

Germain OLEA-OYUGOU

*1<sup>ère</sup> année cycle ingénieur*

*Système d'Information*

2020 – 2021

EiCNAM

le cnam  
Grand Est

# Révolutionner la e-santé avec ReLyfe

2020 – 2021

Rapport de la période d'apprentissage

Maître d'apprentissage : Ambroise Recht

Tuteur enseignant : Jason Mahdjoub

40 Rue du Colisée

75008 Paris

France

RELYFE R



## **Remerciements**

Premièrement je voudrais adresser mes remerciements à Valérie Chevalarias pour avoir poussé ma candidature auprès de ReLyfe et à Magedouline N'Gairi pour l'avoir accepté et m'autoriser ainsi à réaliser mon apprentissage à leur côté.

Je tiens particulièrement à remercier Dany Yacoub, qui m'aide à progresser en m'épaulant continuellement lorsque j'ai des difficultés, ainsi que toute l'équipe technique avec qui je travaille au quotidien pour leur professionnalisme, leur amitié, leur accueil et les connaissances qu'ils ont pu m'apporter directement ou indirectement.

Merci également à mon maître d'apprentissage Ambroise Recht pour sa présence et son partage d'expertise.

Et pour finir, je remercie tout le corps enseignant du CNAM et spécialement mon tuteur Jason Mahdjoub pour la qualité de ses cours et son application dans mon suivi.

## Glossaire

### Merge Request (MR)

Lorsque deux branches d'un dépôt git ont divergé et que l'une est en avance sur l'autre, on crée une merge request pour les remettre au même niveau.

### DevOps

Le DevOps est une approche d'ingénierie informatique qui permet d'unifier les méthodes de développement logiciel et celles de déploiement et d'administration d'infrastructures.

### API REST

REST, pour Representational State Transfer Application Program Interface, est une norme architecturale, un ensemble de principe de développement d'APIs.

### Docker

Docker est une technologie de virtualisation qui permet d'exécuter un système de fichier dans une image virtuelle appelée conteneur, cette technologie permet de lancer une application dans un conteneur de façon simple et rapide.

## Table des figures

Figure 1 : Cœur-Code .....	4
Figure 2 : Carte ReLyfe .....	5
Figure 3 : Organigramme équipe technique .....	6
Figure 4 : Organigramme équipe design .....	7
Figure 5 : Processus DevOps.....	11
Figure 6 : ReLyfe API services .....	12
Figure 7 : Derniers tickets traités .....	16
Figure 8 : Graphique tableaux Jira .....	17
Figure 9 : Jira ReLyfe Developer .....	i
Figure 10 : Onglet développement d'un ticket Jira .....	ii
Figure 11 : Architecture ReLyfe .....	ii

## Table des matières

---

Remerciements .....	i
Glossaire .....	ii
Table des figures .....	ii
Introduction .....	1
1. Présentation Entreprise .....	2
1.1. L'accès aux données dans la santé .....	2
1.2. InnovHealth & PassCare .....	3
1.2.1. Objectifs et Fonctionnement de PassCare .....	3
1.3. Ambitions de InnovHealth .....	4
1.4. ReLyfe .....	5
1.5. Recherche et Développement .....	5
2. Environnement de travail .....	6
2.1. Equipe .....	6
2.2. Gestion de projet .....	8
2.3. Outils .....	10
2.3.1. Jira & Confluence .....	10
2.3.2. GitLab .....	10
2.3.3. Teams (Office 365) .....	11
2.4. Architecture & Outils Dev .....	12
3. Accomplissements .....	13
3.1. Développement .....	13
3.2. Administration .....	16
4. Projection tâches futures .....	18
Conclusion .....	20
Annexes .....	i

## Introduction

Je suis passionné de technologies, mais surtout avare de connaissances. L'informatique est un domaine presque aussi vaste que la plage d'adresse IPv6, une plage qui ne demande qu'à être explorée.

À la suite de mon DUT Informatique à l'IUT de Reims, l'immensité des possibilités de l'univers lié à l'informatique me parut encore plus accrue. Ayant toujours voulu poursuivre mes études, cette envie se fit de plus en plus ressentir, renforcée par mon enthousiasme à découvrir de nouvelles choses. Avec plus de connaissances sur les domaines de l'informatique, acquises grâce à mon DUT, j'ai su que je voulais en apprendre plus tout en approfondissant les compétences déjà assimilées notamment en développement web. S'est donc présenter le choix de l'alternance avec la formation d'ingénieur en système d'information du CNAM dont

L'objectif est de former, en trois ans, des ingénieurs généralistes en informatique avec une dominante en systèmes d'information.<sup>1</sup>

Lors de ma recherche d'entreprise, j'ai contacté une start-up rémoise, présentée lors d'un cours par ma professeure de communication, madame Valérie Chevalarias, qui pris la peine de soutenir ma candidature en tant que développeur back end pour renforcer l'équipe grandissante de InnovHealth. Ces derniers ont misé sur ma détermination à évoluer dans mon domaine plus qu'à mon manque d'expériences professionnelles dans l'optique d'en voir les fruits bénéficier à l'activité de la structure. Ce fut donc le début d'une alternance orienté sur une confiance mutuelle entre les deux parties.

Pour en détailler les bénéfices nous commencerons par une présentation de mon entreprise d'accueil avant d'aborder mon environnement de travail et la suite de technologies que nous utilisons. Je parlerai ensuite de mes réalisations au cours de cette première année puis pour terminer nous essayerons d'envisager les potentielles responsabilités qui me seront attribuées dans mes futures années d'alternance aux côtés de ReLyfe.

---

<sup>1</sup> [CNAM Grand-Est: FIP INFO](#)

## 1. Présentation Entreprise

---

### 1.1. L'accès aux données dans la santé

En santé quand on parle de données on fait principalement allusion à tous les documents, papiers pour la plupart, que les professionnels de santé ou les hôpitaux que nous fréquentons nous fournissent. Cela peut être une ordonnance, une image radiographique, des résultats d'analyse, un certificat, etc. Ces derniers sont conservés par les deux parties, le professionnel de santé par le biais de son lieu d'activité – avant de terminer, « pseudonymisées », auprès du Système national des données de santé (SNDS) – ou le patient par ses propres moyens. Les données tirées de ces documents peuvent servir à différentes fins parmi lesquelles :

- La surveillance, la veille et la sécurité sanitaires,
- La recherche et les études
- Sur le long terme elles servent à avoir une évolution sur la santé générale du patient, pour de la prévention notamment

Cependant l'accès à ces données, dans le cas de la SNDS (instauré depuis le 26 janvier 2016 – voir annexe), est très réglementé et conditionné :

Tous les acteurs publics et privés peuvent accéder aux données du SNDS.

[...]

Un accès aux données à caractère personnel du SNDS ne peut être autorisé que pour permettre des traitements à des fins de recherche, d'étude ou d'évaluation répondant à un motif d'intérêt public et contribuant à l'une des finalités énumérées dans la loi.<sup>2</sup>

Etant donc très contrôlé, l'accès aux données de santé n'est pas une mince affaire, bien que les laisser circuler librement ne soit pas une solution mais les stocker indéfiniment non plus car cela ralentit le traitement de ces dernières à l'échelle de l'individu (le patient) et de la population, par conséquent les avancées de la science elle-même.

