets en récits utilisateurs (tâches, composants) et leur affectation aux développeurs.

- Microsoft Teams :



**Microsoft Teams** est une application de communication collaborative, officiellement lancée par son propriétaire Microsoft en novembre 2016. L'utilisation de cette application est faite par l'équipe pour l'ensemble des échanges et des réunions.

## B. Contraintes du projet

J'étais seul pour mettre en œuvre, sur un temps limité (4 semaines), ces évolutions pour une application complexe et sans un interlocuteur fonctionnel qui maîtrise le périmètre de l'application.

Cette application a été développée à l'origine par une autre entreprise et la période de réversibilité pendant le transfert des connaissances a été minime : la documentation de conception, d'installation et d'utilisation est incomplète, fausse ou inexistante et personne dans l'équipe n'avait encore développé sur l'application Hub Entreprendre ni même lancé en local.

L'essentiel des contraintes que j'ai rencontrées sont techniques et étaient dû à un problème d'installation d'environnement, de configuration pour le déploiement, de connexion avec les nombreuses API, d'expiration des certificats et de droits insuffisants sachant que l'application tournait déjà en production.

Nous n'avons jamais réussi à lancer l'application en local malgré l'intervention d'experts d'autres équipes d'Inetum et je développais donc en aveugle. La dernière semaine, j'ai pu compiler et déployer via Jenkins sur la plateforme azure interne à Inetum pour ensuite voir si les modifications que j'ai apportées étaient bonnes ou pas.

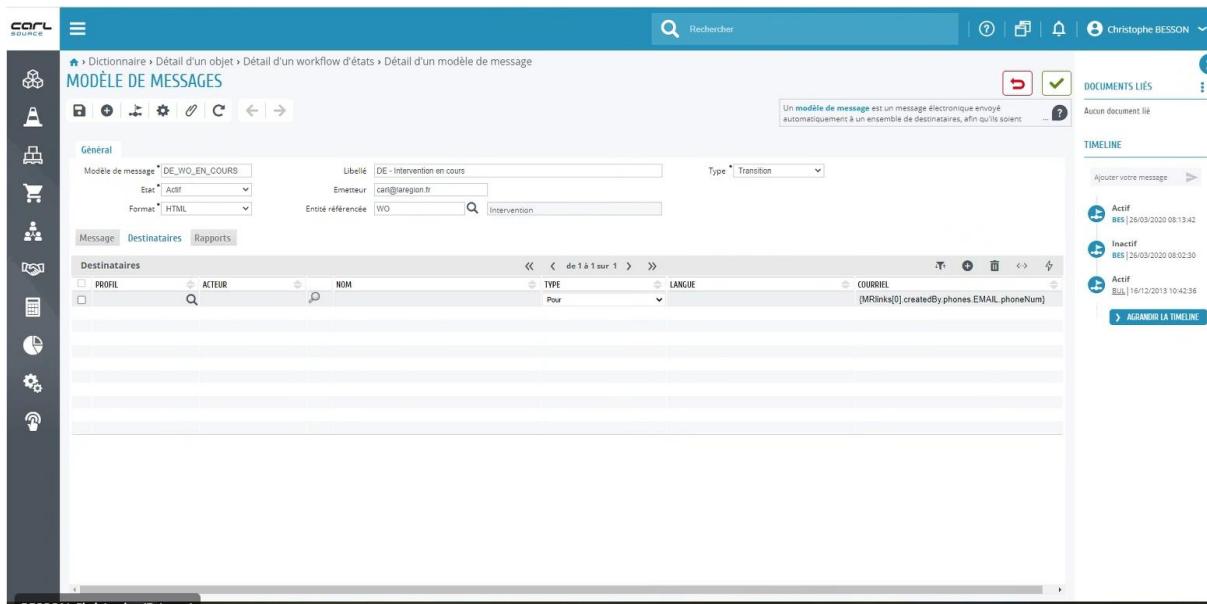
## 4. Missions annexes

Dès le début de mon stage, j'ai été positionné en tant que responsable du projet CARL Education et Port. J'étais en contact direct avec le responsable projet du côté de la région Occitanie et y ai consacré les 2 premiers mois de ma PAE.

Ce projet est géré grâce à **CARL Source** qui est un **GMAO** (Gestion de maintenance assistée par ordinateur). C'est un Logiciel destiné aux différents secteurs de l'industrie, du tertiaire, des institutions publiques, etc. Son intérêt est d'assister quotidiennement les services maintenance dans leurs missions, en adéquation avec les nouvelles technologies (applications de mobilité et de traçabilité).

CARL source a différents modules spécialisés dans des rôles pour faciliter la gestion d'entreprise :

- **Gestion des équipements** : inventaire, localisation, gestion d'information dédiée par type d'équipement.
- **Gestion de la maintenance** : corrective (avec BT : bon de travaux), préventive (systématique, conditionnelle, prévisionnelle), curative.
- Gestion des demandes d'intervention (DI).
- **Gestion des stocks** : magasins, réapprovisionnements, valorisation des stocks.
- **Gestion des achats** : demandes d'achats, commandes, achats de fournitures, pièces et prestations, facturation fournisseurs, etc.
- **Gestion du personnel et planning** : activités, métiers, planning de charge, prévisionnel, etc.
- **Gestion des coûts et budget** : main d'œuvre, stocks, achats, location de matériel, etc., préparation des budgets, suivi périodique, rapports d'écart, etc.
- **Application Mobile** : connectée ou en hors connexion pour gagner en productivité.
- **Indicateurs clés de performance** : tableau de bord (requêtes de base de données concernant des statistiques, des alertes, MTTR, MTBF, Pareto, etc.).



**Malheureusement, CARL Source est édité par « CARL software » qui est propriétaire du code source et je n'y avais donc pas accès.**

Ma mission consistait uniquement à répondre à des tickets incidents reçus sur **IWS** (outils de gestion de ticket qui permet de gérer les processus liés à l'IT Management et au ticketing/helpdesk) portant sur la gestion des lycées et des ports dans la région Occitanie. Ces tickets doivent être résolus dans le temps imparti suivant ses priorités (1 à 3 jours) pour ne pas payer des pénalités.

La solution à ces tickets concernait pour la plupart des configurations via L'IHM du logiciel grâce à un accès super Admin, à des échanges avec l'éditeur via des tickets d'incident que j'ouvrais chez eux, **mais aussi à des interventions sur la BDD de CARL via des requêtes SQL**.

J'ai dû apprendre le contexte métier et analyser le BDD pour enfin construire les requêtes SQL dont j'avais besoin. Voici une partie des requêtes Type que j'ai créé :

```

/*Modification de budget pour DA, LDA et écritures budgétaires*/
/*modification du budget pour DA*/
update cspu_pr set budgacc_id=(select id from csfi_budgacc where code = 'XXXX_SUB_PTM') where code = 'DA-XXXXXX';
/*modification du budget pour lignes de DA*/
update cspu_prlne set budgacc_id=(select id from csfi_budgacc where code = 'XXXX_SUB_PTM') where pr_id=(select id from cspu_pr where code = 'DA-XXXXXX');
/*modification du budget pour écritures budgétaires de LDA*/
update csfi_budgacmvt set budgacc_id=(select id from csfi_budgacc where code = 'XXXX_SUB_PTM') where prline_id=(select id from cspu_prlne where pr_id=(select id from cspu_pr where code = 'DA-XXXXXX')));
/*-----*/
/*Mise à jour status des DA suivant */
update cspu_pr set status_code='XXXXXX' where code in ('DA-XXXXXX');
/*Mise à jour status des CMD suivant */
update cspu_po set status_code='XXXXXX' where code in ('CMD-XXXXXX','CMD-XXXXXX','CMD-XXXXXX');

/*-----*/
/*ANNULATION DA*/
/*suppression de l'écriture budget de la LDA */
delete from csfi_budgacmvt where prline_id IN (select id from cspu_prlne where pr_id=(select id from cspu_pr where code = 'DA-XXXXXX') and linenumber = 'X');
/*suppression de l'écriture costcenter de la LDA*/
delete from csfi_costcentermt where prline_id IN (select id from cspu_prlne where pr_id=(select id from cspu_pr where code = 'DA-XXXXXX') and linenumber = 'X');
/*suppression deliveryLine de la LDA*/
delete from cspu_deliveryline where prline_id IN (select id from cspu_prlne where pr_id=(select id from cspu_pr where code = 'DA-XXXXXX') and linenumber = 'X');
/*suppression updatetrackDA de la LDA*/
delete from cspu_prlneupdatetrack where prline_id IN (select id from cspu_prlne where pr_id=(select id from cspu_pr where code = 'DA-XXXXXX') and linenumber = 'X');
/*suppression ligne d'achat*/
delete from cspu_prlne where pr_id =(select id from cspu_pr where code = 'DA-XXXXXX') and linenumber = 'X';

/*Modification statut de Demande d'achat*/
update cspu_pr set status_code='XXXXXX' where code = 'DA-XXXXXX';

/*Suppression article sur intervention*/
delete from cswo_expwoitem where wo_id=(select id from cswo_wo where code='XXXXXX') and referitem='XXXXXXXXXX';

/*-----*/
/*MODIFICATION MONTANT DA */
/*Modification de l'écriture budget de la LDA en TTC */
update csfi_budgacmvt set committed='XX' where id='XXXXXXXXXXXXXX'; /*Modif du mouv budgétaire d'engagé lié à la validation de la DA*/
update csfi_budgacmvt set committed='-XX' where id='XXXXXXXXXXXXXX'; /*Modif du mouv budgétaire d'engagé lié à la réception de la DA*/
update csfi_budgacmvt set realised='XX' where id='XXXXXXXXXXXXXX'; /*Modif du mouv budgétaire de réalisé lié à la réception de la DA*/

/*Modification du prix HT des lignes de DA /!\ "grossunitpr" en HT*/
update cspu_prlne set grossunitpr= 'XX' where pr_id=(select id from cspu_pr where code = 'DA-XXXXXX'); /*Modification du prix des lignes de DA existantes*/
update cspu_prlne set grossunitpr= 'XX' where pr_id=(select id from cspu_pr where code = 'DA-XXXXXX') and linenumber = 'X'; /*Modification du prix d'une ligne de DA particulière*/
/*-----*/
/*MODIFICATION PERIODE IMPUTATION*/
/*modification de la période d'imputation budget de la DA */
update csfi_budgacmvt set mvtdate='XXXXXX' where prline_id IN (select id from cspu_prlne where pr_id=(select id from cspu_pr where code = 'DA-XXXXXX')));

/*Modification de la période d'imputation lié à l'id de l'écriture*/
update csfi_budgacmvt set mvtdate='XXXXXX' where id='XXXXXXXXXXXXXX';

```

En annexe se trouve un jeu de requêtes pour la résolution de tickets (**Annexe 2**).

## 5. Conception fonctionnelle du projet HUB

La conception fonctionnelle permet de bien définir l'application du point de vue de l'utilisateur. Le langage graphique de **modélisation UML** a été utilisé pour décrire les spécifications fonctionnelles du projet. Les fonctionnalités de l'application sont représentées par des **cas d'utilisation**.

### 5.1. Analyse du besoin

L'analyse du besoin permet d'identifier les **attentes des utilisateurs** afin de décrire le **processus métier** dans lequel l'application informatique à développer devra intervenir, les tâches prises en charge par cette application, son interaction avec les différents intervenants (utilisateurs et éventuellement avec les autres systèmes informatiques).

Pour cela, j'ai organisé un échange sur team-speak avec une représentante du métier côté Région Occitanie pour qu'elle puisse m'expliquer en détail les processus métier et les attentes sur les améliorations que je dois implémenter.

#### A. Contexte et Objectifs

Le Hub s'adresse aux créateurs, aux repreneurs et aux chefs d'entreprises qui souhaitent se développer ou s'implanter en Région Occitanie.

Le Réseau des Développeurs Economiques Occitanie est un réseau intégré territorialisé piloté par la Région et composé de plus de 400 structures et 1.600 membres. Son objectif est de mailler le territoire pour offrir un niveau d'accompagnement de qualité le plus harmonieux et le plus lisible possible.

Chaque membre dispose d'un espace partenaire dans le Hub qui lui permet :

- D'obtenir des informations sur les aides et appels à projets des financeurs suivants : Région Occitanie, Bpifrance, Créalia, France Initiative, Irdi-Soridec, etc.
- De consulter les actualités de la Région Occitanie et les évènements économiques sur le territoire.
- De bénéficier d'un espace personnalisé pour accéder :
  - À un référent – un contact qualifié de proximité qui l'accompagnera sur tous ses projets.
  - À un parcours qui lui permettra d'identifier le(s) financement(s) adapté(s) à son projet.
  - À des actualités et évènements relatifs à sa ou ses filières.
  - Au dépôt en ligne des demandes d'aide.
  - Au suivi de l'état d'avancement des dossiers déposés.

Les référents accèdent, en complément, aux informations sur les entreprises de leur portefeuille : projets déposés par l'entreprise, dispositifs Région associés aux projets, historique des contacts, etc.

Pourquoi choisir un référent ?

C'est au moment de la création de son compte que le choix du référent est fait par l'entreprise. Ce référent est un membre du Réseau des Développeurs Economiques et son rôle consiste à être un interlocuteur privilégié de l'entreprise. Il est en charge de créer le lien avec les autres interlocuteurs spécialistes dans le but de simplifier les démarches des entrepreneurs pour un suivi plus efficace et concerté.

## B. Analyse de l'existant

- **Les acteurs :**

Dans cette partie, je décris les différents acteurs du système qui entrent en jeu pour la compréhension des tâches que j'ai pu réaliser.

**Simple interlocuteur** : Collaborateur sans rôle spécifique auprès des entreprises mais dont les coordonnées seront uniquement visibles par les membres du Réseau des Développeurs Economiques dans l'extranet Agents Partenaires (p.ex. un manager).

**Référent** : Au moment de la création de son compte, une entreprise choisit un référent. Ce référent est un membre du Réseau des Développeurs Economiques Occitanie. Son rôle consiste à être un interlocuteur privilégié de l'entreprise ou du porteur de projet. Il est en charge de créer le lien avec les autres interlocuteurs spécialistes dans le but de simplifier les démarches de l'entreprise ou du porteur de projet pour un suivi plus efficace et concerté. Le référent est choisi par l'entreprise ou le porteur de projet lors de sa première connexion au Hub Entreprendre.

**Expert d'un financement** : Un expert de financement est un agent qui maîtrise l'ensemble des règles d'interventions d'un dispositif d'aide aux entreprises. Un expert est désigné par dispositif. Il est le point d'entrée des demandes de renseignement des référents sur les critères d'éligibilité requis et sur les modalités de dépôt d'un dossier sur ce dispositif. L'expert pourra, le cas échéant, réorienter la demande vers un spécialiste filière ou territoire. Les coordonnées de l'expert de financement sont uniquement accessibles aux membres du Réseau des Développeurs Economiques Occitanie via l'extranet agents et partenaires.

**Spécialiste** : Un spécialiste a une filière particulière d'action ou une compétence, qu'il va pouvoir mettre à disposition des référents pour accompagner certains projets.

**Groupe projet** : Pour chaque entreprise ou projet, le référent peut s'entourer de compétences spécifiques et complémentaires en constituant un groupe projet.

- **Les partenaires et dépendances**

**Les gestionnaires d'identités :**

La solution repose sur deux gestionnaires d'identités :

La **GDI Entreprise** pour les mandataires et les ayants droits des entreprises ainsi que les porteurs de projet.

La **GDI Partenaire** pour les accès au Back-Office et au Front Agent des agents de la Région Occitanie et des partenaires assurant le suivi des démarches (Ad'OCC, etc.).

**Les API Externes :**

L'**API Entreprise** permet de récupérer les informations administratives des entreprises. Les données servies par l'API sont issues directement des organismes de références.

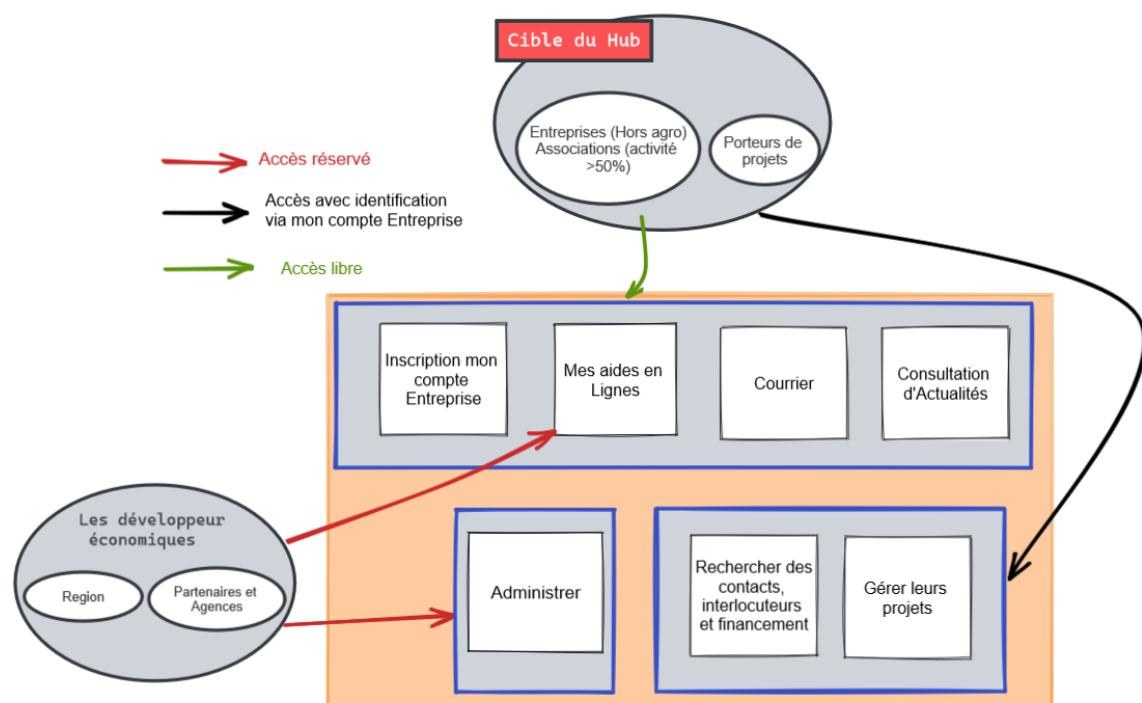
**Progos** est une application de la Région sur laquelle les dossiers de subventions sont récupérées.

