

ENONCE DES TRAVAUX D'ARCHITECTURE TOGAF

Imamou Chaima
Architecte logiciel MedHead+

Table des matières

| | | |
|------|---|----|
| I. | Contexte du projet | 2 |
| A. | Contexte de l'entreprise..... | 2 |
| B. | Objectifs du projet..... | 2 |
| C. | Les hypothèses du PoC..... | 2 |
| II. | Méthode TOGAF..... | 3 |
| A. | Description de la démarche ADM | 3 |
| III. | La phase préliminaire et la vision de l'architecture (Phase A) | 4 |
| IV. | Architecture métier (Phase B) | 4 |
| A. | Analyse de l'architecture de base | 4 |
| 1. | Description de l'architecture | 4 |
| 2. | Description détaillée de l'architecture..... | 6 |
| 3. | Objectifs et exigences qui doivent être atteints par l'état cible de l'architecture..... | 7 |
| 4. | Analyses et diagnostics..... | 7 |
| A. | Analyse de l'architecture cible | 8 |
| 1. | Micros-service mise en place dans le PoC du système d'intervention d'urgence | 9 |
| V. | Architecture SI (phase C) et architecture Techniques (Phase D) | 10 |
| A. | L'architecture du PoC mise en place | 10 |
| B. | Les différentes fonctionnalités des micros-services | 11 |
| C. | Les User Story | 11 |
| D. | Les outils et dépendances utilisés..... | 12 |
| VI. | Passage en revue des opportunités et des solutions (Phase E) et planning de migration (Phase F) | 13 |
| A. | La phase de test..... | 13 |
| B. | Les différents tests automatisés misent en place pour ce PoC..... | 13 |
| VII. | Mise en œuvre de la gouvernance (Phase G)..... | 14 |
| A. | Suivi du projet | 14 |
| 1. | Point journalier (Daily) | 15 |
| 2. | Point hebdomadaire..... | 15 |

I. Contexte du projet

A. Contexte de l'entreprise

MedHead est un regroupement de grandes institutions médicales œuvrant au sein du système de santé britannique et assujetti à la réglementation et aux directives locales (NHS). Les organisations membres du Consortium utilisent une grande variété de plateformes, de technologies et d'appareils qui souhaite utiliser Java comme langage principal.

Le groupe veut mettre en place une plateforme qui pallie les risques liés au traitement des recommandations de lits d'hôpitaux. Cette plateforme va permettre :

- d'améliorer la qualité des traitements d'urgence et de sauver plus de vies ;
- et de gagner la confiance des utilisateurs quant à la simplicité d'un tel système

La discussion de groupe s'est concentrée sur le système d'intervention d'urgence qui permet l'attribution en temps réel de lits d'hôpital en fonction de la pathologie.

Cependant, il a été décidé de mettre en place une preuve de concept afin de valider les différentes exigences et hypothèses du comité.

B. Objectifs du projet

L'objectif principal du projet est de développer une plateforme de services dont les systèmes communiquent grâce à des événements. Ces services ont une responsabilité unique. Ils sont également découplés entre eux et tolérants aux pannes.

La principale préoccupation des parties prenantes est actuellement le système d'intervention d'urgence en temps réel, notamment : sa capacité à fournir une allocation de lit en temps opportun, dans le service hospitalier approprié (en fonction de la pathologie).

C. Les hypothèses du PoC

Des hypothèses de développement de la preuve de concept pour le sous-système d'intervention d'urgence en temps réel, ont été abordées par le comité métier du Consortium MedHead.

Pour plus de détails consulter le document « Hypothèse de validation de principe ».