---

<sup>2</sup> [SNDS : accès aux données](#)

Pour citer le Dr. Adnan El Bakri :

La France [a] le meilleur système de soins au monde, pas le meilleur système de santé. C'est-à-dire qu'on vous traite parfaitement si vous tombez malade, mais la gestion en amont et en aval de la maladie est très complexe, alors que c'est pourtant là que se concentre une grande partie de dépenses inutiles et évitables.

On en conclue donc très rapidement que

L'accès aux soins et à la santé doit passer d'abord par le partage des données, l'information médicale personnalisée, la prévention, le dépistage et une éducation thérapeutique adaptée.<sup>3</sup>

D'où la naissance de **PassCare**.

## 1.2. **InnovHealth & PassCare**

PassCare, produit maître de InnovHealth promet avant tout de numériser les données de santé pour faciliter le parcours de soin du patient en autorisant le partage d'informations médicales avec son ou ses professionnels de santé, permet d'interconnecter ces professionnels par la suite et emmène à un autre niveau la recherche médicale en donnant la possibilité à tous ses utilisateurs de partager, de façon totalement anonyme, leurs données médicales aux équipes de recherche. Tout cela dans le but d'améliorer le réseau de santé tout en répondant plus facilement aux attentes des patients.

### 1.2.1. **Objectifs et Fonctionnement de PassCare**

Le PassCare existe pour redonner au patient les pleins droits sur ses données de santé, et cela avec 3 principaux objectifs : récupérer, structurer, et partager leurs informations avec les professionnels médicaux et paramédicaux de *leur* choix.

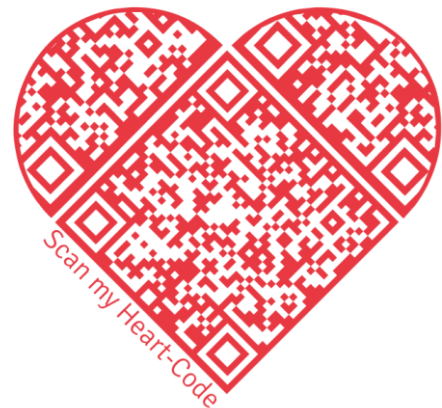
---

<sup>3</sup> [Adexis Solution - Adnan El Bakri](#)



Chaque PassCare possède une empreinte unique que nous appelons « Cœur-Code », un QR code en forme de cœur, avec une clé personnelle permettant d'identifier et d'accéder à un PassCare de partout dans le monde.

PassCare est donc une plateforme, web d'abord, où tout professionnel de santé peut remplir des informations liées à la santé du patient, téléversées des documents et plus encore.



*Figure 1 : Cœur-Code*

### 1.3. Ambitions de InnovHealth

Devenir leader mondial [en santé publique] et créer une nouvelle ère participative et collaborative passant par l'Intelligence Artificielle et la Blockchain.<sup>4</sup>

L'avancée fulgurante des nouvelles technologies est si impressionnante que PassCare ne pourrait se limiter qu'à « juste » partager de les données de santé. Nous voulons apporter des services interactifs supplémentaire qui serviront non seulement aux patients mais aideront aussi les professionnels dans leur métier.

Les données recueillies sont stockées mais aussi traitées pour en tirer des informations cruciales qui ne sont pas forcément détectable sur l'instant par le cerveau humain. Avoir des conseils sur sa santé grâce aux médecins présent sur la plateforme, des notifications rappelant la prise de médicaments, d'examens ou de vaccins, de renouvellement d'ordonnance, etc. sont des services proposés par PassCare aux patients mais nous pouvons aussi, faciliter le suivi d'une maladie pour les médecins voire prévoir son avancé grâce à l'intelligence artificielle.

Pour atteindre ces objectifs il faut une forte une croissance organique et une grande mobilisation des ressources internes de l'entreprise pour dépasser le stade de start-up et développer notre technologie à son paroxysme. Cela nécessite du temps mais surtout de la stratégie, c'est dans cette optique que InnovHealth a signé un partenariat avec le fonds franco-américain Forepont Capital Partners qui permet une augmentation de capital de 40 millions

---

<sup>4</sup> [Adexis Solution - Adnan El Bakri](#)

d'euros et l'acquisition de l'entreprise américaine BeCareLink (spécialisée dans la prévention, la prédiction et le suivi de la progression des maladies chroniques neurodégénératives) devenant ainsi **ReLyfe**.

#### 1.4. ReLyfe

ReLyfe (Real-Life) se définit comme une compagnie de MedTech venue révolutionner les technologies de communication de données dans la santé en offrant un profil de santé interactif (et intelligent) aux patients, et aux médecins la possibilité d'avoir un lien numérique avec ses patients sans se séparer de ses logiciels métier tout en se connectant à d'autres pour le partage et la recherche.

ReLyfe est une plateforme web et mobile, un outil qui promet une prise en charge particulière des maladies chroniques grâce à une prévention et une prédiction avancées faite par notre intelligence artificielle, cette dernière structure et analyse toutes les données des patients pour les suivre tout au long de leur parcours de santé.

#### 1.5. Recherche et Développement

ReLyfe ne compte pas s'arrêter là et grâce au soutien financier de Forepont, au contexte actuel où la santé est un sujet capital, nous pouvons faire une conjecture plutôt positive quant à la croissance de notre structure et au progrès de nos recherches. Notre technologie a été conçue pour progresser, notre intelligence artificielle sera constamment nourrie de données réelles pour son apprentissage, associé à un algorithme décisionnel elle pourra prédire l'avancée d'un certain nombre de maladies voire l'apparition de cancers.

Ce qui fait la force de notre solution, c'est qu'elle répond très simplement à un besoin quasi vital et mondial que les professionnels de santé tout comme les patients identifient depuis longtemps, problème que personne n'a réussi à résoudre, et qui vient bien avant l'IA.

– *Dr. Adnan El Bakri.*

Etant toujours prêt à relever de nouveaux défis je suis très enthousiaste à l'idée de vivre une épopée technologique, je vous invite donc à découvrir l'envers du décor de ce qui fait déjà partie du futur de la MedTech.



Figure 2 : Carte ReLyfe

## 2. Environnement de travail

### 2.1. Equipe

Une bonne start-up c'est avant tout une bonne équipe, c'est pourquoi derrière ReLyfe il y a toute une équipe de passionnés. Passionnés par ce qu'ils font et par le produit qu'ils développent avec la même ambition de rendre le parcours du patient encore plus numérique. De l'administration à l'assurance du service commercial, le cœur de ReLyfe est composé de talent, mais toute suite laissez-moi vous introduire à l'équipe technique qui se charge de faire tourner les boulons de la plateforme.