**SPIP** est le service sur lequel les actualités de la Région sont récupérées.

**Grand Angle (GDA)** permet de récupérer les paiements faits par la Région et versées aux entreprises.

**BPI France** est une api qui permet d'afficher dans le Front Agent les "dossiers projeteurs" des entreprises.

- **Fonctionnement général :**



## C. Fonctionnalités de l'application

Les utilisateurs du hub-entreprendre peuvent se connecter sur trois portails web avec des fonctionnalités différentes :

**Le Front-entreprise** où les **entreprises** (mandataires, collaborateurs, prestataires externes) peuvent :

- Consulter les Actualités et les boîtes à outils
- Rechercher des financements et des évènements
- Choisir un référent
- Créer un projet
- Suivre des dossiers

Accessible à cette adresse : <https://hubentreprendre.laregion.fr>

**Le front Agent** où **Agents région**, Ad'OCC, et les partenaires Région (CMA, CCI) peuvent :

- Rechercher des entreprises (contact), des financements et des interlocuteurs,
- Recevoir des notifications
- Ajouter un acteur dans un groupe projet, un événement d'une structure
- Consultation l'historique, le portefeuille d'entreprise et créateurs-repreneurs, l'annuaire, le dossier projecteurs (BPI)

Accessible à cette adresse : <https://agent.hubentreprendre.laregion.fr>

**Le Back-office** où les **agents région (rôle Admin)** peuvent gérer :

- Des structures d'accompagnement
- Des dispositifs (financements)
- Des interlocuteurs (référents, spécialistes)
- Des outils
- Des territoires
- Des filières
- Des actualités
- Des encart extranet (Front agent)
- Des brèves d'information
- Des événements

Accessible ici : <https://backoffice.hubentreprendre.laregion.fr>

Après découverte de l'application Hub Entreprendre et son cœur de métier et parce qu'il n'y avait pas de documents de conception, j'ai commencé en effectuant un **reverse-engineering** de l'application et de la base de données en m'aidant du schéma architectural, des portails web en ligne en production et des échanges avec le métier en me concentrant sur le Backend et le Back-Office, sujet de ma mission.

## D. Spécification des besoins fonctionnels

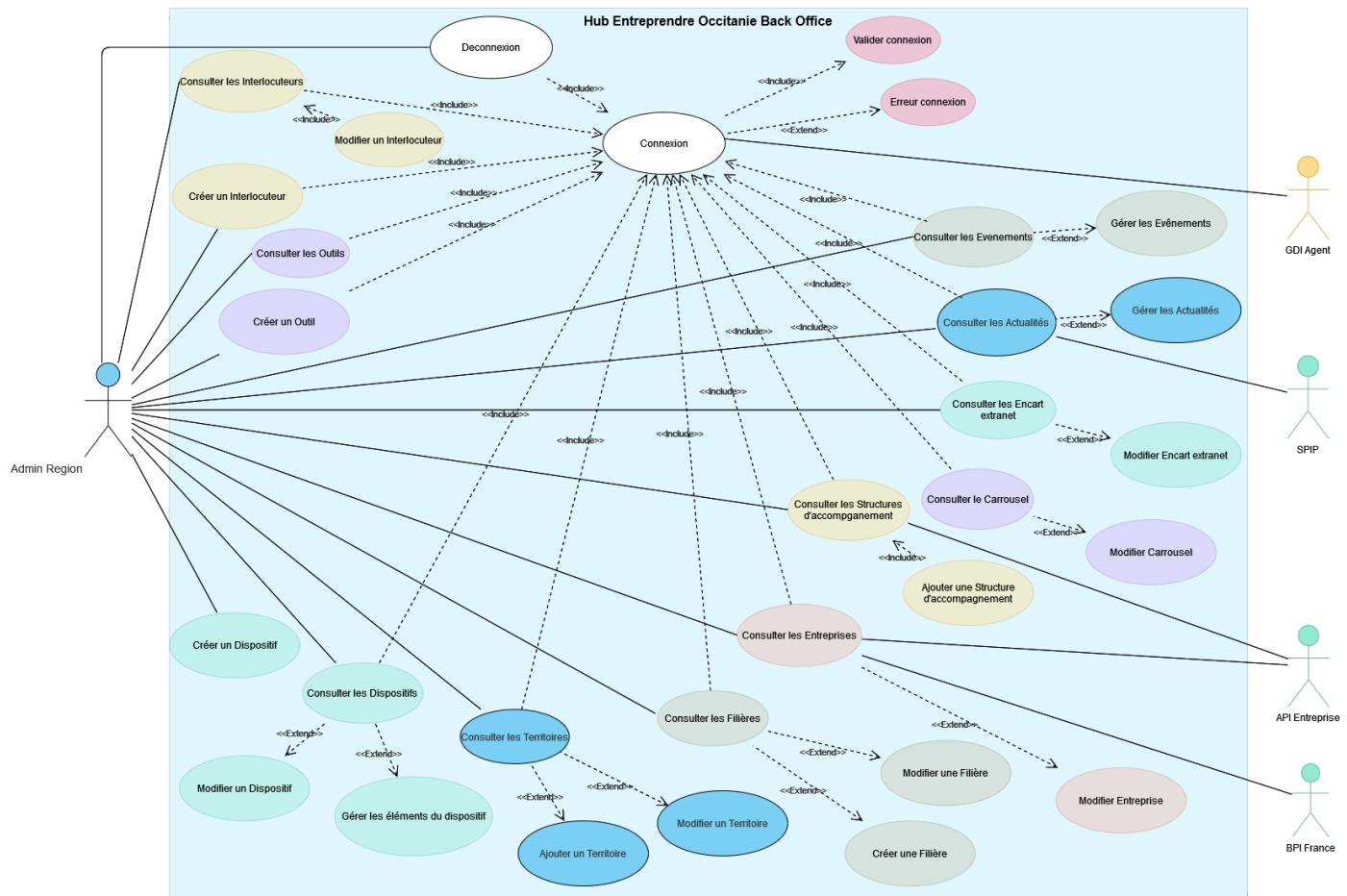
- **Les cas d'utilisation (Use Case) :**

Les rôles des diagrammes de cas d'utilisation sont de recueillir, d'analyser et d'organiser les besoins, ainsi que de recenser les grandes fonctionnalités d'un système. Il s'agit donc de la première étape UML pour la conception d'un système.

Un diagramme de cas d'utilisation capture le comportement d'un système, d'un sous-système, d'une classe ou d'un composant tel qu'un utilisateur extérieur le voit. Il scinde la fonctionnalité du système en unités cohérentes, les cas d'utilisation, ayant un sens pour les acteurs.

Ce travail m'a permis de comprendre et de voir toutes les tâches associées à un Agent Administrateur Région et ses interactions avec les systèmes Back-Office.

Le diagramme ci-dessous représente la modélisation UML des **cas d'utilisation** de l'application pour les Administrateurs Région sur le portail Back-Office.



- **Spécification des données**

Ensuite, j'ai entamé la compréhension de la base de données PostgreSQL de Hub Entreprendre à partir d'un dump de la BDD disponible sur le GitLab interne à Inetum et j'ai modélisé les données avec la réalisation du :

- **Dictionnaire de donnée (Annexe 3).**
- **Modèle conceptuel des données : MCD (Annexe 4).**
- **Modèle logique des données : MLD (Annexe 5).**

Dans la méthodologie Merise, le dictionnaire des données sert à faire la liste de toutes les données à digitaliser, le MCD décrit les données utilisées par le système d'information et leurs relations entre elles. Le MLD représente la structure de la base de données.

## Dossier technique de l'application Hub-Entreprendre | ROMATET Mathieu

J'ai dû demander à une équipe dédiée de me fournir un dump de la BDD en production car il manquait des fichiers de scripts de création de la BDD dans notre Gitlab.

Ces fichiers ont été d'une grande aide et m'ont permis de localiser les tables sur lesquelles j'interviendrais pour réaliser les évolutions demandées par le client.

### • Les IHM

Voici en exemple 3 IHM du Back-Office sur lesquels je vais porter des modifications :

#### La page d'accueil du Back-Office après connexion :

#### La page des dispositifs du Back-Office :

## La page des interlocuteurs du Back-Office :

SIRET	Nom de l'entreprise	Mandataire(s)
5049397[REDACTED]	[REDACTED]	Cynthia [REDACTED] Audrey [REDACTED]
797937[REDACTED]	[REDACTED]	
520709[REDACTED]	[REDACTED]	
7538580[REDACTED]	[REDACTED]	Laurent, andre, germain [REDACTED]
3815320[REDACTED]	[REDACTED]	Benoit [REDACTED]
35303499[REDACTED]	[REDACTED]	
8229420[REDACTED]	[REDACTED]	
52761031[REDACTED]	[REDACTED]	Cedric pierre [REDACTED]
4802890[REDACTED]	[REDACTED]	Anne-catherine benoite [REDACTED] Anne-catherine benoite [REDACTED] Florian, maxime, guy [REDACTED] Florian, maxime, guy [REDACTED]

## 5.2. Ma mission

- Spécification fonctionnelles détaillées :**

Pour les évolutions que j'avais en charge, j'ai écrit les **Spécifications Fonctionnelles Détailées** (« SFD » ou « DFS » en anglais) qui désignent l'ensemble de documents décrivant en détail le comportement des fonctionnalités et sous-fonctions de la solution cible, en conformité avec les besoins du Client collectés en amont. Ce sont des documents à fournir à la livraison des évolutions.

J'ai aussi conçu **des maquettes** quand c'était nécessaire pour les présenter au client et les faire valider. Puis j'ai spécifié les formats des fichiers de sorties.

### PR1 : Modification url associée au clic-bouton "contacter un expert de proximité"

**En tant qu'utilisateur je veux avoir accès aux coordonnées des contacts prioritaires afin d'être mis en relation avec les bons interlocuteurs sur le Front-Entreprise :**

	<p>La zone de contact (L3845,01, H2528, X-5,01, Y0) est composée :</p> <p>D'un titre "VOUS AVEZ UNE QUESTION ?" : Arial, Bold 24, #1B1B1B</p> <p>D'un sous-titre "Différentes options s'offrent à vous" : Arial, Médium, 30, #04797E</p> <p>D'un bloc contact (L1725,19, H229, X87,5, Y2195) composés de 4 contacts chacun composé :</p> <ul style="list-style-type: none"> <li>- D'un logo ou titre : ArialBold, 24, #1B1B1B, Taille du composant : L389; H54; X543,19; Y2143</li> <li>- D'une description : Arial, , Medium, 18, #1B1B1B (certains mot en bold), Taille du composant : L389; H83; X1433,19; Y2333</li> <li>- D'une zone de contact : Téléphone ( L266,67; H43; X53402; Y2381) ou d'un lien</li> <li>- image Téléphone</li> <li>- Numéro de téléphone : Arial, Bold, 24, #04797E</li> <li>- Texte : "Service gratuit + prix d'un appel" : Arial, Bold, 14, #3B3B3B</li> <li>- Si lien : bouton #04797E, L224; H42; X1439; Y2325 / Libellé : Arial, Bold, 17, #FFFFFF, L174; H21; X1469; Y2333</li> </ul>
--	---

Maquette :**L'agence Ad'Occ en Occitanie**

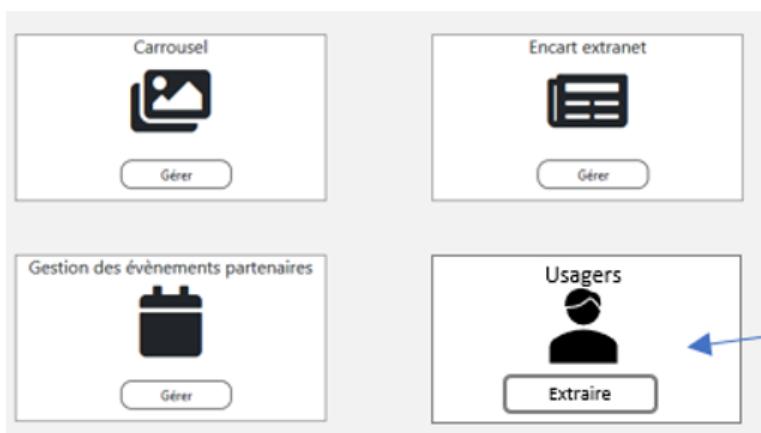
L'agence Ad'Occ accompagne toutes les entreprises en Occitanie. Son objectif : être à vos côtés pour vous soutenir et développer votre activité.

[Contacter un expert de proximité](#)

**PR5 : Extraction des usagers du Hub**

**En tant qu'administrateur du BO je peux visualiser la page d'accueil du Hub Entreprendre afin de voir toutes les données que je peux modifier :**

	<p>La page d'accueil est composée de 12 tuiles réparties sur 3 colonnes et composées :</p> <ul style="list-style-type: none"> <li>- D'un cadre à fond blanc et à bord gris (#797979)</li> <li>- Un titre centré en haut du cadre blanc</li> <li>- Une image centrée</li> <li>- Un bouton "Créer" en bas à gauche du cadre blanc (pour dispositif, interlocuteur et outils uniquement)</li> <li>- Un bouton "Gérer" en bas au milieu du cadre blanc (pour Les outils, Carrousel, Encart extranet, Actualités, Gestion des événements partenaires)</li> <li>- <b>Un bouton Extraire en bas au milieu du cadre blanc pour Usagers</b></li> <li>- Un bouton lien "Voir tout &gt;&gt;" en bas à droite du cadre blanc (sauf Carrousel, Encart extranet, Actualités, Gestion des événements partenaires et usagers).</li> <li>- Chaque carte est cliquable et reliée à sa page dédiée correspondante (sauf Usagers)</li> </ul> <p>Les 12 tuiles sont :</p> <ul style="list-style-type: none"> <li>- Les dispositifs</li> <li>- Les interlocuteurs</li> <li>- Les territoires</li> <li>- Les structures</li> <li>- Les entreprises</li> <li>- Les filières</li> <li>- Les outils</li> <li>- Carrousel</li> <li>- Encarts extranets</li> <li>- Actualités</li> <li>- Gestion des événements partenaires</li> <li>- <b>Usagers (La carte est non cliquable car non reliée à une page dédiée dans le site. Seul le bouton est cliquable)</b></li> </ul> <p>Le picto user en haut à droite devient "User connecté" (en rouge)</p>
--	--

Maquette :

Spécification du format du fichier d'extraction :

Fichier en format csv rassemblant la liste des usagers du Hub avec les champs suivants :

Nom, Prénom, Statut (mandataire, collaborateur, externe), Date de consentement, Coche téléphone (oui/non), Coche mail (oui/non), Mail, Tel, SIRET (vide si projet), Raison sociale/Nom projet, Code INSEE, Localisation (entreprise), Code NAF, Référent obligatoire (oui/non).

**PR6 : Ajout des porteurs de projets dans la fiche référents et l'ajout d'une fonction « affecter à un autre référent » sur la fiche interlocuteur qui doit permettre de changer l'interlocuteur référent.**

**En tant qu'administrateur du BO, je peux visualiser le détail d'une carte Interlocuteur afin de vérifier les informations :**

<p>Pour visualiser le détail d'une carte Interlocuteur, l'administrateur du BO clique sur le rectangle gris "Prénom NOM" de l'interlocuteur.</p>	<p><b>COMPOSITION DE L'ECRAN</b>  En haut à gauche "&lt; Retour à la liste" permet à l'administrateur de revenir sur la page précédente  Un picto "user"</p> <p>Ancrée à droite, alignée avec le haut de la photo un cadre gris (#D7D7D7) contenant  Prénom et NOM sur la gauche du cadre  Référent ou Expert Spécialiste ou Référent / Expert /Spécialiste ou Référent/Spécialiste... sur la droite selon le cas  Sur le cadre gris, ancré à droite en gras : "Charte qualité approuvée :" suivie de "Oui" ou "Non"  Sous le cadre gris, ancrés à gauche  "Poste :"  "Téléphone :"  "Email :"  "Domaine(s) de compétence(s) :"  "Domaine de compétences :"  "Filière :"  Secteur d'activité (uniquement pour les Référent)  Sous le cadre gris, ancré à droite  Le logo de la structure à laquelle l'interlocuteur est rattaché  Le nom de la structure à laquelle l'interlocuteur est rattaché  L'adresse de la structure à laquelle l'interlocuteur est rattaché (n°, rue, code postal ville)  Un onglet "Son portefeuille" souligné d'un trait rouge (#BB0000) affiché uniquement si l'interlocuteur est déclaré à minima "Référent"  Un onglet "Ses dispositifs" affiché uniquement si l'interlocuteur est déclaré à minima "Expert"  Un bouton "Modifier" en Jaune (#FFFF00)</p>
	<p>L'onglet "Son portefeuille" est composé de :</p> <ul style="list-style-type: none"> <li>- Une zone de saisie pour rechercher une entreprise ou un porteur de son portefeuille. La recherche est faite sur le SIRET, le nom de l'entreprise, le nom du porteur</li> <li>- Un bouton "Loupe" pour exécuter la recherche</li> <li>- <b>Un tableau résultat de 4 colonnes : "SIRET (si Entreprise)", "Nom de l'entreprise / du porteur de projet", "Mandataire", " "</b></li> </ul> <p>Par défaut le tableau est trié par ordre alphabétique du Nom de l'entreprise/du porteur de projet. Une fonctionnalité de tri est proposée sur les colonnes "Siret (si entreprise)" et "Mandataires"</p> <p><b>Pour chaque ligne, dans la 4ème colonne (qui n'a pas de libellé), un lien pour accéder à la fonctionnalité "Affecter à un autre référent"</b></p>

Maquette :

Son portefeuille (3)

SIRET (si entreprise)	Nom de l'entreprise / du porteur de projet	Mandataire(s)	
90034659400014	Entreprise 1	Jean Dupond	<a href="#">Affecter à un autre référent</a>
	Sabine LECLAIR	Said KADI	<a href="#">Affecter à un autre référent</a>
88869185400028	Entreprise 2		<a href="#">Affecter à un autre référent</a>

**PR7 : Extraction des dispositifs**

**En tant qu'administrateur du BO, je peux visualiser tous les dispositifs saisis sur la plateforme afin d'avoir une vue d'ensemble :**

Lorsque l'administrateur clique sur dispositifs du menu de gauche (ou sur Voir tout de la vignette) il voit tous les dispositifs déjà saisis dans le BO.