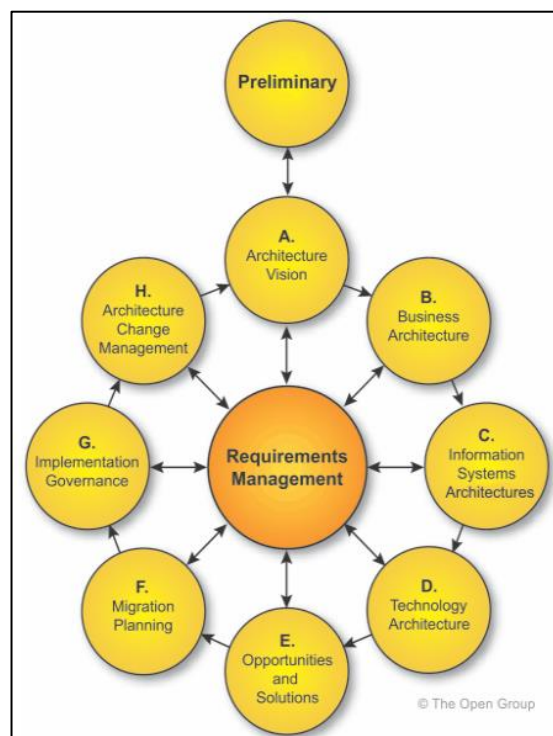
II. Méthode TOGAF

La méthode TOGAF intègre les stratégies, les exigences, les processus métiers, les applications, les infrastructures techniques et des associations efficaces entre ses différents aspects et va même jusqu'à la planification et la gestion du changement.

Le cycle **ADM (Architecture Development Method)** constitue le cœur de cette démarche.

A. Description de la démarche ADM

L'ADM se présente sous forme d'un découpage en phase, dont une phase préliminaire, nécessaire à l'organisation du projet, suivie de 8 phases (numérotées de A à H). Il permet de construire l'architecture des domaines métiers, applications, données et techniques, de planifier son déploiement, de la mettre en œuvre et finalement, de gérer les changements à opérer :



Source : the open Group

- La phase préliminaire
- Phase A : vision de l'architecture
- Phase B : architecture Métier
- Phase C : architecture des systèmes d'information
- Phase D : architecture technologique
- Phase E : solutions et opportunités
- Phase F : planning de migration

- Phase G : Gouvernance de la mise en œuvre
- Phase H : Gestion du changement d'architecture
- Phase gestion des exigences

III. La phase préliminaire et la vision de l'architecture (Phase A)

Cette partie permet de définir le périmètre du projet. C'est l'étape qui permet d'identifier les objectifs, les exigences métiers.

Nous avons abordé cette partie plus haute (contexte de l'entreprise).

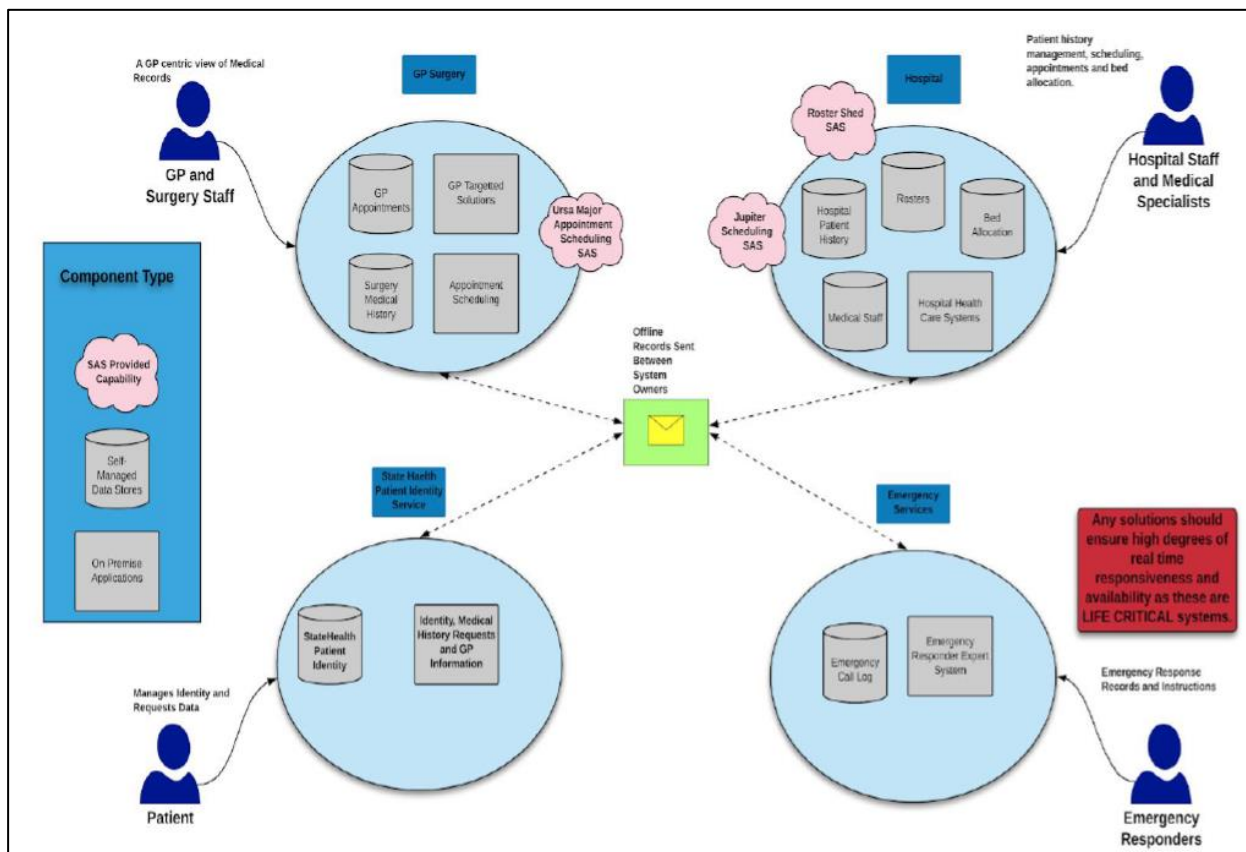
IV. Architecture métier (Phase B)