Pour présenter l'équipe technique :

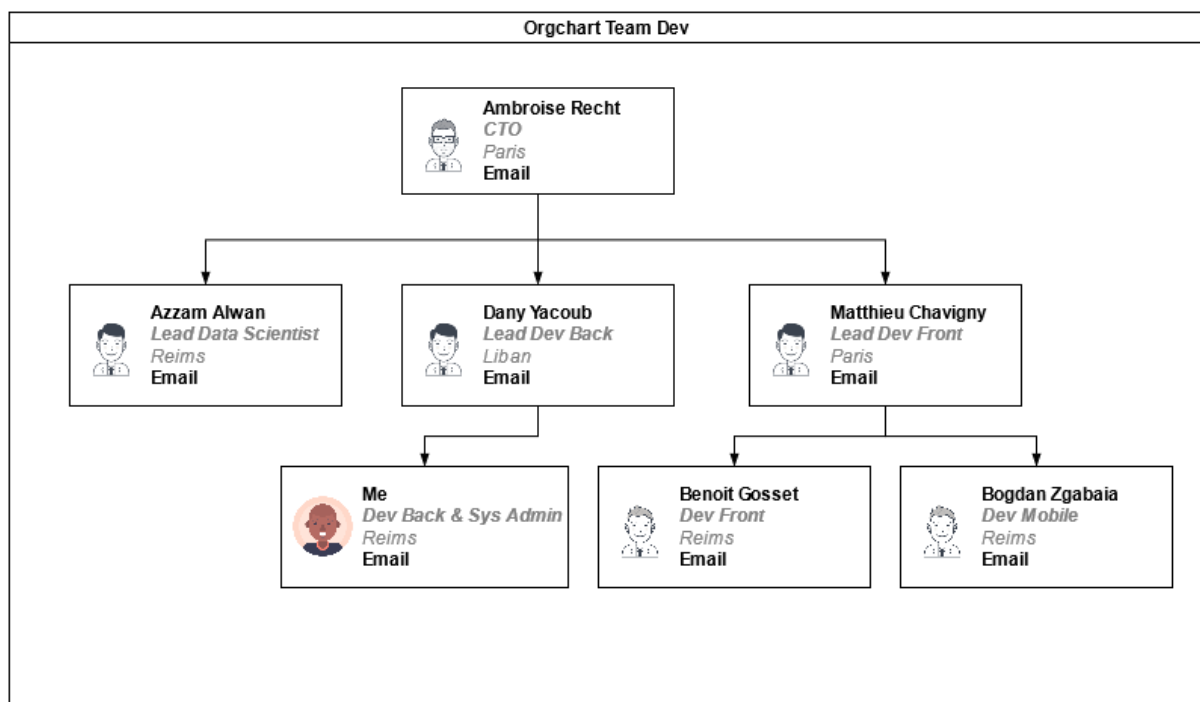
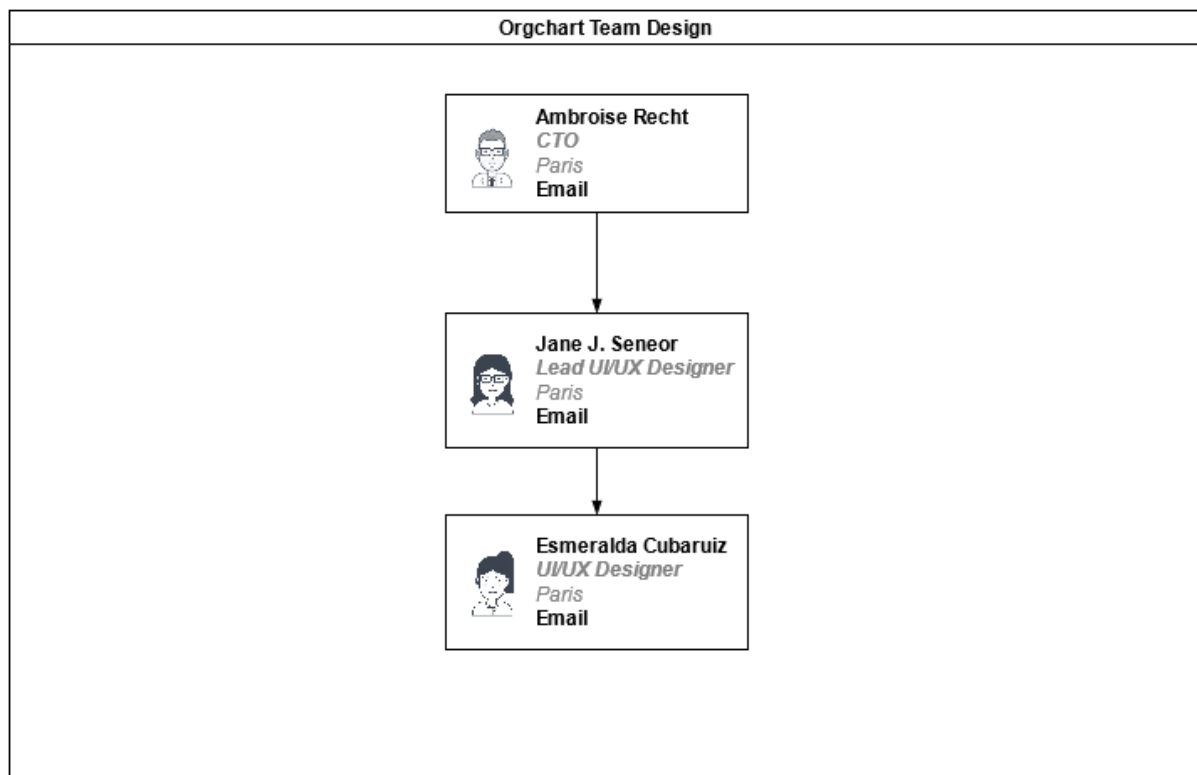


Figure 3 : Organigramme équipe technique

Nous avons tout d'abord

- **Ambroise Recht** notre CTO (Chief Technical Officer) en plus d'être mon tuteur, dont l'expérience n'a d'égale que sa passion pour son travail. Sa fonction consiste à étudier l'ensemble des possibilités de développement technique de la plateforme en faisant des choix à la fois logique, utile et économique dans les outils et technologies que nous utilisons, cependant il a aussi la casquette de Chef de Projet.
- **Azzam Alwan** est notre Lead Data Scientist après avoir été chercheur en Deep Learning chez Orange. Il s'occupe de créer et faire évoluer l'intelligence artificielle, principal atout de ReLyfe.

- **Dany Yacoub**, Lead Dev Back, titulaire de plus de 15 ans d'expérience en développement logiciel. Il se charge du développement, du déploiement et de l'efficacité technique de chaque API, utiles aux développeurs front.
- **Matthieu Chavigny**, Lead Dev Front. En plus de la gestion de l'équipe front, Matthieu développe l'application web tout en s'assurant de l'intégration des écrans proposés par l'équipe Design.
- **Bogdan Zgabei**, Développeur Mobile, il a comme responsabilité le développement et le déploiement des applications mobile iOS et Android tout en étant conforme aux écrans prévus.
- **Benoit Gosset**, Développeur Front, s'assure d'implémenter les pages et écrans de l'application web tout en optimisant l'ergonomie et la navigation de cette dernière.