## COMPOSITION DE L'ECRAN

- D'un titre centré en haut de l'écran "Les dispositifs"
- D'une zone de saisie contenant le libellé "Rechercher" (écrit en gris et italique)
- D'un bouton picto "loupe" permettant d'exécuter la recherche
- D'un bouton picto "Exporter les données" permettant l'export des dispositifs**
- D'un bouton picto "créer" pour créer un nouveau dispositif
- De trois onglets
- "Tous"+ le nombre de dispositifs écrits entre parenthèses
- "Actifs"+ le nombre de dispositifs actifs écrits entre parenthèses
- "Inactifs""+ le nombre de dispositifs inactifs écrits entre parenthèses

Maquette :

Tous (164)    Actifs (105)    Inactifs (59)

Spécification du format du fichier d'extraction :

Fichier en format csv rassemblant la liste des dispositifs du Hub avec les champs suivants :

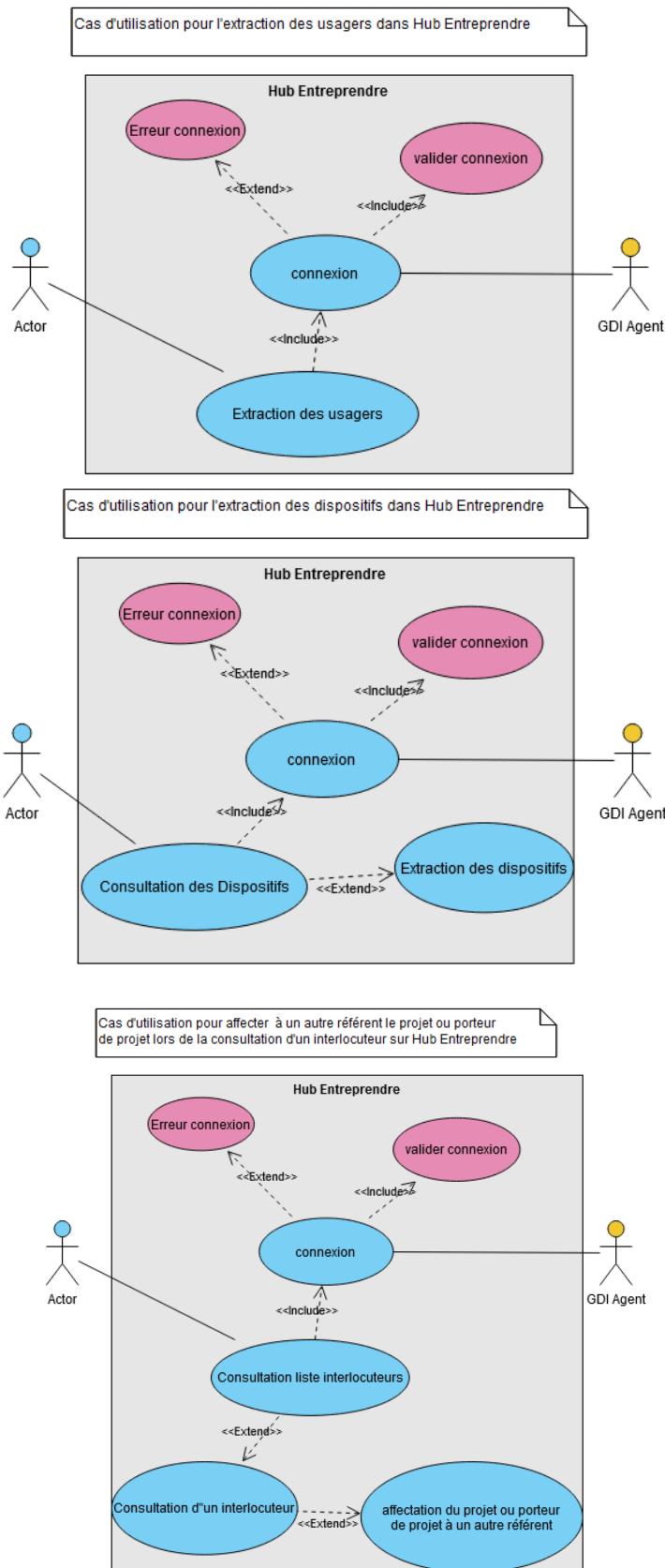
Id, Nom, Courriel, Catégorie, Phrase d'accroche, Chiffres clés, Actif, Type, Filières, Catégories du projet, Dépenses éligibles, Objectifs, Nature aide, Tags, NAF exclus, Date de création, Date Maj, Date de départ, Date de fin, Operateur, Experts, Lien fiche descriptive (pdf) (nom du document + nom pour l'affichage s'il y en a un spécifique), Documents, Documents visibles uniquement sur le portail Agents (Lien vers la FAQ, PDF), Liens de dépôt de dossier PDF (+ nom du lien s'il existe dans la colonne suivante, séparés par des slashes), Liens de dépôt de dossier URL (URL + nom du lien s'il existe, séparés par des slashes), Vidéos (nom de la vidéo : URL de la vidéo, séparés par des slashes), Age min, Age max, Nombre d'employés min, Nombre d'employés max, Lien URL de la page.

**PR8 : Modification url du flux BPI**

Je n'ai pas fait de conception fonctionnelle pour cette évolution car elle n'impliquait que de modifier une URL dans un **fichier .yml** dans le code source du Backend.

- Cas d'usage :**

J'ai aussi produit les cas d'usage (use cases) pour les améliorations demandées : PR5 – PR6 – PR7 :



## 5.3. Besoins non fonctionnels

Quand les besoins fonctionnels expriment les fonctionnalités concrètes du produit, les besoins non fonctionnels sont des indicateurs de qualité de l'exécution des besoins fonctionnels.

Ils sont importants et ne doivent pas être négligés, pour cela il faut répondre aux exigences suivantes :

- **Conformité du système à un environnement** : normes réglementaires, documentaires, conformité aux licences acquises, etc.
- **Maintenabilité du système** : traçage des erreurs, possibilité des mises à jour, extensibilité / modifacibilité du système initial, supportabilité en fonction de l'implantation géographique du futur système, testabilité, etc.
- **Performance du système** : charge utilisateurs / transactions.
- **Portabilité du système** : compatibilité avec diverses plateformes, facilité de remplacement d'autres systèmes en place, facilité d'installation et de désinstallation de l'application, etc.
- **Fiabilité du système** : capacité à gérer les erreurs du système, densité des défauts de qualité, capacité à être remis en état rapidement, capacité à résister aux attaques, etc.
- **Sécurité du système** : traçage des mises à jour des données dans le système, gestion de la confidentialité, gestion de l'intégrité des données, protection des données personnelles, etc.
- **Utilisation du système** : facilité d'utilisation en limitant le nombre de clic à maximum 3 clics pour finaliser la transaction, rendre l'application attractive, certification du système à une technologie particulière, capacité à respecter les exigences d'un pays, réutilisabilité de certains composants, etc.

# 6. Conception technique du projet Hub

## 6.1. Présentation des différents composants

### A. Le Backend Hub Entreprendre

Le Backend est une API centralisée répondant aux sollicitations des 3 portails Web : le Front-Office, le Back-Office et le Front-Agent.

L'outil **Maven** est intégré au projet afin de gérer les dépendances et lancer les builds.

Le framework **SPRING** et son module **Spring Boot** ont été utilisés afin de tirer parti des principaux modules que composent SPRING :

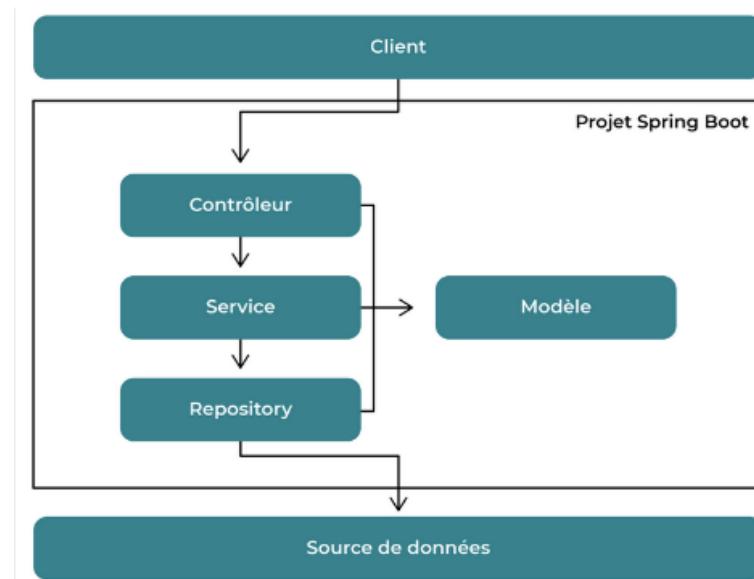
- **Spring Data REST** afin d'obtenir une **API REST** (ensemble de contraintes architecturales qui facilite la communication entre les API) qui fournira des services à la partie front grâce à des requêtes **HTTPS** (GET, POST...).
- **Spring data jpa** qui permet d'améliorer les couches d'accès aux données basées sur **JPA**. Java Persistence API (**JPA**) est une norme Java qui permet de définir des objets Java (**entity**) qui sont simplement des instances de classes qui seront persistantes. Il s'agit d'une approche de l'Object Relationship Mapping (**ORM**), qui nous permet de récupérer, stocker, mettre à jour et supprimer les données dans la base de données relationnelle. Les annotations JPA décrivent comment une classe Java donnée et ses variables correspondent à une table donnée et à ses colonnes dans une base de données.

- **Spring MVC** qui permet d'utiliser les contrôleurs Spring Web MVC qui sont là pour permettre de gérer les requêtes HTTP entrantes tout en nous apportant des facilités de développement pour le binding de données, la validation, la gestion du modèle.
- **Spring Security** pour l'authentification et le contrôle d'accès afin de sécuriser l'application. Il est configuré avec Keycloak via un adaptateur client. **Keycloak** est une solution open source de gestion des identités et des accès destinés aux applications et services modernes afin de sécuriser l'application. Il offre des fonctionnalités telles que l'authentification unique (SSO), les adaptateurs clients, une console d'administration et une console de gestion des comptes. Hub Entreprendre utilise l'API REST de Keycloak pour générer et rafraîchir les jetons d'accès afin de gérer les accès à l'application.

L'application a été divisé en plusieurs couches en conformité avec les principes de conception (responsabilité unique, couplage faible et cohésion forte) ce qui permet de faciliter la maintenance du code et d'intervenir indépendamment sur chacune des couches de l'application sans que les autres soient impactées.

Chaque couche a un rôle et intègre le **Design Pattern MVC** (Modèle Vue Contrôleur) :

- Le **Modèle** étant l'implémentation des objets métiers qui seront manipulés par les autres couches.
- La **Vue** au niveau du client pour afficher les informations (assuré par le front Angular).
- Le **Contrôleur** gère les interactions entre l'utilisateur et le système. Dans le cadre d'une application Web, les interactions avec le serveur correspondent aux requêtes HTTP émises par le navigateur client.
- La couche **Service** qui implémente des traitements métiers spécifiques à l'application Hub.
- La couche **Repository** qui traite les interactions avec les sources de données externes (BDD ici)



Voici la description des packages organisant le Backend :

**Packages Config** : Package rassemblant les classes de configuration pour les clients Elasticsearch, progos et WebMvcConfig (définit des options pour la personnalisation ou l'ajout à la configuration Spring MVC par défaut)

**Package Constant** : Package qui rassemble des classes définissant les constantes utilisées dans le reste de l'application comme entre autres pour la gestion des autorisations d'accès des différents Endpoint suivant le rôle de l'utilisateur.

**Package controller** : Les Contrôleurs gèrent les requêtes des utilisateurs. Ils sont responsables de retourner une réponse avec l'aide du Model en exposant des Endpoints afin que l'IHM puisse recevoir un service à travers des requêtes http.

**Package domain** : Il rassemble des classes par domaine comme les énumérations, les Jsons, les XML, les emails.

**Package entity** : Les entités sont des objets Java qui sont simplement des instances de classes persistantes grâce à JPA (approche de l'Object Relationship Mapping). Ce package contient aussi les DTO et les mappers entre DTO et Entity.

**Package exception** : On peut y trouver toutes les classes pour les exceptions

**Package gda** : Il contient toutes les classes qui concernent les interactions avec l'API Grand Angle (récupération des données relatif aux paiements faits par la Région et versées aux entreprises)

**Package logging** : On y trouve la classe intercepteur utilisée pour loger avec log4j2 toutes les actions REST effectuées par un utilisateur.

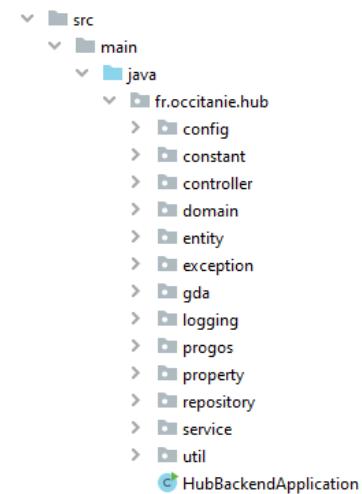
**Package progos** : Il contient toutes les classes qui concernent les interactions avec l'API progos (pour récupérer les dossiers de subventions)

**Package property** : Contient la classe pour une configuration externalisée afin de lier et valider des propriétés externes (par exemple, à partir d'un fichier «.properties »)

**Package repository** : Ce package rassemble tous les repository qui implémentent soit l'interface « CrudRepository » pour la BDD relationnelle, soit l'interface « ElasticsearchRepository » pour la BDD no SQL.

**Package Service** : Les Interfaces et leurs implémentations pour les traitements métiers spécifiques à l'application.

**Package util** : Les classes pour filtrer, classer et gérer les dates.



## B. Front Back-Office

L'application web Back-Office est une application permettant aux administrateurs Région d'édition le contenu éditorial, d'administrer les dispositions et les interlocuteurs du Hub Entreprendre.

Le Back-Office a été réalisé avec **ANGULAR** qui est un framework permettant de créer des **SPA** (Single page Application) en utilisant du **HTML** et du **TypeScript**. Il implémente les fonctionnalités principales et optionnelles sous la forme d'un ensemble de bibliothèques TypeScript que vous importez dans l'application.

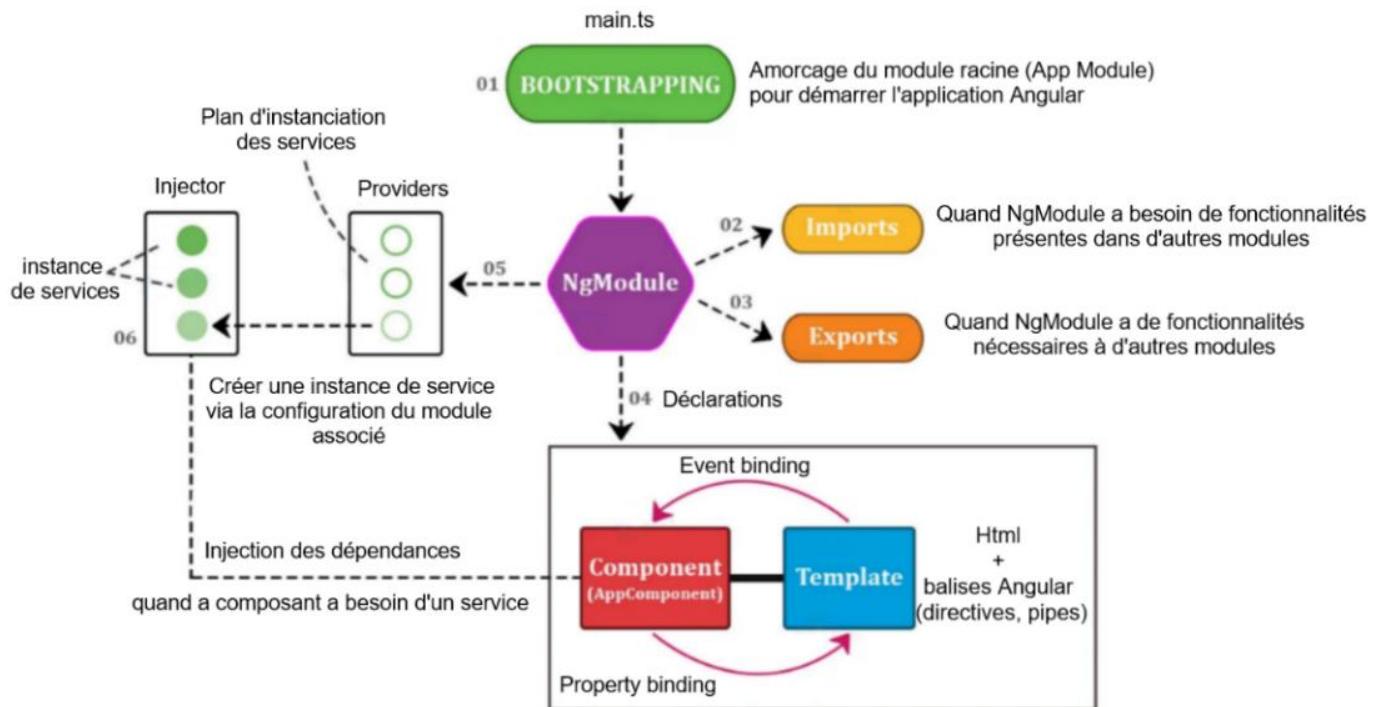
L'architecture de l'application Angular repose sur certains concepts fondamentaux. Les éléments de base du cadre Angular sont des composants organisés en **NgModules**. Les NgModules rassemblent le code connexe en ensembles fonctionnels et une application Angular est définie par un ensemble de NgModules.

Les **composants** définissent des **vues**, qui sont des ensembles d'affichage qu'Angular peut choisir et modifier en fonction de la logique et des données de l'application. Les composants utilisent des **services**, qui fournissent des fonctionnalités spécifiques non directement liées aux vues. Les fournisseurs de services (**provider**) peuvent être injectés dans les composants en tant que dépendances, ce qui rend votre code modulaire, réutilisable et efficace.