A. Analyse de l'architecture de base

1. Description de l'architecture

L'architecture de base se compose de systèmes disparates ciblés selon les cas d'utilisation et les contextes médicaux spécifiques :

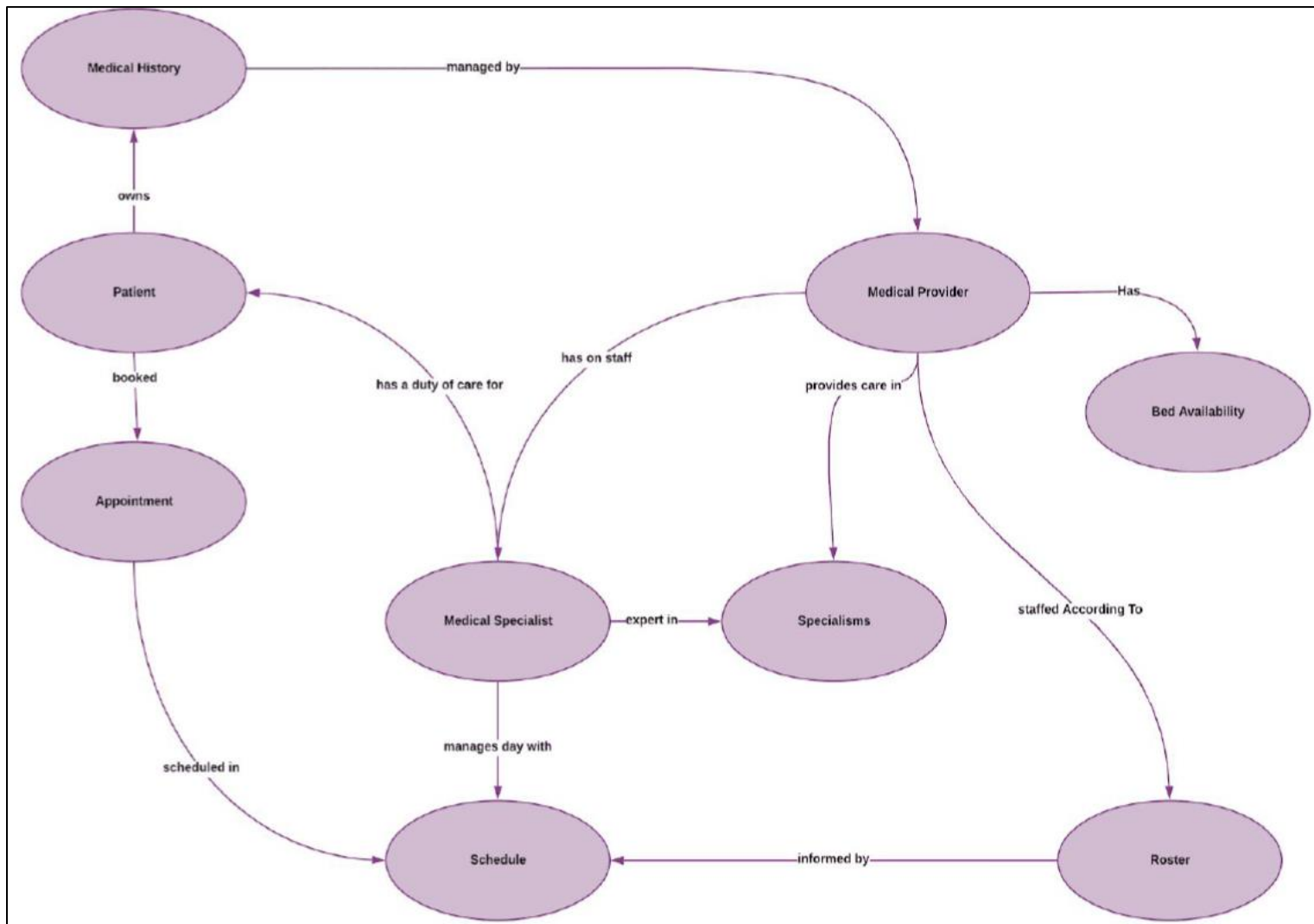
- Chirurgie (médecine générale) ;
- Hôpitaux ;
- Intervenants d'urgence ;
- Systèmes de gestion de l'identité médicaux gérés par l'État.