*Figure 4 : Organigramme équipe design*

L'équipe front travaille en collaboration avec l'équipe design composée de **Jane Seneor** et d'**Esmeralda Cubaruiz**, elles ont entre leurs mains le visage de la plateforme car elles leur fournissent les écrans à implémenter tout en respectant des contraintes d'UI/UX et d'accessibilités.

Malgré la modeste taille de cette équipe, il faut une rigueur et un respect dans le travail de chacun pour pouvoir collaborer en étroite harmonie. De plus les aléas de la pandémie qui nous force à rester en télétravail, ne nous facilite pas la tâche.

C'est pourquoi nous apportons une grande importance dans les bonnes pratiques de gestion projet assurée par Ambroise Recht.

## 2.2. Gestion de projet

Plus que de bonnes pratiques, la gestion de projet est un impératif qui aide une équipe à être plus performante en s'assurant que les objectifs fixés soient atteints. C'est dans cette optique que nous utilisons une méthode agile de gestion de projet en appliquant ces principes clés :

1. Prioriser la satisfaction du client : les responsables projet sont intégrés dans nos cycles de développement (sprint) et valident les écrans proposés par l'équipe design.
2. Accepter les changements : principe même de l'agilité, nous sommes dans l'obligation de travailler en cycles, deux semaines dans notre cas, ce qui permet de recadrer et repartir dans la bonne direction en cas d'erreur.
3. Livrer en permanence des versions opérationnelles de l'application : chaque sprint a un objectif et est terminé avec au moins une mise en prod ou en préprod avec de nouvelles fonctionnalités.
4. Assurer le plus souvent possible une coopération entre l'équipe du projet et les gens du métier : en plus des responsables produit nous avons également un médecin présent lors de nos sprints review.
5. Construire les projets autour de personnes motivées : comme mentionné plus haut chacun des membres de notre équipe ont été recrutés parce que ce sont avant tout des passionnés de e-santé mais surtout parce qu'ils veulent voir le produit évoluer.
6. Favoriser le dialogue direct : grâce à Teams nous pouvons contacter n'importe quel membre de l'équipe par appel si ce dernier n'est pas déjà en communication, l'oral est privilégié pour tout échange jugé important.
7. Mesurer l'avancement du projet en fonction de l'opérationnalité du produit : en début et fin de sprint nous évaluons les tickets accomplis et restant pour pouvoir en faire un bilan de productivité.
8. Adopter un rythme constant et soutenable par tous les intervenants du projet : les sprints sont toujours de deux semaines et chaque requête de changement est évaluée pour entrer dans un sprint, si ce dernier est plein la requête est repoussée au prochain sprint.
9. Contrôler continuellement l'excellence de la conception et la bonne qualité technique : chaque ticket est composé de spécifications pour en comprendre l'objectif et est validé par une phase de test du chef de projet. Par ailleurs chaque incrémentation de code est validée par un lead dev en merge request avant déploiement.

10. Privilégier la simplicité en évitant le travail inutile : chaque epic story (histoire/cas d'utilisation/grosse fonctionnalité) est divisé en tickets, eux-mêmes subdivisés en sous tickets pour simplifier ce qui pourrait ressembler à une montagne.
11. Autoorganiser et responsabiliser les équipes : notre chef de projet utilise un management soft et laisse l'équipe auto-gérer l'avancement des tickets sachant que nous faisons des Daily Meeting chaque matin pour favoriser également l'échange.
12. Améliorer régulièrement l'efficacité de l'équipe en ajustant son comportement : on a un temps bilan après chaque sprint review pour donner notre ressenti sur le sprint passé et apporter les changements nécessaires dans les méthodes que nous appliquons.

Ci-dessus sont les principes clés d'une méthode agile que nous implémentons avec la méthodologie Scrum dont le principe de base est que l'équipe avance ensemble et soit toujours prête à réorienter le projet au fur-et-à-mesure de sa progression avec les différentes réunion mises en place :

- Sprint Planning : en début de sprint pour planifier son déroulement et définir son objectif.
- Daily Meeting : tous les jours pour échanger sur l'évolution des tickets et résoudre les blocages des membres de l'équipe.
- Sprint Review : en fin de sprint pour en faire le bilan.
- Backlog Grooming : une fois par sprint pour planifier les sprints à venir et replacer les tickets en Backlog.

Et les artefacts, des documents utiles à l'équipe tout au long de l'avancement du projet, parmi lesquelles :

- Le Product Backlog : liste hiérarchisée des exigences initiales du client.
- Le Sprint Backlog : plan détaillé de la réalisation de l'objectif du sprint.
- Le Burn down Chart : graphique indiquant l'état d'avancement dans la réalisation des tâches du Sprint Backlog.

Ces derniers sont faciles d'accès à toute l'équipe vu l'outil que nous utilisons au quotidien pour appliquer toutes ces règles de gestion de projet, que je vais présenter sans plus attendre.

## 2.3. Outils

En tant que start-up, ou entreprise en général, nous avons un portefeuille d'outils qui nous aident à concrétiser nos rêves tout en étant productif sur tous les plans.

### 2.3.1. Jira & Confluence

Jira est un système de suivi de bugs, un système de gestion des incidents, et un système de gestion de projets développé par Atlassian.

Utilisé par plus de 180 000 clients dans près de 190 pays, nous le faisons également en mode cloud.

Avec Jira nous pouvons suivre l'avancement de notre projet en coordonnant l'équipe avec des user stories et des tickets qui représentent les tâches associées à chaque membre de l'équipe, ces derniers sont priorisés et catégorisés selon que c'est une fonctionnalités front ou back (Voir annexe).

Confluence lui est un outil de création et d'illustration collaborative, aussi développé par Atlassian il nous donne un moyen de créer et centraliser toutes les informations liées à nos processus interne.

Sans aucun doute la meilleure combinaison pour une équipe agile. Nous pouvons par exemple créer des rapports (artefacts) qui permettent d'évaluer la vélocité (et d'autres indicateurs Scrum) de l'équipe avec Jira et garder un historique dans Confluence. De surcroît, avec les applications et module complémentaire nous somme capable d'intégrer nos autres outils à Jira pour automatiser nos processus et centraliser l'information, l'avancement des branches git par rapport aux tickets notamment.

### 2.3.2. GitLab

Nous versionnons donc le code de nos API et du front de la plateforme sur un serveur GitLab auto-héberger, les dépôts y sont structurés en groupe de projet (front, back, IA, etc.) avec un processus d'incrémentation de code bien précis. Chaque dépôt possède une branche `master` contenant toujours du code stable et une branche `develop` contenant toutes les modifications des développeurs après merge request. Justement, chacun des développeurs pousse son code sur une ou plusieurs branches du dépôt git lié au ticket en cours de traitement, et fait une merge request (MR) pour faire valider ses modifications par un lead dev.

Par ailleurs notre serveur GitLab nous sert aussi pour faire du DevOps et de l'intégration continue.