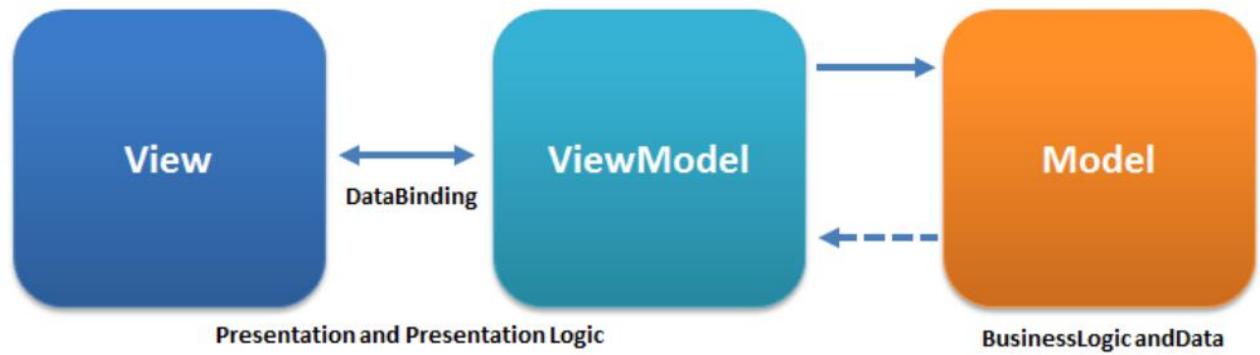
Les **modules**, **composants** et **services** sont des classes qui utilisent des **décorateurs**. Ces décorateurs marquent leur type et fournissent des métadonnées qui indiquent à Angular comment les utiliser. Les métadonnées d'une classe de composant l'associent à un **modèle** qui définit une vue. Un modèle combine du HTML ordinaire avec des **directives** Angular et des balises de liaison qui permettent à Angular de modifier le HTML avant de le rendre pour l'affichage.

Les métadonnées d'une classe de service fournissent les informations dont Angular a besoin pour la mettre à la disposition des composants par le biais de l'**injection de dépendances** (DI).

Les composants d'une application définissent généralement de nombreuses vues, organisées de manière hiérarchique. Angular fournit le service **Router** pour vous aider à définir des chemins de navigation entre les vues.



L'architecture d'Angular est simplement un arbre hiérarchique de composants. Ces composants sont réutilisables et indépendants rendant le code Angular hautement testable. Ils interagissent avec des services, qui servent à gérer la logique non-graphique de l'application, et forment ainsi un pattern **MVVM** (Model -View - ViewModel)



L'application est organisée en couches et voici la description détaillée :

**app-auth.module.ts** : Module qui gère l'authentification unique avec Keycloak. C'est le mécanisme d'authentification qui permet aux utilisateurs d'accéder à plusieurs applications avec un seul ensemble d'informations d'identification en se connectant une seule fois grâce à un **JWT** (Json WEB Token).

**app-routing.module.ts** : Ce composant active la navigation en interprétant une URL de navigateur et en chargeant le composant correspondant, et donc les vues.

**app.component.\*** sont les composants racine avec son code html, css et TS.

**app.module.ts** : Ce composant décrit la façon dont les parties de l'application s'assemblent. Il s'agit du module racine utilisé pour le démarrage et l'ouverture de l'application. On y définit l'ensemble des imports qui seront utilisé dans notre module (librairies NPM), on y définit l'ensemble des composants qui y seront déclaré, mais aussi des services.

**Dossier components** : Rassemble tous les composants réutilisables Angular.

**Dossier constant** : Un fichier index.ts contenant des objets définissant les constantes utilisées par l'application.

**Dossier directive** : Contient la classe qui définit une directive Angular pour attacher un comportement personnalisé aux éléments du DOM concernant l'affichage des numéros de téléphone.

**Dossier guards** : Les guards dans Angular permettent de contrôler l'accès à une route ou le départ depuis une route mais n'est pas un mécanisme de sécurité. Ils servent à améliorer la "User eXperience" en évitant par exemple l'accès à des routes qui ne fonctionneraient pas car l'accès aux données serait rejeté par l'API.

**Dossier interceptors** : Classe qui intercepte et traite une requête HTTP et transforme la requête sortante avant de la passer à l'intercepteur suivant dans la chaîne en y incorporant le token de Keycloak.

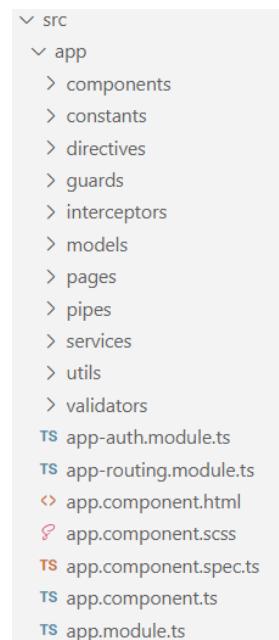
**Dossier models** : Ce dossier rassemble les modèles pour gérer les actualités du Hub et les emails.

**Dossier pages** : Rassemble tous les composants correspondant aux pages du portail Back-Office organisés de manière hiérarchique.

**Dossier pipes** : On y trouve tous les Pipes qui sont des filtres utilisables directement depuis la vue afin de transformer les valeurs lors du « binding » (date, heure, les noms, etc.)

**Dossier service** : Y sont décrit tous les services définis pas des classes TS composée d'attributs et de méthodes, dont l'instanciation est gérée par Angular. Une fois instancié, il est injectable dans n'importe lequel de vos composants ou dans un autre service. Ces services sont utilisés par exemple pour la communication avec une API ou une base de données, l'implémentation d'un cache d'objets ou la gestion de la session utilisateur.

**Dossier utils** : Une classe utils.ts contenant des méthodes utiles à travers l'application comme la manipulation de chaîne de caractères.



**Dossier validators** : Les validators sont des fonctions de validation que l'on passe en paramètre dans les constructeurs des controls (FormControl, FormGroup et FormArray). C'est pour par exemple la configuration de la validation de formulaire (FormControl).

- **Front Office et Front Agent**

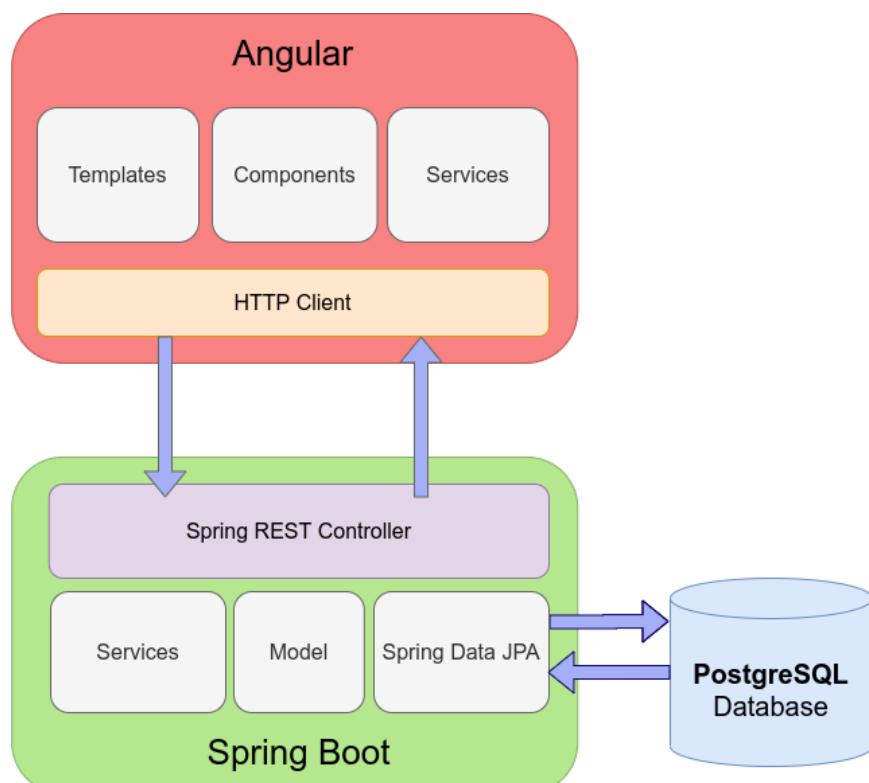
Les applications web Front Office et Front Agent sont aussi des applications SPA développer avec Angular. Front Office permet aux entreprises de se renseigner, de soumettre et de suivre les dossiers d'aide aux entreprises de la région Occitanie. Front Agent est le portail des Agents et Partenaires de la région permettant le suivi et l'accompagnement des entreprises et des projets.

Je ne décris pas en détail leurs architectures car c'est hors champs des améliorations que j'avais à faire.

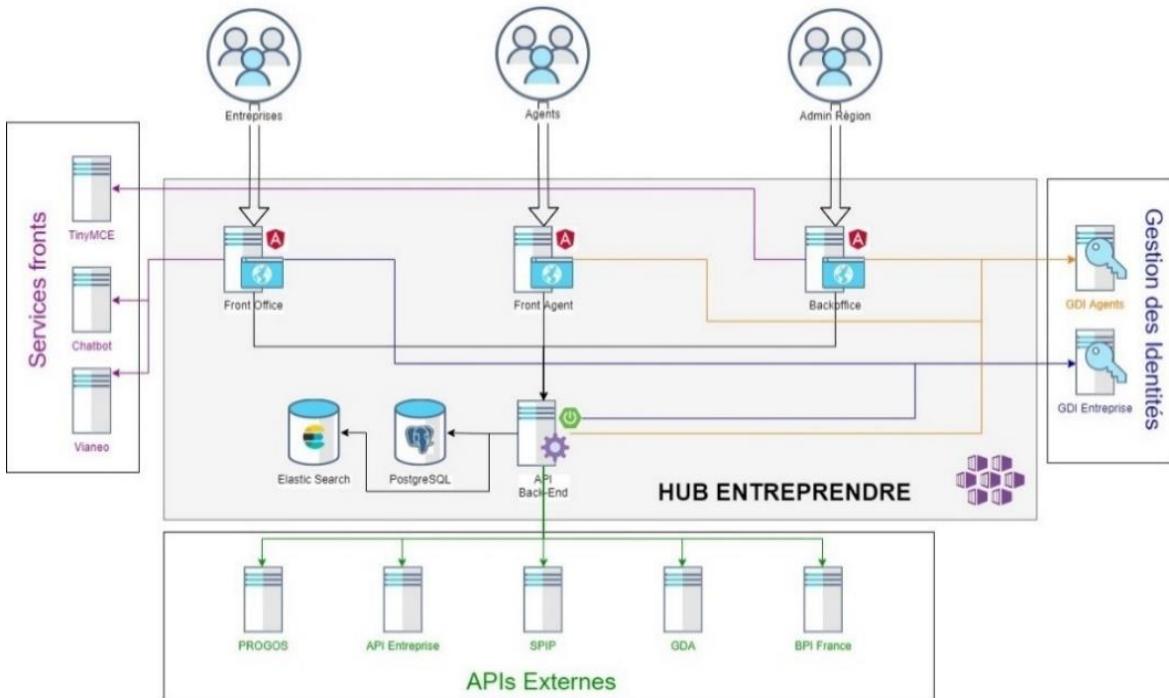
- **Bases de données**

Les bases de données de l'application Hub sont une **BDD relationnelle « PostgreSQL »** et une **BDD NoSQL « ElasticSearch »** liée au Backend du Hub.

## C. Fonctionnement général de Hub Entreprendre



## D. Plateforme technique



Un schéma d'architecture détaillé est disponible avec tous les flux implémentés en HTTPS ([annexe 6](#))

## 6.2. Les évolutions

Je vais ici me concentrer sur l'évolution concernant l'extraction des dispositifs pour illustrer la méthodologie utilisée.

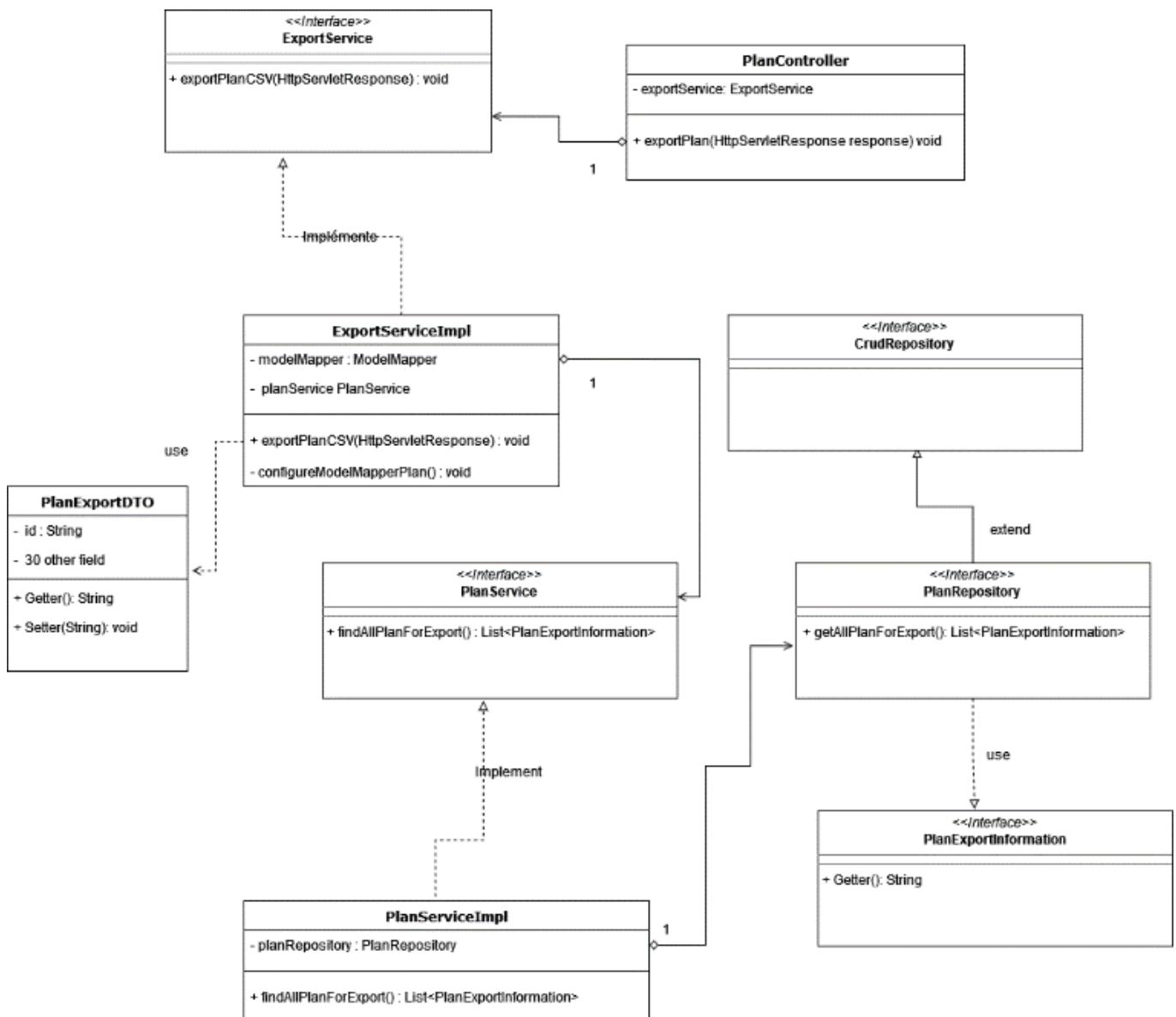
### A. Diagramme de classe

Pour valider les cas d'utilisation, et pour comprendre la logique de l'application j'ai continué avec les diagrammes de classes.

Alors que le diagramme de cas d'utilisation montre un système du point de vue des acteurs, le diagramme de classes en montre la structure interne. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation. Il s'agit d'une vue statique, car on ne tient pas compte du facteur temporel dans le comportement du système. Il modélise les concepts du domaine d'application ainsi que les concepts internes créés de toutes pièces dans le cadre de l'implémentation d'une application.

Un diagramme de classe est un type de diagramme UML qui décrit un système en visualisant les différents types d'objets au sein d'un système et les types de relations statiques qui existent entre eux. Il illustre également les opérations et les attributs des classes. Les principaux éléments de cette vue statique sont les classes et leurs relations :

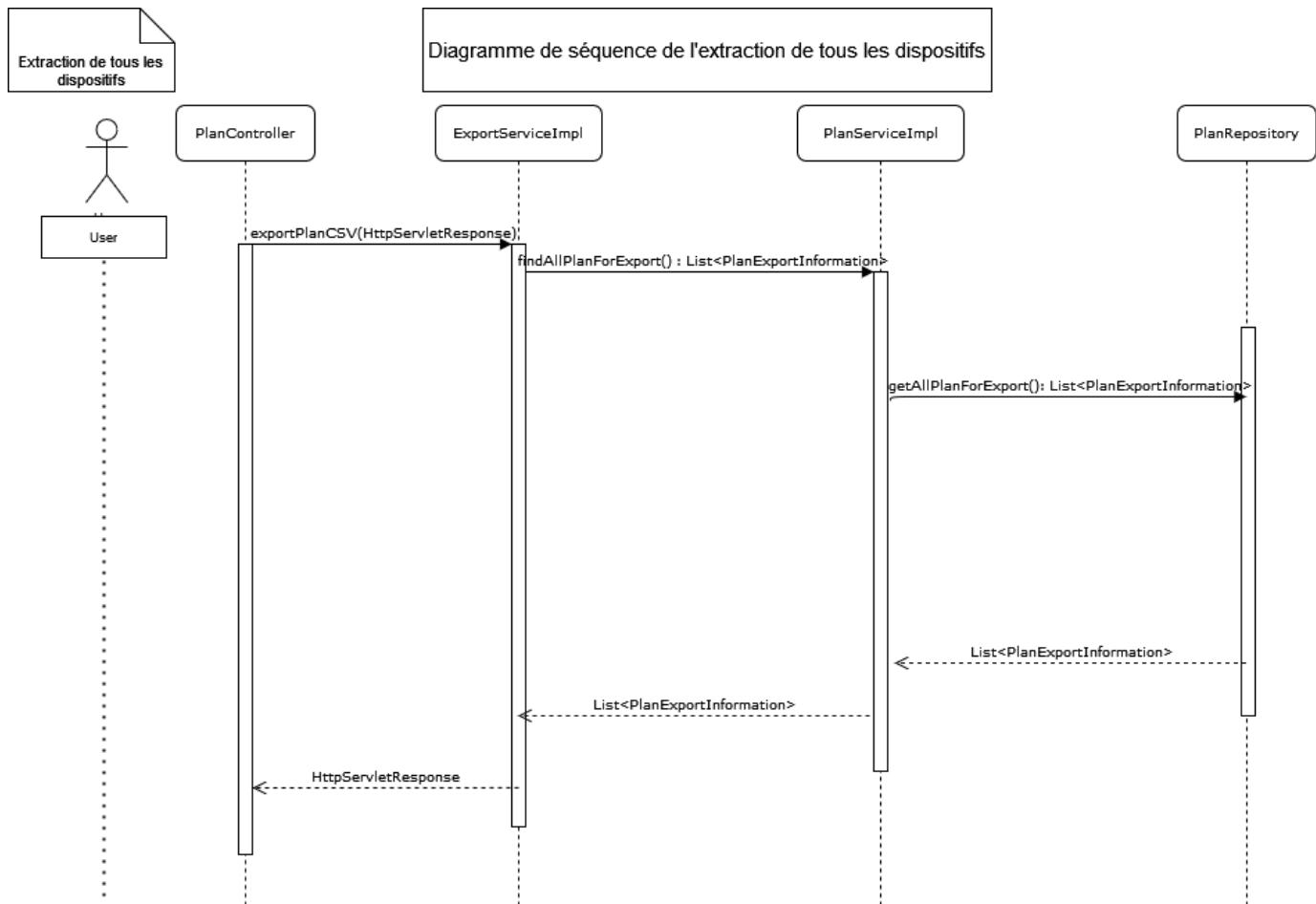
Ci-dessous l'exemple du diagramme de classe de l'extraction des dispositifs :



## B. Diagramme de séquence

Le diagramme de séquence modélise les interactions entre les objets dans un cas d'utilisation unique de façon chronologique. Ils illustrent la manière dont les différentes parties d'un système interagissent entre elles pour exécuter une fonction, et l'ordre dans lequel les interactions se produisent lorsqu'un cas d'utilisation particulier est exécuté.