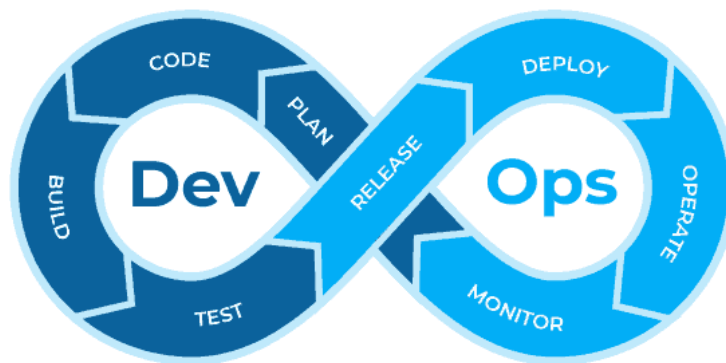


Figure 5 : Processus DevOps

Le DevOps offre un Framework conçu pour dynamiser et améliorer le développement d'applications [...] Il favorise l'intégration, la visibilité et la transparence continues entre les équipes chargées du développement d'applications (Dev) et celles responsables des opérations IT (Ops).<sup>5</sup>

A chaque push sur un dépôt des tests automatiques sur la qualité du code fourni sont réalisés et des scripts de déploiement peuvent être réalisés manuellement par le lead dev quand il valide une merge request.

Ces informations sont donc transmises à notre tableau de tickets Jira dans l'onglet développement. De plus, les développeurs peuvent effectuer des actions sur les tickets (comme les faire transiter d'un état à un autre) en indiquant des mots clés dans les messages de commit. Mais Jira ne s'arrête pas là car nous pouvons également l'intégrer à notre plateforme d'échange, plus connue sous le nom de Teams.

### 2.3.3. Teams (Office 365)

On ne présente plus Teams, la plateforme collaborative signé Microsoft. Grâce à notre espace Office 365 nous avons accès à tous les outils de bureau de Microsoft dont Teams, en application de bureau ou sur mobile elle nous aide à organiser nos réunions, échanger de façon directe en orale ou en écrit entre chaque équipe et ainsi garder le contact en toute

---

<sup>5</sup> [NetApp : DevOps](#)



situation. Ainsi, la flexibilité de Teams nous permet de recevoir les événements Jira et GitLab dans un channel dédié.

En misant sur les bons outils nous espérons avant tout un maximum de productivité, et cela dans la collaboration de l'équipe comme dans les engrenages qui font tourner la plateforme.

## 2.4. Architecture & Outils Dev

Notre plateforme est destinée à évoluer c'est pourquoi nous nous devons de la construire sur de bonnes bases c'est à dire avec une architecture stable et évolutive, capable de croître à la même échelle que le nombre d'utilisateurs.

Pour les raisons évoquées et d'autres plus spécifiques nous avons, pour le back end, une architecture en micro-services avec des API REST s'exécutant dans des dockers.

La technologie docker permet non seulement d'exécuter les APIs dans des « machines virtuelles », mais aussi de simplifier leur conception et leur création en automatisant des actions comme l'envoi d'images, le contrôle des versions d'images, etc. Une image docker représente son contenu à un instant t, en utilisant des dockers les APIs fonctionnent indépendamment les unes des autres, si l'une d'elles est hors service les autres continuent de tourner malgré tout. Ce qui est l'un des principes d'une architecture micro-services, en plus d'optimiser les performances elle décompose une application pour en isoler les fonctions clés. En outre, les services peuvent communiquer entre eux, cela encourage nos pratiques de DevOps en facilitant la détection de bug et l'ajout de nouvelles fonctionnalités.

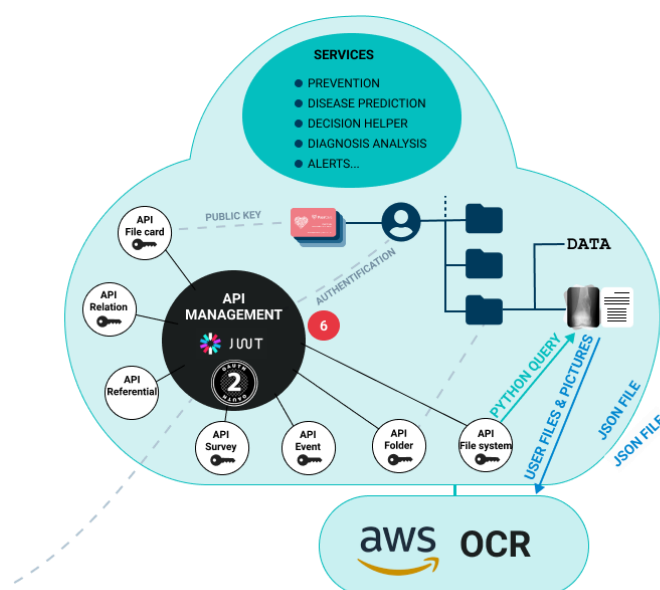


Figure 6 : ReLyfe API services

C'est ainsi que nous avons l'API Management développé en Node.js qui fait office de contrôleur vers les autres services, développé pour la plupart en Symfony, un Framework PHP multi-services dans lequel j'ai un minimum d'expériences – expériences renforcées par ma période d'apprentissage.

Ces services sont déployés et évoluent dans 3 environnements distincts respectant l'approche DevOps :

- Production (Prod) : un environnement stable sur lequel les services tournent de façon opérationnelle et où sont stockés les « vrais » utilisateurs.
- Pre-Prod : un environnement répliquant celui de la production, sur lequel on peut faire des tests supplémentaires avant un déploiement définitif en production.
- Qualif : similaire à un environnement de développement, il sert à tester les incréments de code qui sont faites sur `develop`.

Chaque environnement exécute les services de manière autonome et indépendamment des autres, chacune a sa base de données et ses instances de dockers.

En tant que développeur back, j'ai pu participer à l'incrémentation des fonctionnalités et à l'assurance de la qualité du code de ces services, regardons en détail en quoi ont pu consister mes tâches.

### **3. Accomplissements**

---

Depuis le 25 août 2020 exactement, j'ai pour mission chez ReLyfe, d'administrer et de gérer les outils cloud que nous utilisons au quotidien, cela inclus la gestion des utilisateurs, du bon fonctionnement des systèmes et la maintenance. Cependant mes principales tâches restent orientées autour du développement des APIs.

#### **3.1. Développement**

Justement après avoir passé une à deux semaines sur un « projet » test dans l'objectif de découvrir l'architecture et les conventions de code, l'un de mes premiers tickets a porté sur l'ajout d'une nouvelle fonctionnalité, la gestion de traitement, sur l'API Folder. On devait être

capable d'ajouter un traitement, l'associé à une maladie et y rajouter des médicaments et une ordonnance. En collaboration avec mon ancien tuteur en entreprise, développeur backend, on a dû produire un MCD validé par l'ancien lead dev et j'ai dû implémenter dans l'API tout ce qui y figurait. Pendant le processus de développement, après des échanges répétés avec le Product Manager j'ai pris le soin d'ajuster les imperfections de conception de départ relatives à la présence et au type de certains attributs, toujours avec une validation préalable du lead dev.

L'ajout de nouvelles fonctionnalités s'accompagne toujours de tests fonctionnels et d'une documentation, les premiers s'effectuent avec Postman qui sert à exécuter des appels HTTP vers l'API directement depuis une interface graphique et la dernière avec Swagger qui est un outil servant à décrire les spécifications d'API RESTful, sa plus-value étant sa capacité de créer des pages interactives avec la possibilité d'exécuter les requêtes décrites sans quitter son navigateur. J'ai dû parcourir des documentations pendant un peu plus de deux jours pour apprendre l'utilisation de ces deux outils, sans oublier ma découverte de docker qui continue jusqu'à présent. En effet avant d'intégrer l'équipe de développement de ReLyfe je n'avais jamais utilisé de docker avant et j'ai dû apprendre sur le tas en installant en local toutes les APIs et en me documentant à chaque action que je faisais et chaque commande que je lançais.

Bien que je me sente plus à l'aise avec les images dockers, je n'en connais pas encore toutes les subtilités et je bloque sur certaines problématiques quand il s'agit de faire des déploiements en local. Dans ces situations je peux compter sur l'expertise de Dany qui me débloque et me fait progresser par la même occasion. De la sorte j'ai pu accomplir l'un de mes plus grands succès avec docker : installer localement Xdebug dans un docker et l'intégrer à mon EDI.

Pour contextualiser, il faut savoir que je développe avec PhpStorm l'un des meilleurs éditeurs de développement intégré pour PHP par JetBrains. Il est très contraignant de suivre l'exécution d'un programme en affichant la valeur de différentes variables à chaque nouvelles requêtes c'est pour cela qu'il existe des outils comme Xdebug permettant de faire du debugging étape par étape tout en avançant librement dans le programme pour voir le changement d'états des variables. Son intégration avec PhpStorm va encore plus loin car elle permet de le faire pendant l'exécution même du programme avec des informations détaillées sur le contenu des variables et sur la pile d'exécution des fonctions. Xdebug est une extension PHP, elle dépend donc de l'exécutable et de la version installé sur le système. Cependant

avec Docker le code est exécuté dans une machine virtuelle, ce n'est par conséquent pas le même PHP qui s'y exécute, même en installant Xdebug localement il ne pourra pas interagir avec la version de PHP dans le docker. Pour résoudre ce problème il m'a fallu lire plusieurs articles traitant du sujet pour enfin :

- Modifier le script de création du docker
- Exposer le port utiliser par Xdebug dans le docker
- Configurer les bonnes variables d'environnement
- Mettre en place un remote debugging
- Configurer PhpStorm en créant un serveur virtuel et en utilisant l'interpréteur PHP installé dans le docker

Par ailleurs il a fallu prendre en paramètre le fait que les images docker sont censés être les mêmes en local qu'en environnement de production, il me fallait donc tenir compte de ce détail dans mes modifications.

Ces modifications m'ont pris du temps mais m'ont été utile non seulement à en apprendre plus sur docker et Xdebug mais aussi à booster ma productivité.

J'ai par la suite travaillé sur la communication entre les APIs, les notifications sur la plateforme, l'optimisation et le refactoring du code de certaines APIs que l'on peut voir dans mes derniers tickets attribués. Une mention spéciale à Dany Yacoub qui prend le soin d'étudier toutes mes recommandations en me disant quand elles sont bonnes et pourquoi elles ne le sont pas quand c'est le cas.

En ce qui concerne les notifications, implémentées sur API Notifier – un service qui envoie des notifications push au navigateur et à l'OS de l'utilisateur, j'y ait passé beaucoup de temps car il s'agissait de faire évoluer le code de cette API aux nouvelles conventions en automatisant certaines tâches avant de rajouter de nouvelles notifications. Ce fut la première fois où j'ai eu autant de liberté sur ce que j'avais à faire, définir une petite architecture et un format de données est un exercice de conception qui me plaît énormément. Ce service n'étant pas présent dans API Management – le contrôleur des APIs – il a fallu le rajouter plus tard, après avoir testé le fonctionnement de ce dernier au sein de l'architecture globale en environnement de production.












Résumé	T	Mise À Jour	Attribution	Rapporteur	État	Résolution
Générer la carte sous forme d'image à afficher directement dans le front (avec le nom et le la clés public et QRcode)		07/juin/21 9:37 AM	Germain OLEA-OYUGOU	Ambroise Recht	À FAIRE	Unresolved
Ajout d'un paramètre 'source' sur le call api en front		23/avr./21 1:53 PM	Dany Yacoub	Germain OLEA-OYUGOU	TERMINÉ(E)	Unresolved
Ajout des Notifications liées à l'ajout d'un patient par un docteur		23/avr./21 1:53 PM	Dany Yacoub	Ambroise Recht	TERMINÉ(E)	Unresolved
BACK - Notification : corriger dans api-relation		18/mai/21 5:10 PM	Ambroise Recht	Germain OLEA-OYUGOU	TERMINÉ(E)	Unresolved
BACK - ErrorType api-notifier		07/avr./21 5:59 PM	Dany Yacoub	Dany Yacoub	TERMINÉ(E)	Unresolved
Move ErrorType from api-common to other api		08/avr./21 9:53 AM	Germain OLEA-OYUGOU	Germain OLEA-OYUGOU	TERMINÉ(E)	Unresolved
BACK - faire une documentation api-common		19/mars/21 5:20 PM	Ambroise Recht	Germain OLEA-OYUGOU	À FAIRE	Unresolved
BACK - Doctor - Erreur 500 sur get_doctor_files - all		18/mars/21 11:02 AM	Germain OLEA-OYUGOU	Dany Yacoub	À FAIRE	Unresolved
BACK - Notification : docteur reçoit un document		25/mars/21 8:33 PM	Ambroise Recht	Dany Yacoub	TERMINÉ(E)	Unresolved
Refact notifier : step 2		15/mars/21 9:26 AM	Dany Yacoub	Dany Yacoub	TERMINÉ(E)	Unresolved
Affichage de 10 sur 39 demandes  Actualiser						

Figure 7 : Derniers tickets traités

Ces tickets sont liés au tableau ReLyfe Developers, il existe un deuxième tableau qui m'est attribuer pour les tickets liés à l'administration : Administrator.