Ci-dessous l'exemple du diagramme de séquence de l'extraction des dispositifs :



## 7. Implémentation

Après l'analyse des besoins et la conception j'ai entamé la réalisation des évolutions. Cette partie présentera les travaux réalisés.

### 7.1. PR1 : Modification url associée au bouton "contacter un expert de proximité"

Cette demande concerne le changement d'une URL se trouvant dans le backend.

J'ai simplement modifié dans le fichier fr.json ([annexe 7](#)), l'url se rapportant à la quatrième carte du composant Angular [contact-cards.component.html](#) du Front-Entreprise.

```
<hub-contact-card class="contact-card"
  [title]="'contact-cards.card-4.title'"
  [content]="'contact-cards.card-4.content'"
  [buttonLabel]="'contact-cards.card-4.buttonLabel'"
  [buttonLink]="'contact-cards.card-4.buttonLink'">
</hub-contact-card>
```

### 7.2. PR8 : Modification url du flux BPI

Comme au-dessus, cette évolution est très simple. J'ai dans le fichier **application.yml** de configuration du backend Hub Entreprendre modifié valeur de la clé « url ».

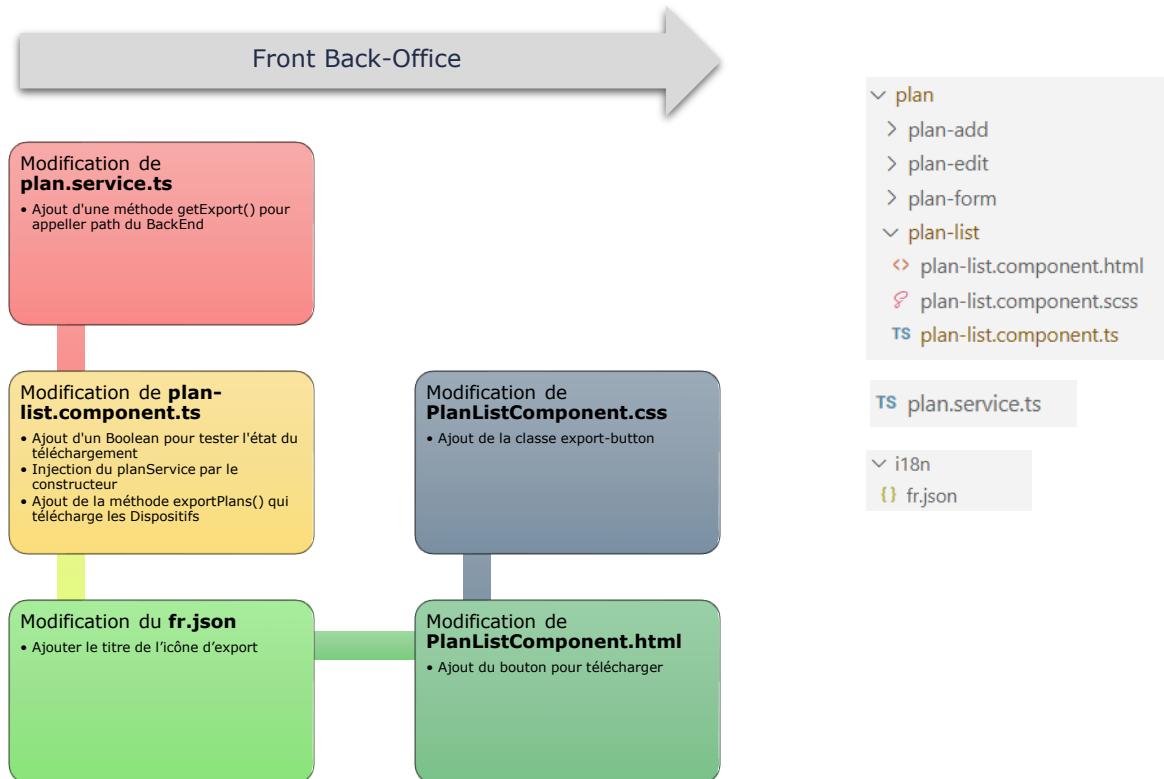
```
bpi:
  dossiers-projecteurs:
    url: https://bpifrance-creation.fr/syndication/dossiers_projecteurs_direct
    user: adocc_dev
    password: 38P8B9!K!wi!k
    timeout: 15000
```

## 7.3. PR7 : Extraction des dispositifs

Cette évolution a nécessité mon intervention sur le backend et sur le front Back-Office.

### A. Le front Back-Office

On peut voir en dessous un résumé des étapes des interventions sur le front Back-office :



- **PlanService**

J'ai d'abord commencé par la modification du `PlanServices.ts` se trouvant dans le dossier Service et j'y ai ajouter la méthode `getExport()` :

```

//méthode pour export les Dispositifs à l'endpoint du Backend
getExport(): Promise<ArrayBuffer> {
    return this.http
        .get(`${this.api.hubApiHost}/dispositifs/export`, {
            responseType: 'arraybuffer'
        })
        .toPromise();
}

```

**ArrayBuffer** : Représente un tampon de données binaires, qui est utilisé pour stocker des données pour les différents tableaux typés. Les ArrayBuffers ne peuvent pas être lus ou écrits directement, mais peuvent être passés à un tableau typé pour interpréter le tampon brut selon les besoins, ce qui sera fait plus tard.

L'interface **Promise** représente un intermédiaire vers une valeur qui n'est pas nécessairement connue au moment de sa création. Cela permet d'associer des gestionnaires au succès éventuel d'une action asynchrone et à l'obtention d'une erreur. Ainsi, des méthodes asynchrones renvoient des valeurs comme les méthodes synchrones, la seule différence est que la valeur retournée par la méthode asynchrone est une promesse (d'avoir une valeur plus tard).

La propriété **api** est de type  **ApiService** qui est un service qui a pour propriété **hubApiHost** qui renseigne l'URL du backend.

La propriété **http** est de type **HttpClient** fournit par Angular. Ce service est disponible sous la forme d'une classe injectable avec des méthodes pour effectuer des requêtes http.

J'ai utilisé la méthode **HttpClient.get ()** pour effectuer une requête GET sur le backend. La méthode asynchrone envoie une requête HTTP, et récupère un Observable qui émet les données demandées lorsque la réponse est reçue. La méthode prend deux arguments : l'URL du Endpoint, et un objet qui est utilisé pour configurer la requête.

Au final, la méthode permet de recevoir une promesse de ArrayBuffer suite à une requête GET sur le endpoint /dispositif/export du backend.

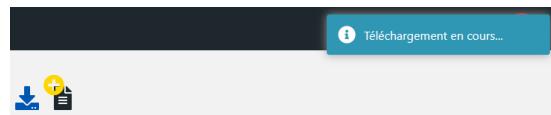
- **Plan-list-component**

#### Fichier TypeScript

J'ai ensuite modifié le fichier TS du composant **Plan-list** en y ajoutant la méthode **exportPlans ()** :

```
 /**
 * permet le téléchargement de tous les Dispositifs
 */
exportPlans() {
  if (!this.isExportDownloading) {
    this.isExportDownloading = true;
    const toastr = this.toastr.info('Téléchargement en cours...', '', {
      disableTimeOut: true,
      tapToDismiss: false
    });
    this.planService
      .getExport()
      .then(data => {
        this.isExportDownloading = false;
        toastr.remove(toastr.toastId);
        const filename = `${moment().format(
          dateFormatExport
        )} - Export dispositifs HUB.csv`;
        const file = new Blob([data], { type: 'text/csv' });
        FileSaver.saveAs(file, filename);
      })
      .catch(err => {
        this.isExportDownloading = false;
        toastr.remove(toastr.toastId);
        toastr.error('Le téléchargement a échoué.');
      });
  }
}
```

- 1) La méthode va tester si y'a pas un téléchargement déjà en cours suivant la valeur du boolean `isExportDownloading`.
- 2) Elle va ensuite afficher avec `Toastr` une notification en haut de la page signalant que le téléchargement est en cours. `Toastr` est une bibliothèque JavaScript utilisant jQuery et permettant l'affichage de notifications en tout genre : succès, information, mise en garde, erreur. Chaque notification entièrement personnalisable.



- 3) Alors elle va faire appel à la méthode `getExport()` du `planService` vu précédemment qui récupère une `Promise<ArrayBuffer>`. Les données vont être passés à un tableau typé `Blob` (objet, semblable à un fichier, qui est immuable et qui contient des données brutes. On peut le lire grâce à l'objet `FileReader`) et stockées dans la variable `file`.
- 4) Enfin, elle va sauvegarder grâce à `FileSaver` un fichier contenant les données de `file` et portant le nom constitué de la concaténation de la date d'aujourd'hui et de « Export dispositifs Hub.csv ». `FileSaver.js` est une bibliothèque JS pour enregistrer des fichiers côté client.
- 5) S'il y a une erreur, `Toastr` va afficher un message que ça a échoué.

#### Fichier HTML :

Pour finir, j'ai ajouté un bouton cliquable qui fait appel à la méthode `exportPlans()` mais qui est désactivé si le téléchargement est en cours :

```
<div class="top-search">
  <div class="search-bar">
    <input class="col-md-10" type="text" id="search" name="search" placeholder="Rechercher"
      (keyup)="filter($event)"
      <i class="fa fa-search pointer"></i>
    </div>

    <a title="{{ 'support.export' | translate | ucfirst }}" class="hub-back-office-search--add-item export-button fa fa-download" [ngClass]="{disabled: isExportDownloading}"
      (click)="exportPlans()"></a>

    <a title="{{ 'plan.create-plan' | translate | ucfirst }}" class="fa fa-file-text pointer"
      routerLink="/plan-add"
      <i class="fa fa-plus-circle pointer" title="{{ 'plan.create-plan' | translate | ucfirst }}></i>
    </a>
  </div>
```

#### Fichier CSS :

Ajout de la classe `export-button` :

```
.export-button {
  margin-right: 20px;

  &:disabled {
    pointer-events: none;
  }
}
```

- **fr.json**

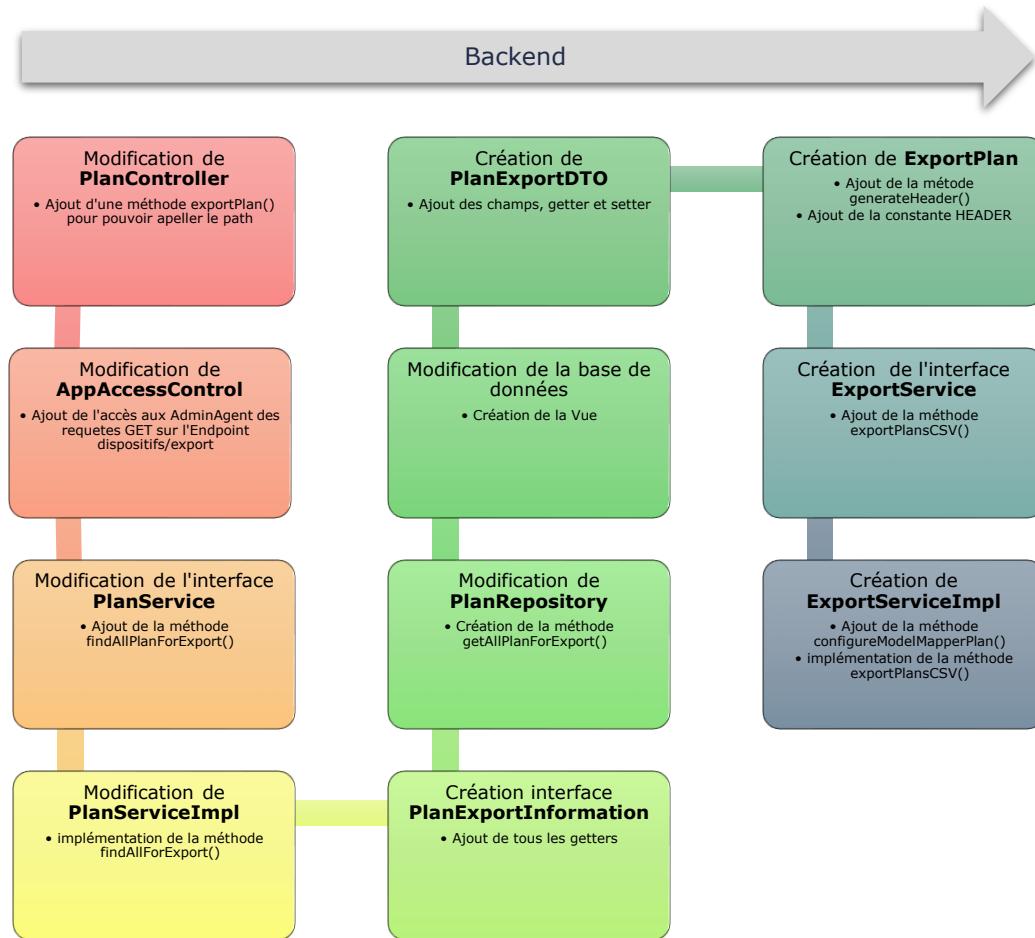
C'est le fichier json dans le dossier **i18n** qu'on se sert pour prendre en compte l'internationalisation de l'application. J'ai rajouté la correspondance de la valeur pour « export » correspondant au texte qui s'affiche quand on met la souris dessus.

```
- "export": "exporter les données",
```



## B. Le Backend

On peut voir en dessous un résumé des étapes des interventions sur le Backend :



- **PlanController**

J'ai commencé par ajouter la méthode **exportPlan ()** dans le **Controller** qui sera appelé suite à une requête GET à l'Endpoint **dispositifs/export**

```

    @RequestMapping(value = "/dispositifs")
    @RestController
    public class PlanController extends AbstractController<Plan> {

        /**
         * GET /export is a function that allows to export plans in a csv file
         */
        @mathieu.romatet
        @GetMapping(value="/export")
        public void exportPlan(HttpServletRequest response) throws IOException, CsvDataTypeMismatchException, CsvRequiredFieldEmptyException {
            exportService.exportPlansCSV(response);
        }
    }

```

- **AppAccessControl**

Puis j'ai donné accès à ce Endpoint aux utilisateurs ayant le rôle Admin et faisant une requête GET en utilisant la classe **AccessControl** dédiée.

```

// PlanController
ac.forMethod(HttpMethod.POST, uri: "/dispositifs", AccessControl.ROLE_ADMINAGENT);
ac.forMethod(HttpMethod.DELETE, uri: "/dispositifs/{id}", AccessControl.ROLE_ADMINAGENT);
ac.forMethod(HttpMethod.PUT, uri: "/dispositifs", AccessControl.ROLE_ADMINAGENT);
ac.forMethod(HttpMethod.GET, uri: "/dispositifs/export", AccessControl.ROLE_ADMINAGENT);

```

- **PlanService**

Ajout de la méthode **findAllPlanForExport ()** dans l'interface

```

/**
 * retourne toutes les informations nécessaires pour extraction des dispositifs
 * @return
 */
1 implementation
List<PlanExportInformation> findAllPlanForExport();

```

- **PlanServiceImpl**

J'implémente la méthode **findAllPlanExport ()** pour appeler la méthode **getAllPlanExport ()** dans la couche repository

```

/**
 *Retourne toutes les informations nécessaires pour l'export en CSV
 * @return
 */
1 usage
@Override
public List<PlanExportInformation> findAllPlanForExport() {
    return planRepository.getAllPlanForExport();
}

```

- **PlanExportInformation**

C'est une interface que je vais utiliser avec la fonctionnalité de projection de Spring Data : l'interface contenant que des getters va recevoir le résultat d'une requête SQL personnalisée

```
/**  
 * This is an Interface-Based Projection !  
 *  
 */  
14 usages  
public interface PlanExportInformation {  
  
    String getId();  
    String getName();  
    String getCategory();  
    String getCatchPhrase();  
    String getKeyNumber();
```

- **PlanRepository**

La méthode récupère le résultat de la vue **planExtractView** créé en BDD

```
@Query(value = "Select * from plansExtractView"  
       , nativeQuery = true)  
List<PlanExportInformation> getAllPlanForExport();
```

- **planExtractView**

J'ai créé la vue **planExtractView** dans la base de données correspondant aux données que l'on veut récupérer pour l'extraction.

Script de création de la vue en (**annexe 8**).