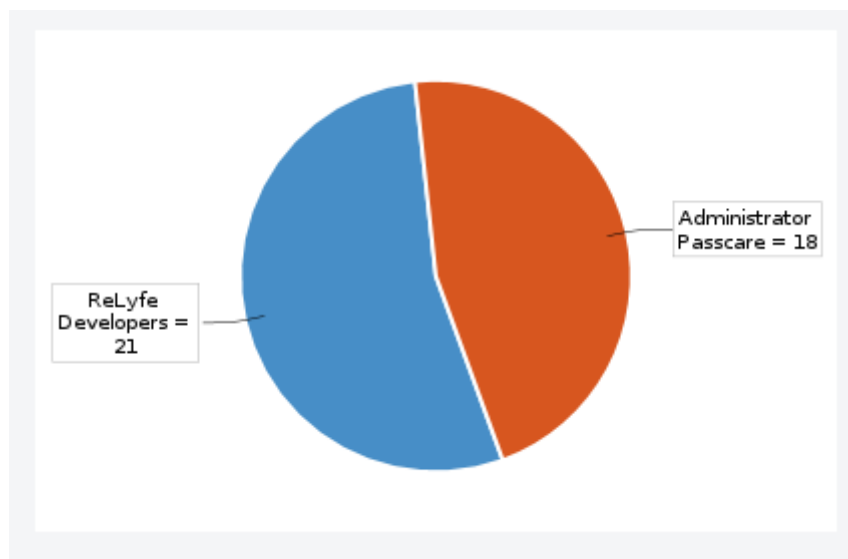
### 3.2. Administration

En tant qu'administrateur j'ai eu l'occasion d'ajouter les nouveaux utilisateurs, supprimer les sortants, gérer les licences Office et les bibliothèques OneDrive partagées, dépanner ceux qui avaient des soucis avec l'un des outils quand mes priorités de développement le permettaient, etc. Toutefois le plus plaisant a été de faire l'intégration d'un outil à un autre. Notamment celle de GitLab vers Jira et Teams.

La première nous a été très utile car elle a ouvert le champ des possibilités de ce qui nous servait jusqu'à lors « que » de recueil de tickets. Effectivement, en plus de nous donner un moyen de voir les actions de développement sur le ticket associé, nous pouvions également créer des workflows, c'est à dire l'automatisation de certaines tâches répétitive de notre processus de développement comme la transition d'un ticket de IN PROGRESS à CODE

REVIEW lors de la création d'une merge request. Et la seconde nous envoie des notifications sur chaque événement de chaque utilisateur sur le GitLab dans un canal sur Teams.

Comme on peut le voir dans le graphe ci-dessous, mes tâches d'administration sont nettement inférieures à celle de développement, d'abord parce que ces dernières me prennent plus de temps et ensuite parce qu'il y a une plus grande priorité à développer de nouvelles fonctionnalités mais la principale des raisons est que l'actuel CTO se charge d'une partie de l'administration et me délègue le reste quand il en trouve la nécessité.



*Figure 8 : Graphique tableaux Jira*

Mes journées sont donc réparties entre administration de services et développement d'APIs, cela est nettement suffisant pour les remplir mais l'on en fait vite le tour. L'un des nombreux avantages qu'il y a à être dans une start-up est de monter rapidement en compétences, j'espère donc continuellement toucher à de nouvelles technologies.

## 4. Projection tâches futures

---

En tant développeur dans une start-up on est amené à toucher à plusieurs aspects du métier, pour ne citer que Dany et Matthieu qui sont lead dev et DevOps. Le métier de DevOps est très large et m'intrigue beaucoup, je voudrais donc avoir, ne serait-ce qu'un peu de connaissance dans ce domaine avec la maîtrise de technologies comme Kubernetes ou Docker. Mais d'abord j'aspire à mieux connaître l'architecture front et dépasser le stade de développeur back end pour devenir full stack.

Il est prévu dans ma prochaine période d'apprentissage de faire du pair-programming avec l'un des développeurs front, j'en attends de bons résultats quant à mes ambitions à courts termes. Mon objectif est d'en apprendre le plus possible tout en maîtrisant tout le processus de développement d'une application en général et de la plateforme ReLyfe en particulier.

En ce qui concerne celles à long termes elles incluent l'acquisition de nouvelles compétences en gestion de projet et en conception comme prévu par ma formation, je compte sur mes cours pour m'apporter les bases et sur mon apprentissage pour l'expérience. Voir un Product Manager et un chef de projet sur le terrain est plus formateur que de la simple théorie. Et être par la suite capable de :

- De s'occuper d'une équipe en communiquant sans ambiguïté
- D'analyser les risques globaux d'une prise de décision
- D'établir par la suite une stratégie pour y répondre
- D'innover constamment.

Quant à la plateforme on peut évaluer plus ou moins facilement les futures fonctionnalités à implémenter en faisant un tour sur notre tableau Jira, on peut voir que les tickets concernent :

- L'intégration de la récente API IA faisant tourner l'intelligence artificielle dans le front. Elle doit être capable d'extraire les informations clé d'un document de santé, notamment son type (analyse, ordonnance, radiologie, etc.), les médicaments présents avec leur posologie ou les examens à réaliser s'il s'agit d'une ordonnance.
- L'ajout du processus de paiement.
- L'évolution de la fonction de partage.
- Correction de bugs et amélioration.

- Etc.

<p>Patient can pass his account as Premium account with subscription</p> <p>Freemium/Premium</p> <p>BACK, FRONT</p> <p><input checked="" type="checkbox"/> = <input type="radio"/></p> <p> RLD-1015</p>	<p>Partage Partiel - Partage via la création d'un lien ou d'un QR Code</p> <p>Patient - Medical ong...</p> <p>BACK, FRONT</p> <p><input checked="" type="checkbox"/> = <input type="radio"/> 13</p> <p> RLD-926</p>	<p>Améliorer la recherche du patient par le pro</p> <p>Doctor - My Patients</p> <p>BACK, FRONT</p> <p><input checked="" type="checkbox"/> = <input type="radio"/> 5</p> <p> RLD-603</p>
--	--	--

En partant donc du principe que je ferai du front dans les prochaines semaines et que les écrans de ces User Stories soient déjà prêt, je pourrais après avoir assimiler les bases de l'architecture front créer mes premières vues pour l'un de ces tickets dans le dépôt front. Sans oublier mes tâches d'administration régulière.

Ces conjectures pourraient être justes comme fausses, il est aussi tout à fait possible aussi que je ne fasse pas du front mais que je continue à faire du back pour soutenir Dany Yacoub qui est actuellement seul développeur back. Néanmoins il reste intéressant de prédire ses futures tâches pour mieux se projeter dans leurs réalisations.



## Conclusion

Dans le cadre de ma poursuite d'études, le choix de l'alternance m'est paru comme une illumination compte tenu des avantages multiples qu'elle procure à la fois au niveau scolaire que professionnel. ReLyfe, ex InnovHealth, est l'entreprise avec laquelle j'ai voulu passer mes trois années de formation. Se définissant comme future leader de la MedTech leurs valeurs et les miennes coïncident vers un idéal commun : l'accès à l'information médicale pour tous. Arrivée au terme de cette première année haute en couleurs il m'est dans l'obligation d'en faire un bilan.

Cette année fut sans doute très enrichissante, pour une première expérience professionnelle, elle a comblée presque toutes mes attentes. Faire partie d'une équipe émerveillée par le produit qu'elle représente, participer au développement d'une plateforme qui bénéficiera au plus grand nombre sont quelques raisons qui justifient mes mots. De plus j'ai eu la chance de travailler sur des technologies nouvelles et de renforcer les connaissances que j'avais déjà sur les anciennes, dans le cas de docker et de Xdebug avec PhpStorm il est sûr que cette découverte me sera utile tout le long de ma vie professionnelle. Cependant le télétravail constant imposé par la situation sanitaire a été selon moi un facteur qui fait légèrement pencher la balance de mon ressenti du côté négatif.