Pour l'extraction de données en CSV, j'ai utilisé **OpenCSV** qui est une bibliothèque d'analyse CSV pour Java. Pour se faire j'ai créé les classes **PlanExportDTO** et **ExportPlan** que je vais utiliser pour générer mon CSV :

- **PlanExportDTO**

DTO où l'on spécifie une liaison entre un numéro de colonne de l'entrée CSV et un attribut.

```
public class PlanExportDTO {  
  
    /**  
     * This annotation is used to define the columns of the csv file  
     */  
2 usages  
@CsvBindByPosition(position = 0)  
private String id;  
  
2 usages  
@CsvBindByPosition(position = 1)  
private String name;  
  
2 usages  
@CsvBindByPosition(position = 2)  
private String category;
```

- **ExportPlan**

MappingStrategy pour mapper les colonnes du fichier CVS avec des noms personnalisés à des attributs du DTO.

```
1 /**
 * set the name of the columns in the csv file
 *
 * @param <PlanExportDTO>
 */
2 usages
public class ExportPlan<PlanExportDTO> extends ColumnPositionMappingStrategy<PlanExportDTO> {
    1 usage
    private static final String[] HEADER = new String[]{"id", "name", "Courriel", "catégorie", 'chiffres_clés', "active", "type", "filières", "catégories_de_projet", "dépenses_eliminées", "tags", "naf_exclus", "createdat", "updatedat", "startdate", "enddate", "opérateur", "documents", "Documents visibles uniquement sur le portail Agents (Lien vers la FAQ, "Liens de dépôt de dossier URL (URL + nom du lien s'il existe, séparés par des slashes "maxnbemployees", "Lien URL de la page"};

    @Override
    public String[] generateHeader(PlanExportDTO dto) throws CsvRequiredFieldEmptyException {
        super.generateHeader(dto);
        return HEADER;
    }
}
```

- **ExportService**

```
1 implementation  ± mathieu.romatet +1 *
public interface ExportService {

    /**
     * @param response
     * @throws IOException
     * @throws CsvDataTypeMismatchException
     * @throws CsvRequiredFieldEmptyException
     */
    1 usage 1 implementation  ± mathieu.romatet
    void exportPlansCSV(HttpServletRequest response) throws IOException, CsvDataTypeMismatchException, CsvRequiredFieldEmptyException;
}
```

- **ExportServiceImpl**

```
@Service
public class ExportServiceImpl implements ExportService {

    private static final String DATE_FORMAT_DD_MM_YYYY = "dd/MM/yyyy";

    private ModelMapper modelMapper;

    private PlanService planService;

    private Converter<Date, String> convertDate;

    public ExportServiceImpl(PlanService planService) {
        modelMapper = new ModelMapper();

        this.planService = planService;

        SimpleDateFormat dateFormat = new SimpleDateFormat(DATE_FORMAT_DD_MM_YYYY);
        convertDate = new AbstractConverter<Date, String>() {
            protected String convert(Date date) {
                if (date == null) {
                    return StringUtils.EMPTY;
                }
                return dateFormat.format(date);
            }
        };
    }
}
```

J'utilise la classe **ModelMapper** pour effectuer le mappage d'objets, charger les définitions de mappage et enregistrer les mappeurs.

Premièrement, j'implémente **convertDate** de Type **Converter<>** qui convertit les instances du type source **Date** en instances du type destination **String** (vide si pas de **Date**)

```
@PostConstruct
private void configureModelMapper() { configureModelMapperPlan(); }

/**
 * Convertit et mappe les données de chaque Dispositifs pour passer des Dispositifs aux PlanExportDTO
 */
private void configureModelMapperPlan() {
    TypeMap<PlanExportInformation, PlanExportDTO> typeMap = modelMapper.createTypeMap(PlanExportInformation.class, PlanExportDTO.class);
    typeMap.addMappings(mapper -> {
        mapper.using(this.convertDate).map(PlanExportInformation::getCreatingDate, PlanExportDTO::setCreatingDate);
        mapper.using(this.convertDate).map(PlanExportInformation::getUpdatingDate, PlanExportDTO::setUpdatingDate);
        mapper.using(this.convertDate).map(PlanExportInformation::getStartingDate, PlanExportDTO::setStartingDate);
        mapper.using(this.convertDate).map(PlanExportInformation::getEndingDate, PlanExportDTO::setEndingDate);
    });
}
```

Puis je configurer le **modelMapper** à l'aide de **convertDate** pour qu'il mappe les attributs **Date** que je souhaite.

```


/*
 * Ecrit la réponse http dans un fichier CSV (composé de tous les InterlocutorExportDTO)
 *
 * @param response
 * @throws IOException
 * @throws CsvDataTypeMismatchException
 * @throws CsvRequiredFieldEmptyException
 */
1 usage
@Override
public void exportPlansCSV(HttpServletRequest response) throws IOException, CsvDataTypeMismatchException, CsvRequiredFieldEmptyException {
    Type listType = new TypeToken<List<PlanExportDTO>>().getType();
    ArrayList<PlanExportInformation> plans = (ArrayList<PlanExportInformation>) planService.findAllPlanForExport();
    ArrayList<PlanExportDTO> dtos = modelMapper.map(plans, listType);

    String filename = "export.csv";

    response.setContentType("text/csv");
    response.setCharacterEncoding("CP1252");
    response.setHeader(HttpHeaders.CONTENT_DISPOSITION,
        value: "attachment; filename=\"" + filename + "\"");

    ExportPlan<PlanExportDTO> mappingStrategy = new ExportPlan<>();
    mappingStrategy.setType(PlanExportDTO.class);

    //create a csv writer

    StatefulBeanToCsv<PlanExportDTO> writer = new StatefulBeanToCsvBuilder<PlanExportDTO>(response.getWriter())
        .withMappingStrategy(mappingStrategy)
        .withQuotechar(CSVWriter.DEFAULT_QUOTE_CHARACTER)
        .withEscapechar('"')
        .withSeparator(';')
        .withOrderedResults(false)
        .build();

    writer.write(dtos);
}


```

Et enfin je récupère la liste des dispositifs que je mappe en DTO. Je créé un CSV correspondant grâce **OpenCSV** en lui donnant un nom et un type et un encodage que j'intègre dans la réponse http.

## 7.4. PR5 : Extraction des Utilisateurs

### A. Le front Back-Office

- **Index.ts**

```


export const HomePageCards = [
    {
        title: 'home.connected.users',   Modification du fichier index.ts pour mettre les
        icon: 'fas fa-user-tie',         informations de la nouvelle carte (label, icone, etc...).
        extractURL: 'extract'          C'est le 12em objet « card » qui se trouve dans le tableau
    }
];


```

- **home.connected.component**

Modification de **home.connected.component** qui est le composant affiché juste après la connexion de l'utilisateur.

```
@Component({
  // tslint:disable-next-line:component-selector
  selector: 'back-office-home-connected',
  templateUrl: './home.connected.component.html',
  styleUrls: ['./home.connected.component.scss']
})
export class HomeConnectedComponent implements OnInit {
  constructor() {}

  datas = homePageCards;

  ngOnInit() {}
}
```

Il est composé de plusieurs composants fils **back-office-card\_menu**

```
<div class="row mb-5">
  <div class="col-4">
    <back-office-card-menu [data]="datas[9]" [centerBottom]="true"></back-office-card-menu>
  </div>
  <div class="col-4">
    <back-office-card-menu [data]="datas[10]" [centerBottom]="true"></back-office-card-menu>
  </div>
  <div class="col-4">
    <back-office-card-menu [data]="datas[11]" [centerBottom]="true"></back-office-card-menu>
  </div>
</div>
```

J'utilise le « property binding » d'Angular pour récupérer le 12em objet de **homePageCards** et le transmettre au composant fils

- **card-menu\_component**

```
@Component({
  // tslint:disable-next-line:component-selector
  selector: 'back-office-card-menu',
  templateUrl: './card-menu.component.html',
  styleUrls: ['./card-menu.component.scss']
})
export class CardMenuComponent extends ListComponent implements OnInit {
  @Input() data: any;
  @Input() centerBottom = false;
```

Pour transmettre des données d'un composant père à un fils, Angular fournit le décorateur **@Input**, celui-ci va permettre d'identifier une propriété du composant pouvant recevoir la donnée.

J'ai ajouté dans l'html un nouveau type de bouton qui s'affiche en testant la variable **data** avec une directive **\*ngIf** pour afficher le bon type de card.

```
<!-- Si on a pas le bouton 'create' alors on force le 'voir tous >' sur la droite -->
<div class="container-bottom" [ngClass]="!data?.createUrl ? 'btn-right' : ''">*ngIf="!centerBottom">
  <a *ngIf="data?.createUrl" class="container-bottom--create-btn pointer" routerLink="/{{data.createURL}}">{{
    'create' | translate | ucfirst }}</a>
  <a class="container-bottom--show-all pointer" routerLink="/{{data.showAllURL}}">{{
    'show-all-arrow' | translate | ucfirst }}</a>
</div>
<!-- Gestion des cards avec seulement le bouton gérer -->
<div *ngIf="centerBottom" class="d-flex justify-content-center">
  <a *ngIf="data?.manageURL" routerLink="/{{data.manageURL}}" class="pointer manage-url-button">{{{
    'handle' | translate | ucfirst }}</a>
</div>

<!-- Gestion des cards avec seulement le bouton extraire -->
<div *ngIf="centerBottom" class="d-flex justify-content-center">
  <a *ngIf="data?.extractURL" class="pointer manage-extract-button" (click)="exportUsers()">{{{
    'extract' | translate | ucfirst }}</a>
</div>
```

Puis j'ai créé dans le fichier Typescript la méthode **exportUsers ()** déclenché quand on clique sur le bouton.

Je ne vais pas détailler car c'est exactement la même chose par la suite que pour l'évolution précédente.

En annexe l'IHM de la page ([annexe 9](#)).

## B. Le Backend

Pour le backend, j'ai suivi exactement la même méthode que pour l'évolution précédente. Il n'y a que la requête SQL qui diffère.

Mais je me suis rendu compte que certaines informations qu'on me demande d'extraire (mail, numéro de téléphone) ne sont pas accessibles dans notre Base de données. Il fallait que je passe par l'api **GDI Entreprise**, en charge par une autre équipe, pour récupérer les données correspondantes aux **keycloakid** des utilisateurs.

Je les ai contactés pour qu'il m'explique la démarche à suivre mais je n'ai pas eu l'opportunité de concrétiser.

## 8. Sécurité

L'Open Web Application Security Project (OWASP) est une organisation à but non lucratif fondée en 2004 pour prévenir de manière proactive les attaques sur les applications web. Le Top 10 de l'OWASP fournit une base de référence avec une liste de contrôles à effectuer pour atténuer les risques les plus courants en matière de sécurité. Cette base de référence est également utilisée pour répondre à des normes réglementaires plus strictes, telles que le RGPD par exemple.

Voici en général les plus courant risques :

- L'injection
- Le piratage de session
- L'exposition de données sensibles
- Les entités externes XML (XXE)
- Le contournement des contrôles d'accès
- La mauvaise configuration de sécurité
- Cross-Site Scripting (XSS) ou failles XSS
- La désérialisation non sécurisée
- L'utilisation de composants présentant des vulnérabilités connues
- Le manque de surveillance et de monitoring.

### 8.1. Prévention des XSS dans Angular

Cross-site scripting (XSS) est une faille de sécurité qui permet à un attaquant d'injecter dans un site web un code client malveillant. Angular offre une plateforme sécurisée qui permet de construire l'application de manière à minimiser l'exposition aux attaques XSS. Mais on peut utiliser les moyens suivants pour sécuriser l'application contre :

- S'assurer-vous que le serveur Web renvoie l'en-tête HTTP **Content-Security-Policy** pour maintenir une liste de sources de confiance. Par la suite, l'application n'exécutera que le code provenant de ces sources de confiance. Cet en-tête pourrait ressembler à la ligne suivante : « Content-Security-Policy : script-src https://hostDuBackend »
- Utilisez les **Trusted Types** pour renforcer la sécurité du codage dans l'application. Pour activer les Trusted Types, le serveur Web doit envoyer un en-tête HTTP contenant les mots-clés de la politique d'Angular.
- Évitez d'utiliser des modèles de codage dangereux, comme l'accès direct aux éléments du DOM avec par exemple **innerHTML**

### 8.2. Prévention des XSS avec SPRING

Le module **Spring Security** fournit plusieurs en-têtes de sécurité par défaut et en particulier **X-XSS-Protection**. Il indique au navigateur de bloquer ce qui ressemble à du XSS et **Spring Security** ajoute automatiquement cet en-tête de sécurité à la réponse.

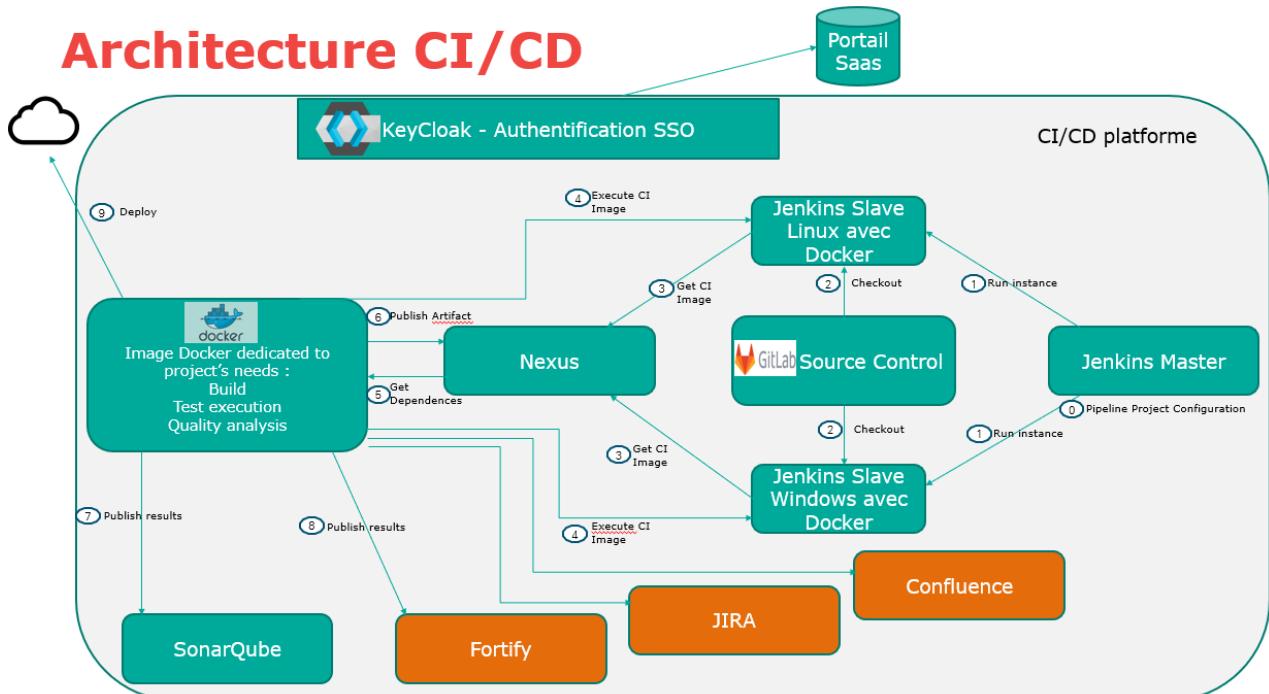
## 9. Déploiement

CodeP est la plateforme Inetum **d'intégration continue et de déploiement continu**. Elle permet de mettre en place une pratique DevOps consistant à développer, à tester et à délivrer des améliorations au code d'un logiciel ou aux environnements utilisateurs à l'aide d'outils automatisés.

<b>Sécurisée</b>
<ul style="list-style-type: none"> <li>• SSO et LDAP privé</li> <li>• Droits adaptés à chaque rôle</li> <li>• Cloisonnement des projets entre eux</li> </ul>
<b>Évolutive, ouverte et fiable</b>
<ul style="list-style-type: none"> <li>• Par la sélection de ses composants</li> <li>• Par la mise à jour régulière de ses composants</li> <li>• Par sa capacité à intégrer de nouveaux composants</li> </ul>

<b>Totalement intégrée</b>
<ul style="list-style-type: none"> <li>• Etroitement connectée aux outils digitaux collaboratifs et d'optimisation et renforcement du code</li> <li>• CI : Centrale pour assurer une vue unifiée des applications et produits</li> <li>• CD : Pour les déploiements multi-tenant Cloud ou interne</li> </ul>





**Gitlab** : Hébergement et gestion du code source



**Jenkins Master** : porte la configuration des pipelines (description du mécanisme d'intégration continu) du projet et effectue leurs exécutions sur une instance Jenkins slave



**Jenkins Slave Windows** : gère les projets de technologies Microsoft (Dot Net, ASP)



**Jenkins Slave Linux** : gère les projets hors technologies Microsoft (Java, NodeJS)



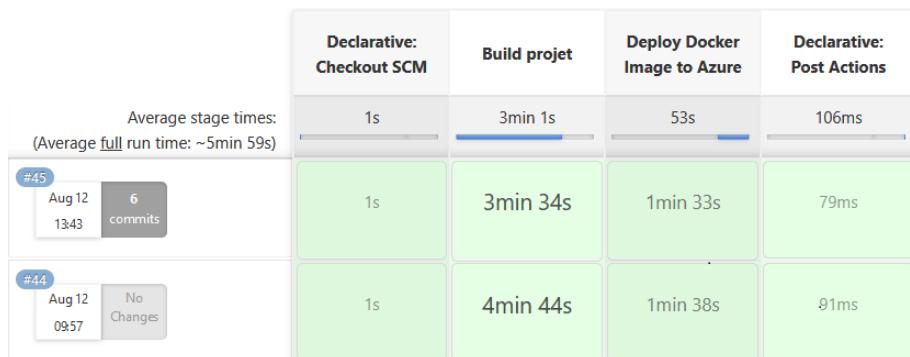
**Nexus** : Repository des composants tiers du projet (Librairie OJDBC, Cerbere) nécessaire pour stocker les livrables du projet (fichier EAR, WAR, Image docker)

**Docker** : Effectue le build du projet en s'appuyant sur l'image docker préalablement définie et contenant les outils nécessaires à la construction du projet

**SonarQube** : effectue l'analyse de qualité de code du projet et publie ses indicateurs

Le déploiement pour mon projet n'était pas opérationnel. J'ai dû dans un premier temps identifier le problème en commençant par consulter des ressources au sujet de Docker, Jenkins et Azure et programmer des réunions avec des responsables de la plateforme **Code P**. J'ai pu ensuite construire un fichier Jenkins à mettre à la racine du projet sur le GitLab. ([annexe 10](#))

Je n'avais plus qu'à aller sur le compte Jenkins pour lancer le déploiement de l'application.



## 10. Tests

Avec l'aide du testeur de l'équipe de la Squad 3, j'ai préparé des plans de tests à exécuter à la phase de validation, avant la recette, pour répondre directement au cahier des charges fonctionnel.

Je les ai mis en place grâce à JIRA en passant dans le kanban de conception de test.