En somme, cette première année d'apprentissage aurait assemblée plus de bons points que de mauvais. Je n'en suis que plus motivé pour les années à venir attisant de plus en plus ma curiosité intellectuelle.

## Annexes

### Article 193 - Loi de modernisation de notre système de santé du 26 janvier 2016

X. Le groupement d'intérêt public " Institut des données de santé ", mentionné à l'article L. 161-36-5 du code de la sécurité sociale, dans sa rédaction antérieure à la présente loi, devient le groupement d'intérêt public " Institut national des données de santé ", mentionné à l'article L. 1462-1 du code de la santé publique, à la date d'approbation de la convention constitutive de celui-ci. L'Institut national des données de santé se substitue à l'Institut des données de santé dans l'ensemble des droits et obligations de ce dernier.

XI. Les organismes bénéficiant, à la date de la publication de la présente loi, d'un accès à tout ou partie du système national d'information interrégimes de l'assurance maladie mentionné à l'article L. 161-28-1 du code de la sécurité sociale conservent cet accès, dans les mêmes conditions, pendant une durée de trois ans à compter de cette publication.

XII. Les autorisations délivrées par la Commission nationale de l'informatique et des libertés sur le fondement des chapitres IX et X de la loi n° 78-17 du 6 janvier 1978 relative à l'informatique, aux fichiers et aux libertés, dans leur rédaction antérieure à l'entrée en vigueur de la présente loi, demeurent applicables sauf modification de l'un des éléments mentionnés à l'article 30 de la même loi.

## Jira

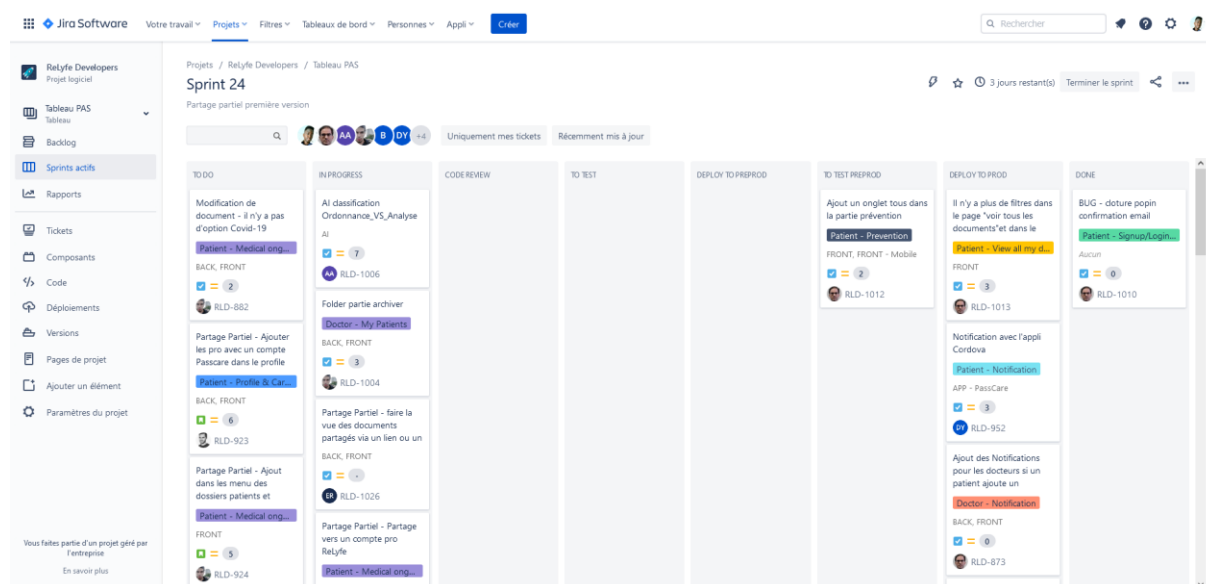


Figure 9 : Jira ReLyfe Developer

## Développement RLD-1004

Donnez votre avis

Branches Commits Pull requests Builds Déploiements Feature flags

Dépôt	Branche	Pull request	Action
innovhealth@back@api-folder	archive_doctor/RLD-1004	MERGÉE(S)	Créer une pull request
innovhealth@back@api-folder	delete_folder/RLD-1004	MERGÉE(S)	Créer une pull request
innovhealth@back@api-skeleton	archive_doctor/RLD-1004		Créer une pull request
innovhealth@front@pas-web	RLD-1004	MERGÉE(S)	Créer une pull request

Figure 10 : Onglet développement d'un ticket Jira

### Tech

RELIFE  
TECH BLUEPRINT

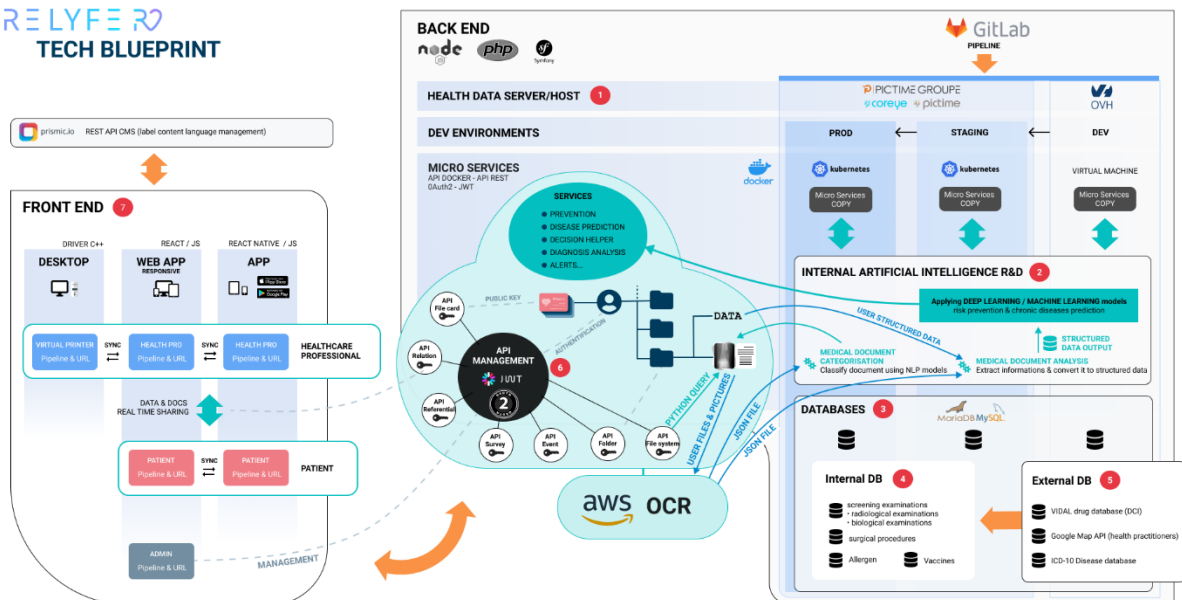


Figure 11 : Architecture ReLife