Tableau Kanban

The screenshot shows a JIRA Kanban board with the following columns and data:

- A FAIRE 0**: Empty column.
- EN COURS 0**: Empty column.
- A VALIDER 5**:
  - TV-1 : Modification de l'url associée au clic-bouton "contacter un expert de proximité" (PR-12)
  - TV-1 : [BO HUB] Ajout d'une carte non-cliqueable "Usagers" et de son bouton d'extraction des usagers du Hub (PR-16)
  - TV-1 : [BO HUB] ajout des porteurs de projets dans la fiche référents - modification du tableau (PR-20)
  - TV-1 : [BO HUB] ajout des porteurs de projets dans la fiche référents - bouton « affecter à un autre référent » (PR-21)
  - TV-1 : [BO HUB] Fonctionnalité d'extraction des dispositifs (PR-25)
- TERMINÉ 0**:
  - Nous n'a

Par exemple j'ai créé le test **TV-1** concernant l'évolution **PR-1** en détaillant les étapes à effectuer (Action, Données, Résultat Attendu).

**TV-1 : Modification de l'url associée au clic-bouton "contacter un expert de proximité"**

**Généralités** Qualification Ticket

**Description**

Description :

Ce test a pour but de vérifier :

- qu'au survole du bouton "contacter un expert de proximité" la bonne url s'affiche en bas à gauche de la page ;
- qu'au clique sur le bouton "contacter un expert de proximité" un onglet est ouvert redirigeant vers la bonne url.

Criticité : 3

Contexte :

A tester sur environnement : site public ( HubEntreprendre)

Ce test convient pour tous les types d'utilisateurs

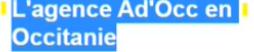
Prérequis :

accès au site public ( HubEntreprendre)

Puis j'ai conçu un plan de test et une exécution de test en liant le test TV-1

Segment	Clé	Résumé	Type de Test	Jeu de Données	#Défauts	Statut	Actions
PR-	12	TV-1 : Modification de l'url associée au clic-bouton "co...	Manual		0	TO DO	...

En appuyant sur le bouton « play », l'exécution se lance et on revoit les étapes constituant le test. On peut alors signifier si le test est valide et éventuellement mettre des commentaires. S'il y a un default, on peut créer un Jira pour le résoudre qui sera à son tour fermé après résolution.

1	Action   Pièces jointes (2) Se rendre sur l'url du site public : <a href="https://hubentreprendre.laregion.fr/">https://hubentreprendre.laregion.fr/</a>	Données 	Résultat Attendu Le site <a href="https://hubentreprendre.laregion.fr/">https://hubentreprendre.laregion.fr/</a> s'affiche correctement.
2	Action   Pièces jointes (2) Se rendre en bas de page. et survoler dans un premier temps puis cliquer dans la section "Vous avez des questions ? - L'agence Ad'Occ en Occitanie" sur le bouton "Contacter un expert de proximité".	Données  L'agence Ad'Occ accompagne toutes les entreprises en Occitanie. Son objectif : être à vos côtés pour vous	Résultat Attendu Au survole du bouton "Contacter un expert de proximité", l'url <a href="https://www.agence-adocc.com/agence-adocc/trouver-adocc/">https://www.agence-adocc.com/agence-adocc/trouver-adocc/</a> s'affiche en bas à gauche de la page. Au clique sur le bouton "Contacter un expert de proximité" un nouvel onglet s'ouvre bien sur la page <a href="https://www.agence-adocc.com/agence-adocc/trouver-adocc/">https://www.agence-adocc.com/agence-adocc/trouver-adocc/</a>

Si toutes les étapes sont passées, le test est validé. Si tous les tests sont validés, l'amélioration est prête à être livré en recette.

Certains tests peuvent devenir des **tests de non régression** (TNR) pouvant être utilisés si d'autres améliorations sont apportées plus tard au même domaine.

Tous ces plans de test et leurs exécutions sont traçables et accessibles par le client s'il en fait la demande.

## 11. Bilan du projet

Les objectifs fixés initialement n'ont pas tous été atteints et il a été reconnu par mes responsables que de nombreuses contraintes non anticipées m'ont empêché d'y parvenir.

La dernière semaine de mon stage, j'ai fait un transfert de connaissance du projet à deux autres développeurs de mon équipe qui ont pris en charge le reste des évolutions à faire. On a pu valider ensemble le travail que j'avais effectué et échanger pour définir la stratégie pour livrer toutes les améliorations à la date prévue.

Au terme de ce stage, j'ai réussi à prendre en main les différentes technologies du périmètre pour la réalisation du projet, même si au départ ça m'a demandé beaucoup de temps.

## 12. Bilan personnel et perspectives

Les 3 mois que j'ai passé en stage au sein d'Inetum se sont écoulés rapidement de mon point de vue. J'ai particulièrement apprécié la collaboration et l'esprit de groupe qui régnait dans mon équipe. Je suis satisfait d'avoir mené à bien une partie des tâches qui m'ont été confiées.

Cette première expérience m'a permis de mobiliser l'ensemble des compétences acquises durant ma formation, de les approfondir, de gagner en autonomie, d'élargir mes compétences en abordant des technologies telles que docker, jenkins ou la plateforme azure.

Cette expérience m'a aussi permis de découvrir le monde du développement dans une entreprise dédiée au numérique et d'interagir avec les acteurs du projet, responsable de service, chef de projet et équipes de développement. Tous ensemble, ils ont enrichi ma vision de ce milieu, m'ont convaincu de l'importance d'une bonne communication et su me confirmer mon goût pour le développement d'applications.

Enfin, la fin de mon stage s'est soldée par un retour positif de la part de mes responsables et de mon équipe. J'ai eu une proposition de CDI que j'ai acceptée.

C'est donc confiant que j'aborde la suite pour progresser jusqu'à l'autonomie complète.

# 13. Annexes

## 13.1. Tableau Kanban

Tableau Kanban

The screenshot shows a Jira Kanban board with the following columns and their contents:

- BACKLOG 1**: Contains one ticket: "Décommissionnement ancienne architecture LRC" (PR-3).
- A ANALYSER 2**: Contains two tickets: "Mise à jour de la version API : Version 3" (PR-9) and "Intégration tag de suivi BPI" (PR-11).
- EN ANALYSE 1**: Contains two tickets: "[HUB] Amélioration du référencement du site" (PR-4) and "[BO HUB] ajout des porteurs de projets dans la fiche référents" (PR-6).
- A APPROUVER 0**: Contains zero tickets.
- EN ATTENTE 0**: Contains zero tickets.
- APPROUVÉ 3**: Contains three tickets: "[BO HUB] Fonctionnalité d'extraction des dispositifs" (PR-7), "[JIRA mère pour PR-1, 5, 6, 7 et 8 : lot pour recette commune" (PR-10), and "[HUB] Extraction des usagers du Hub" (PR-5).
- EN COURS DE RÉALISATI... 4**: Contains two tickets: "Solution de contournement du format de téléphone" (PR-2) and "[HUB] - mise à jour flux BPI "dossiers projecteurs"" (PR-8).
- FERMÉ 0**: Contains zero tickets.

A ticket in the **APPROUVÉ 3** column, specifically the one titled "[JIRA mère pour PR-1, 5, 6, 7 et 8 : lot pour recette commune]", is highlighted with a red rectangular box.

## 13.2. Requêtes SQL pour résolution de ticket sur CARL

```
SOL0189466 clôturé le 01/06/2022
Je vous prie de bien vouloir trouver ci-joint 2 redressements de données à opérer dans CARL.
Num DA Etat actuel Etat souhaité Com LM
DA-036869 Imputée sur 2023 A imputer sur 2021 Mme VERY Rascol
DA-065089 Imputée sur budget

1102_DMOE A imputer sur budget 3421_DMOE

DA-036869 OK
/*modification du période d'imputation budget de la DA */
update cspu_budgacmvnt set mvdate='2021-12-31' where prline_id=(select id from cspu_prline where pr_id=(select id from cspu_pr where code ='DA-036869')));
DA-065089 OK

/*modification du budget pour DA*/
update cspu_pr set budgacc_id=(select id from csfi_budgacc where code = '3421_DMOE') where code = 'DA-065089';
/*modification du budget pour lignes de DA*/
update cspu_prline set budgacc_id=(select id from csfi_budgacc where code = '3421_DMOE') where pr_id=(select id from cspu_pr where code ='DA-065089');
/*modification du budget pour écritures budgétaires de LDA*/
update csfi_budgacmvnt set budgacc_id=(select id from csfi_budgacc where code = '3421_DMOE') where prline_id=(select id from cspu_prline where pr_id=(select id from cspu_pr where code ='DA-065089'));

SOL0184888 ok clôturé le 01/06/2022
Vous trouverez ci-joint la liste des redressements de données à réaliser sur Carl Source

DA Commande Budget Initial Budget final A réaliser sur le Budget (Budgacc) lié à la DA visée
DA-063504 CMD-006948 1102_DMOE 6613_DMOE Supprimer les lignes d'écriture sur le 1102_DMOE
et les réaffecter sur le 6613_DMOE
et le repositionner sur la DA visée
DA-063504 CMD-006948 1102_DMOE 6601_DMOE Supprimer les lignes d'écriture sur le 1102_DMOE
et les réaffecter sur le 6601_DMOE
et le repositionner sur la DA visée
DA-064230 CMD-006892 1102_DMOE 3410_DMOE Supprimer les lignes d'écriture sur le 1102_DMOE
et les réaffecter sur le 3410_DMOE
et le repositionner sur la DA visée
DA-063050 CMD-006872 1102_DMOE 3411_DMOE Supprimer les lignes d'écriture sur le 1102_DMOE
et les réaffecter sur le 3411_DMOE
et le repositionner sur la DA visée

DA-063504 OK
/*modification du budget pour DA*/
update cspu_pr set budgacc_id=(select id from csfi_budgacc where code = '6613_DMOE') where code = 'DA-063504';
/*modification du budget pour lignes de DA*/
update cspu_prline set budgacc_id=(select id from csfi_budgacc where code = '6613_DMOE') where pr_id=(select id from cspu_pr where code ='DA-063504');
/*modification du budget pour écritures budgétaires de LDA*/
update csfi_budgacmvnt set budgacc_id=(select id from csfi_budgacc where code = '6613_DMOE') where prline_id=(select id from cspu_prline where pr_id=(select id from cspu_pr where code ='DA-063504'));

DA-064274 OK
/*modification du budget pour DA*/
update cspu_pr set budgacc_id=(select id from csfi_budgacc where code = '6613_DMOE') where code = 'DA-064274';
/*modification du budget pour lignes de DA*/
update cspu_prline set budgacc_id=(select id from csfi_budgacc where code = '6613_DMOE') where pr_id=(select id from cspu_pr where code ='DA-064274');
/*modification du budget pour écritures budgétaires de LDA*/
update csfi_budgacmvnt set budgacc_id=(select id from csfi_budgacc where code = '6613_DMOE') where prline_id=(select id from cspu_prline where pr_id=(select id from cspu_pr where code ='DA-064274'));

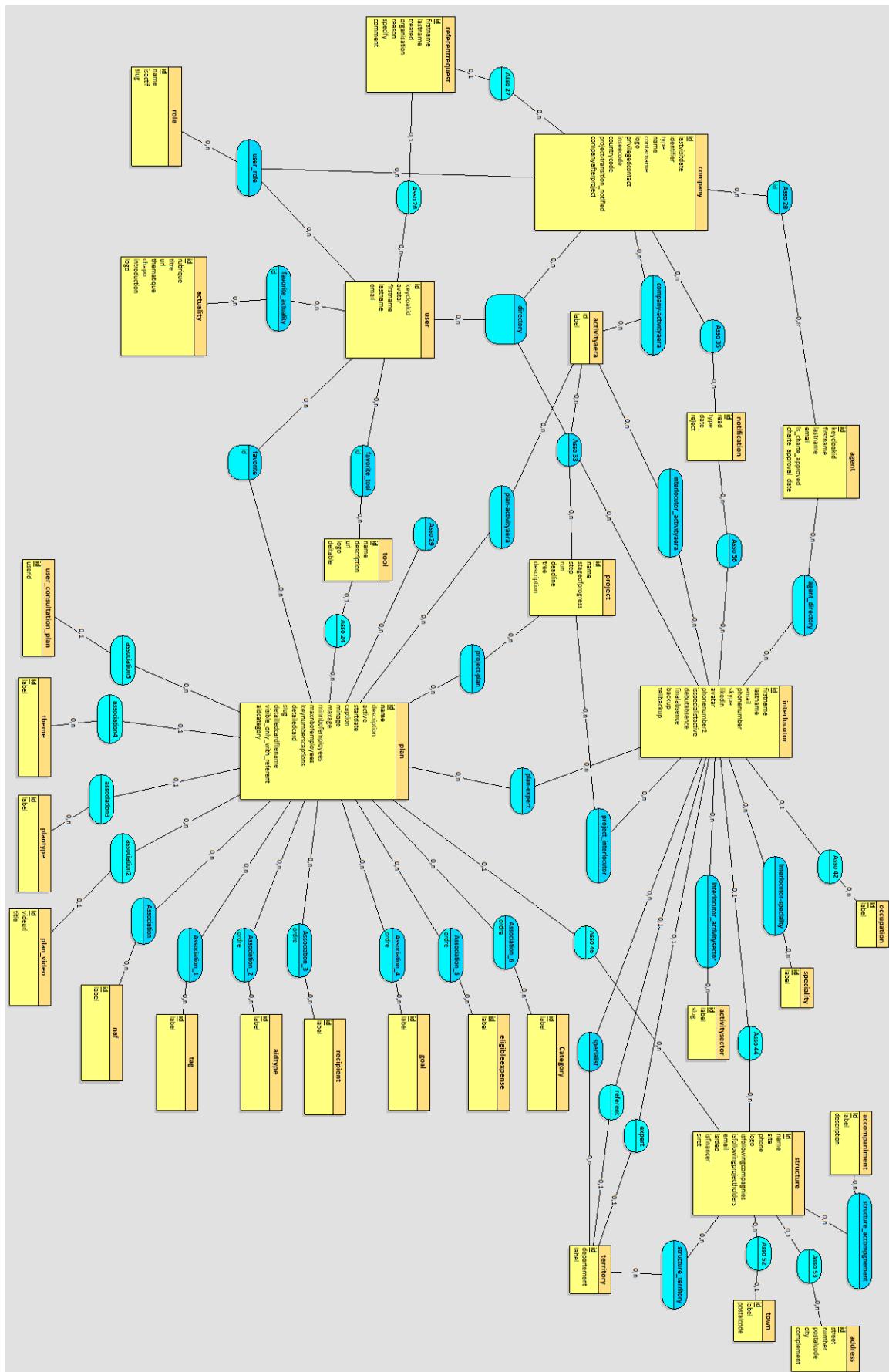
DA-064230 OK
/*modification du budget pour DA*/
update cspu_pr set budgacc_id=(select id from csfi_budgacc where code = '3410_DMOE') where code = 'DA-064230';
/*modification du budget pour lignes de DA*/
update cspu_prline set budgacc_id=(select id from csfi_budgacc where code = '3410_DMOE') where pr_id=(select id from cspu_pr where code ='DA-064230');
/*modification du budget pour écritures budgétaires de LDA*/
update csfi_budgacmvnt set budgacc_id=(select id from csfi_budgacc where code = '3410_DMOE') where prline_id=(select id from cspu_prline where pr_id=(select id from cspu_pr where code ='DA-064230'));

DA-063050 OK
/*modification du budget pour DA*/
update cspu_pr set budgacc_id=(select id from csfi_budgacc where code = '3411_DMOE') where code = 'DA-063050';
/*modification du budget pour lignes de DA*/
update cspu_prline set budgacc_id=(select id from csfi_budgacc where code = '3411_DMOE') where pr_id=(select id from cspu_pr where code ='DA-063050');
/*modification du budget pour écritures budgétaires de LDA*/
update csfi_budgacmvnt set budgacc_id=(select id from csfi.budgacc where code = '3411_DMOE') where prline_id=(select id from cspu_prline where pr_id=(select id from cspu_pr where code ='DA-063050'));
```

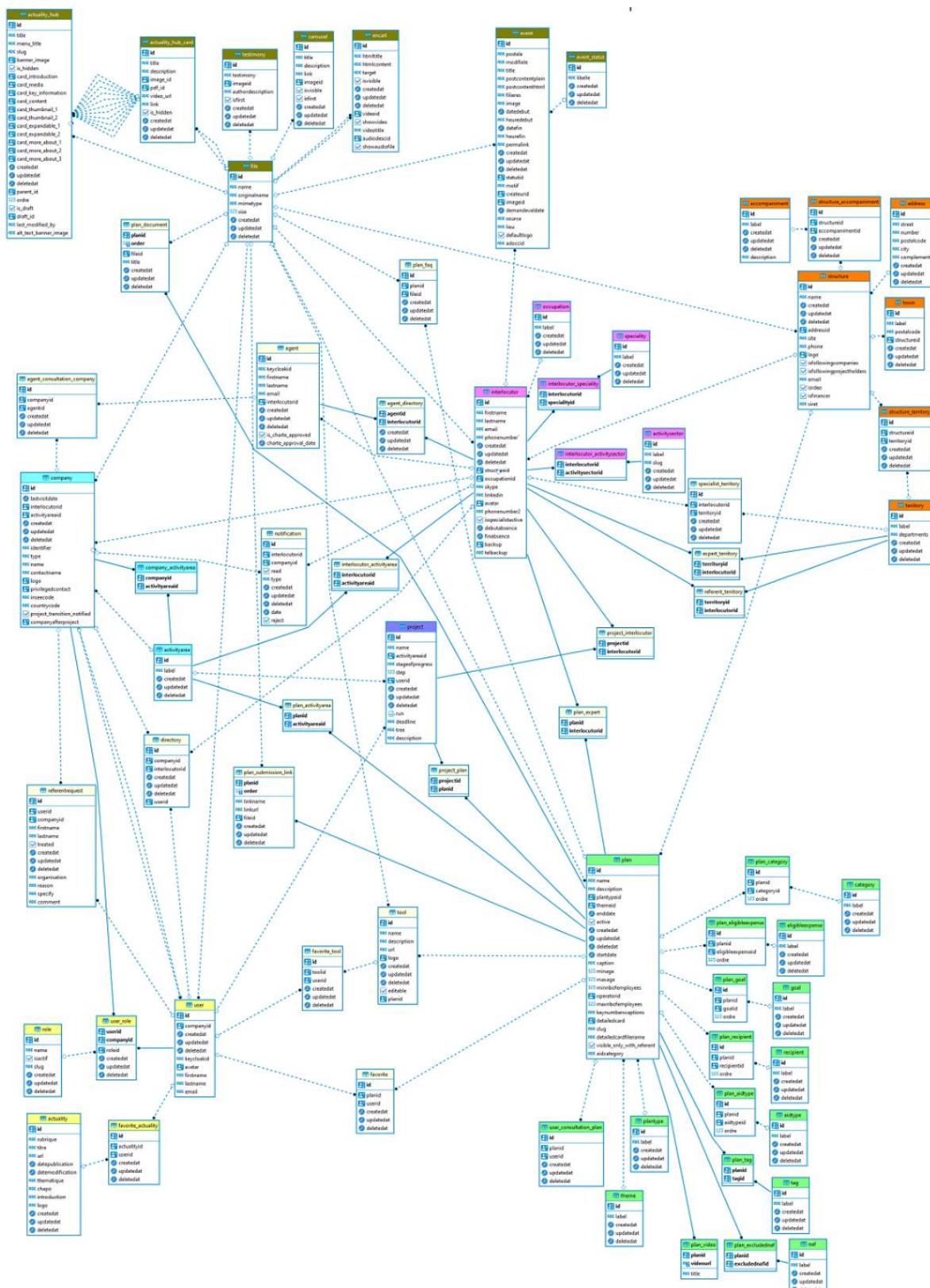
### 13.3. Dictionnaire des données

NOM	DESCRIPTION	TYPE	null/non	Taille	IDENTIFIANT	MODE	REGLE DE D'OBTENTION	CALCUL	DOMAINE DE VALEURS
<b>Hub Entreprendre</b>									
<b>User</b>									
<b>id</b>		UUID	non Null		X	Mémorisé			
Firstname		ALPHA	Null	255		Mémorisé			
Lastname		ALPHA	non Null	255		Mémorisé			
companyid		UUID	Null			Mémorisé			
email		ALPHANUM	Null	255		Mémorisé			
<b>Role</b>									
<b>Id</b>		UUID	non Null		X	Mémorisé			
name		ALPHA	non Null	255		Mémorisé			
slug		NUM	non Null	50		Mémorisé			
isactif		ALPHANUM	non Null	255		Mémorisé			
<b>Actuality</b>									
<b>id</b>		UUID	non Null		X	Mémorisé			
rubrique		ALPHANUM	non Null	255		Mémorisé			
titre		ALPHANUM	non Null	255		Mémorisé			
url		ALPHANUM	non Null	255		Mémorisé			
thematique		ALPHA	non Null	255		Mémorisé			
chapo		ALPHA	non Null	255		Mémorisé			
introduction		ALPHA	non Null	255		Mémorisé			
logo		ALPHANUM	non Null	255		Mémorisé			
datepublication		DATE	non Null			Mémorisé			
datemodification		DATE	non Null			Mémorisé			
<b>Plan</b>									
<b>id</b>		UUID	non Null		X	Mémorisé			
name		ALPHA	non Null	255		Mémorisé			
description		ALPHA	Null	255		Mémorisé			
enddate		DATE	Null			Mémorisé			
active		BOOLEAN	non Null			Mémorisé			
startdate		DATE	non Null			Mémorisé			
caption		ALPHA	Null	255		Mémorisé			
minage		NUM	Null			Mémorisé			
maxage		NUM	Null			Mémorisé			
minnbofemployees		NUM	Null			Mémorisé			
maxnbofemployees		DATE	Null			Mémorisé			
keynumberscaptions		ALPHA	non Null	255		Mémorisé			
detailedcardfilename		ALPHANUM	Null	255		Mémorisé			
visible_only_with-referent		BOOLEAN	Null			Mémorisé			
aidcategory		ALPHA	Null	255		Mémorisé			
slug		ALPHA	non Null	250		Mémorisé			
<b>Theme</b>									
<b>id</b>		UUID	non Null		X	Mémorisé			
Label		ALPHA	non Null	255		Mémorisé			

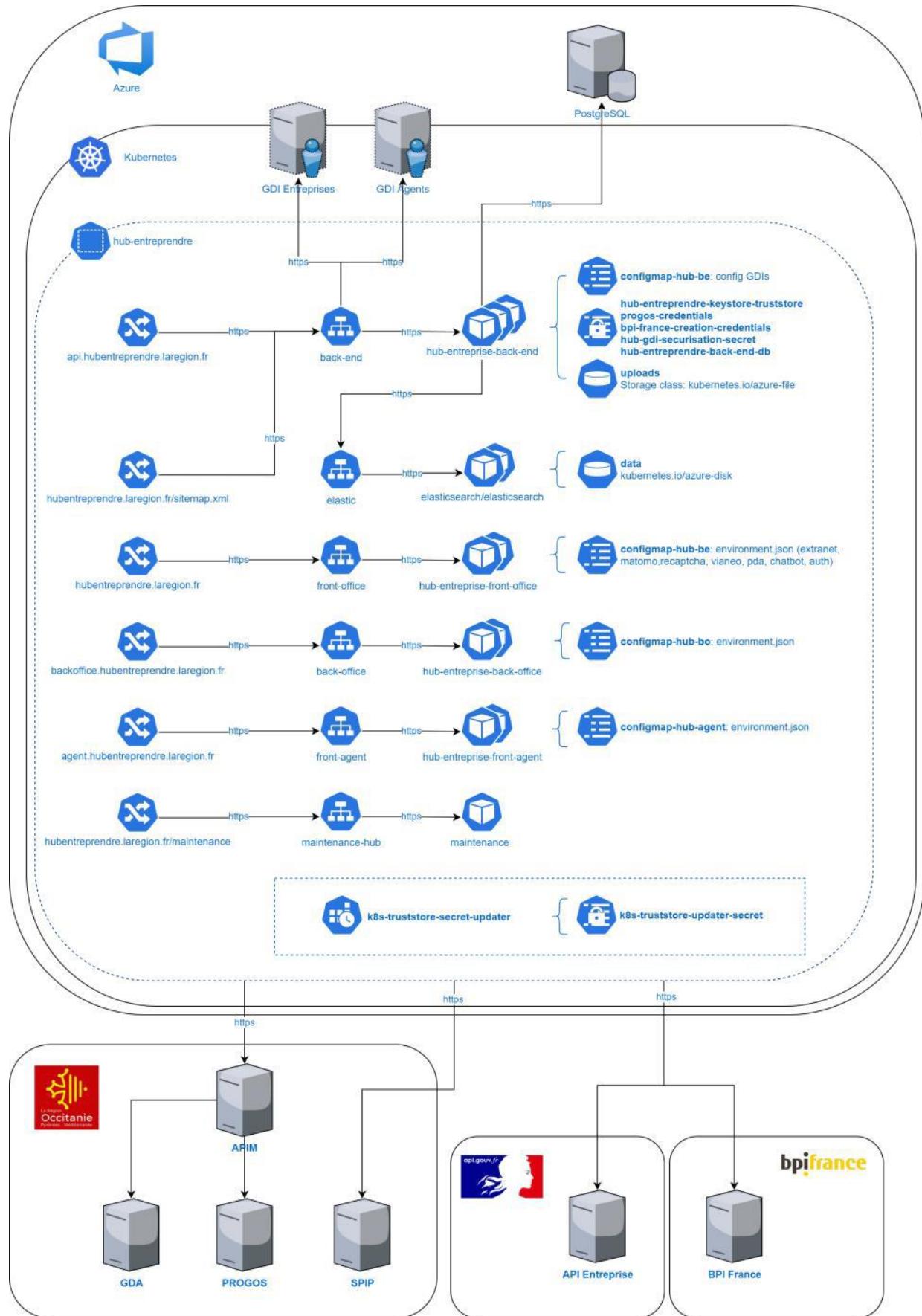
## 13.4. Modèle conceptuel des données - MCD



## 13.5. Modèle logique des données - MLD



## 13.6. Schéma d'architecture détaillé de Hub-Entreprendre



## 13.7. Fichier fr.json du Front-Entreprise

```

"contact-cards": [
    {
        "circle": "Vous avez une question ?",
        "subtitle": "Différentes options s'offrent à vous",
        "label-phone-number": "Service gratuit + prix d'un appel",
        "label2-phone-number": "Service et appel gratuit",
        "card-1": {
            "title": "La Région Occitanie",
            "content": "Des conseillers vous délivrent un premier niveau d'informations et vous orientent vers le bon interlocuteur.",
            "phoneNumber": "0 800 31 31 01",
            "buttonLabel": "",
            "buttonLink": ""
        },
        "card-4": {
            "title": "L'agence Ad'Occ en Occitanie",
            "content": "L'agence Ad'Occ accompagne toutes les entreprises en Occitanie. Son objectif : être à vos côtés pour vous soutenir et développer votre activité.",
            "phoneNumber": "",
            "buttonLabel": "Contacter un expert de proximité",
            "buttonLink": "https://www.agence-adocc.com/agence-adocc/trouver-adocc"
        }
    }
]

```

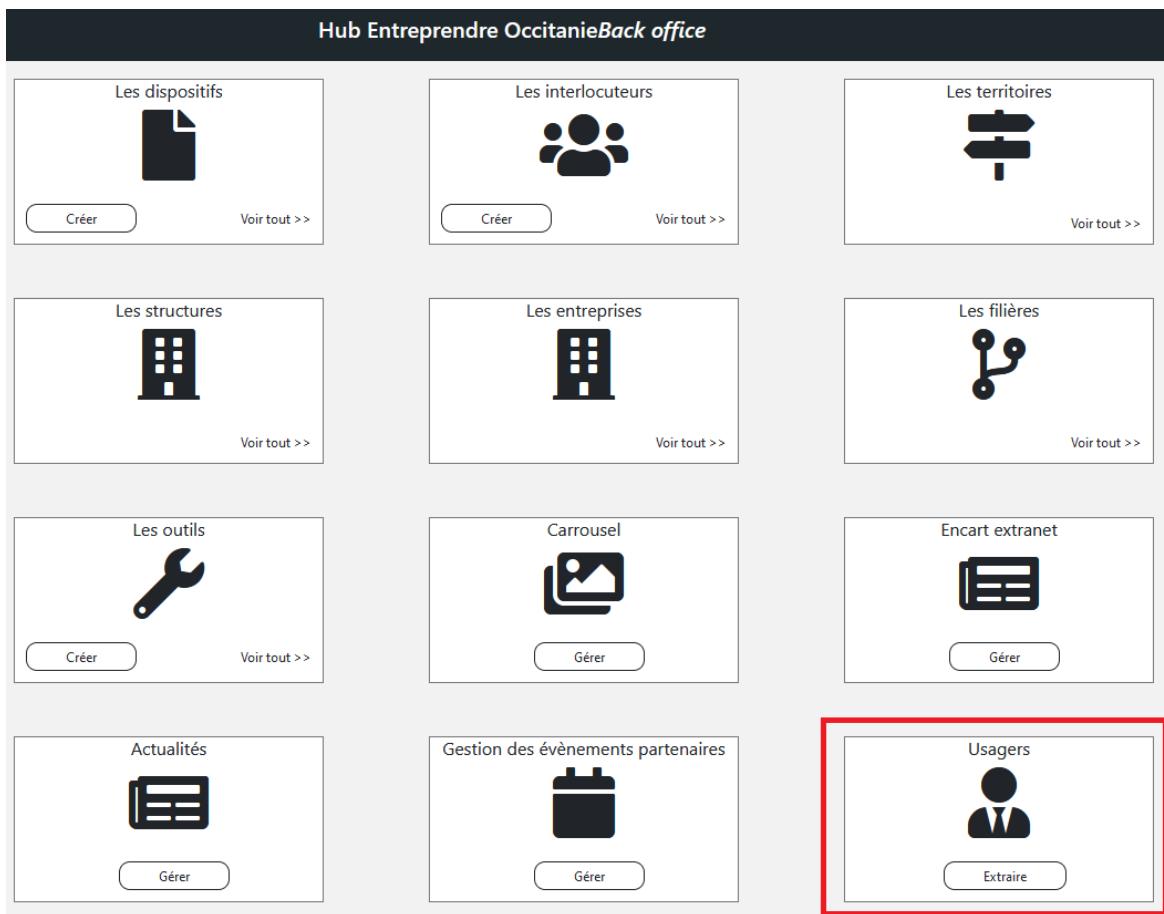
## 13.8. Requête de création de la Vue pour les Extraction des Dispositifs

```

CREATE OR REPLACE VIEW plansExtractView AS
select p.id, p.name, p.aidcategory as catégorie,
regexp_replace(regexp_replace(regexp_replace(p.caption, E'[\n\r]+', ' ', 'g'), ' ', ' ', 'g'), E'<[^>+>', ' ', 'gi') as phrase_acroche,
regexp_replace(p.keynumberscaptions, E'[\n\r]+', ' ', 'g') as chiffres_clés,
p.active, p2."label" as type",
array_to_string(array(select a2."label" from activityarea a2
left join plan_activityarea pa on a2.id = pa.activityareaid
where pa.planid = p.id), ' / ') as filières,
array_to_string(array(select c2."label" from category c2
left join plan_category pc on c2.id = pc.categoryid
where pc.planid = p.id), ' / ') as catégories_de_projet,
array_to_string(array(select e2."label" from eligibleexpense e2
left join plan_eligibleexpense pe on e2.id = pe.eligibleexpenseid
where pe.planid = p.id), ' / ') as dépenses_eligibles,
array_to_string(array(select regexp_replace(g2."label", E'[\n\r]+', ' ', 'g') from goal g2
left join plan_goal pg on g2.id = pg.goalid
where pg.planid = p.id), ' / ') as objectifs,
array_to_string(array(select r2."label" from recipient r2
left join plan_recipient pr on r2.id = pr.recipientid
where pr.planid = p.id), ' / ') as bénéficiaires,
array_to_string(array(select a3."label" from aidtype a3
left join plan_aidtype pa3 on a3.id = pa3.aidtypeid
where pa3.planid = p.id), ' / ') as nature_aide,
array_to_string(array(select t2."label" from tag t2
left join plan_tag pt on t2.id = pt.tagid
where pt.planid = p.id), ' / ') as tags,
array_to_string(array(select n2."label" from naf n2
left join plan_excludednaf pe2 on n2.id = pe2.excludednafid
where pe2.planid = p.id), ' / ') as naf_exclus,
p.createdat, p.updatedat, p.startdate, p.enddate, s."name" as opérateur,
array_to_string(array(select concat(i2.firstname, ' ', i2.lastname) from interlocutor i2
left join plan_expert pe3 on i2.id = pe3.interlocutorid
where pe3.planid = p.id), ' / ') as experts,
concat(coalesce(f2.originalname, f2."name"), ' ', p.detailedcardfilename) as "Lien fiche descriptive (pdf) (nom du document + nom pour l'affichage s'il y en a un spécifique)",
array_to_string(array(select COALESCE (f.originalname, f."name") from file f
left join plan_document pd on f.id = pd.fileid
where pd.planid = p.id), ' / ') as documents,
array_to_string(array(select COALESCE (f.originalname, f."name") from file f
left join plan_faq pf on f.id = pf.fileid
where pf.planid = p.id), ' / ') as "Documents visibles uniquement sur le portail Agents (Lien vers la FAQ, PDF)",
array_to_string(array(select COALESCE (f.originalname, f."name") from file f
left join plan_submission_link ps1 on f.id = ps1.fileid
where ps1.planid = p.id), ' / ') as "Liens de dépôt de dossier PDF (+ nom du lien s'il existe dans la colonne suivante, séparés par des slashes)",
array_to_string(array(select concat(ps1.linkurl, ' ', ps1.linkname) from plan_submission_link ps1
where ps1.planid = p.id), ' / ') as "Liens de dépôt de dossier URL (URL + nom du lien s'il existe, séparés par des slashes)",
array_to_string(array(select concat(pv.title, ' : ', pv.videourl) from plan_video pv
where pv.planid = p.id), ' / ') as "Vidéos (nom de la vidéo : URL de la vidéo, séparés par des slashes)",
p.mimage, p.maxage, p.minnbemployees, p.maxnbemployees,
CONCAT('https://hubentreprendre.laregion.fr/financement/', p.slug) as "Lien URL de la page"
from plan P
left join plantype p2 on p2.id = p.plantypeid
left join "structure" s on s.id = p.operatorid
left join file f2 on f2.id = p.detailedcard
group by p.id, p.name, p.description, phrase_acroche, chiffres_clés, p.active, "type",
"lien fiche descriptive (pdf) (nom du document + nom pour l'affichage s'il y en a un spécifique)", p.createdat, p.updatedat, p.startdate, p.enddate, opérateur,
p.mimage, p.maxage, p.minnbemployees, p.maxnbemployees, "lien URL de la page"
order by p.name asc

```

## 13.9. IHM Extraction de usagers



## 13.10. fichier Jenkins pour déploiement sur plateforme azure

```
@Library('codep-shared-library') _
pipeline {
    agent none

    environment {
        URL_REPO_NEXUS= 'https://nexus.coder.inetum.world/repository'
        CREDENTIALS_ID_NEXUS = 'XXXXXXXXXXXXXX'
        HOME = '.'
        URL_AZURE_REGISTRY = 'XXXXXXXXXXXXXX'
        CREDENTIALS_AZURE_REGISTRY_ID = 'XXXXXXXXXXXXXX'
        NOM_IMAGE = "hub-entreprendre/hub-entreprise-backend"
        VERSION = "latest"
        WORK_DOCKERFILE = "."
    }

    stages {
        stage('Build projet') {
            agent {
                docker {
                    label 'slave-linux'
                    image 'di-codep/codep-java8-mvn-nodejs10:latest'
                    args ' -v $DATA_CACERTS:$CACERTS_URL -i --entrypoint='
                    registryUrl 'XXXXXXXXXXXXXX'
                    registryCredentialsId 'XXXXXXXXXXXXXX'
                    reuseNode true
                }
            }
            steps {
                withCredentials([usernamePassword(credentialsId: "${CREDENTIALS_ID_NEXUS}", usernameVariable: 'USERNAME_VAR', passwordVariable: 'PASSWORD_VAR')]) {
                    sh '''
                        mvn package -Pprod -DskipTests -Phexus,private-auth,prod -Dmaven.repo.user='${USERNAME_VAR}' -Dmaven.repo.password='${PASSWORD_VAR}' -Dmaven.repo.remote='${URL_REPO_NEXUS}'
                    '''
                }
            }
        }

        stage('Deploy Docker Image to Azure') {
            agent {
                label 'slave-linux'
            }
            steps {
                buildDockerImage("${URL_AZURE_REGISTRY}", "${CREDENTIALS_AZURE_REGISTRY_ID}", "${NOM_IMAGE}", "${VERSION}", "${WORK_DOCKERFILE}")
            }
        }
    }
}
```