En effet, ce type d'architecture ne parvient pas à fournir une assistance en temps réel et une meilleure intégration et un partage des données entre les systèmes. Les échanges hors ligne entre les différentes spécialités se font par mails.

Par conséquent, un prototype sera construit en vue d'améliorer l'architecture actuelle et faciliter les échanges entre les hôpitaux, mais aussi de réduire les risques. On va commencer par le composant d'intervention d'urgence de l'architecture globale cible.

2. Description détaillée de l'architecture



- **Un patient** possède un historique médical et il peut réserver un rendez-vous sur un calendrier (en fonction des effectifs des prestataires des soins). Le patient a un devoir de diligence envers le médecin spécialiste.
- **Le prestataire de soins médicaux** gère l'historique médical. Et en fonction des effectifs, remplit la liste pour le calendrier de prise de rendez-vous. Le prestataire médical gère la disponibilité des lits, fournit des soins selon sa spécialité et dispose d'un médecin spécialiste sur place.
- **Le médecin spécialiste** est expert en sa spécialité. Le médecin spécialiste gère la journée avec le calendrier et la disponibilité des prestataires de soins médicaux.

3. Objectifs et exigences qui doivent être atteints par l'état cible de l'architecture

- Les systèmes et processus existants ne doivent pas être significativement entravés pendant les phases du projet.
- La confidentialité des données des patients doit être respectée et tous les prototypes doivent anonymiser les données ou utiliser des données factices.
- Les données réelles des patients doivent à tout moment rester conformes aux réglementations européennes, notamment le RGPD.
- Les phases initiales du projet devraient viser la création de modules de construction réutilisables.
- Fournir une API RESTful qui tient les intervenants médicaux informés en temps réel sur : le lieu où se rendre.
- Le PoC doit être entièrement validé avec des tests d'automatisation reflétant la pyramide de test (tests unitaires, d'intégration, d'acceptation et E2E) et avec des tests de stress pour garantir la continuité de l'activité en cas de pic d'utilisation
- Le PoC doit être facilement intégrée dans le développement futur : rendre le code facilement partageable, fournir des pipelines d'intégration et de livraison continue (CI/CD) et documenter votre stratégie de test.

4. Analyses et diagnostics

| Exigences / technologies manquant dans la solution initiale | Solutions |
|--|---|
| Gestion des évolutions futures de la solution <ul style="list-style-type: none">- Permettre des améliorations et modifications futures du système | <ul style="list-style-type: none">- Une architecture avec un faible couplage- Une indépendance des modules (autonomie)- Une possibilité de réutilisations des modules développés- Une architecture facile à maintenir <p>⇒ Mise en place de l'architecture micro-service</p> |
| Gestion de montée en charges : <ul style="list-style-type: none">- Obtenir un temps de réponse de moins de 200 millisecondes avec une charge de travail allant jusqu'à 800 requêtes par seconde) | <ul style="list-style-type: none">- Le système doit être capable de s'adapter à une forte augmentation de la demande sans que son niveau de performances n'en soit impacté (Scalabilité)- Mise en place d'une API Gateway et d'un load balancer |

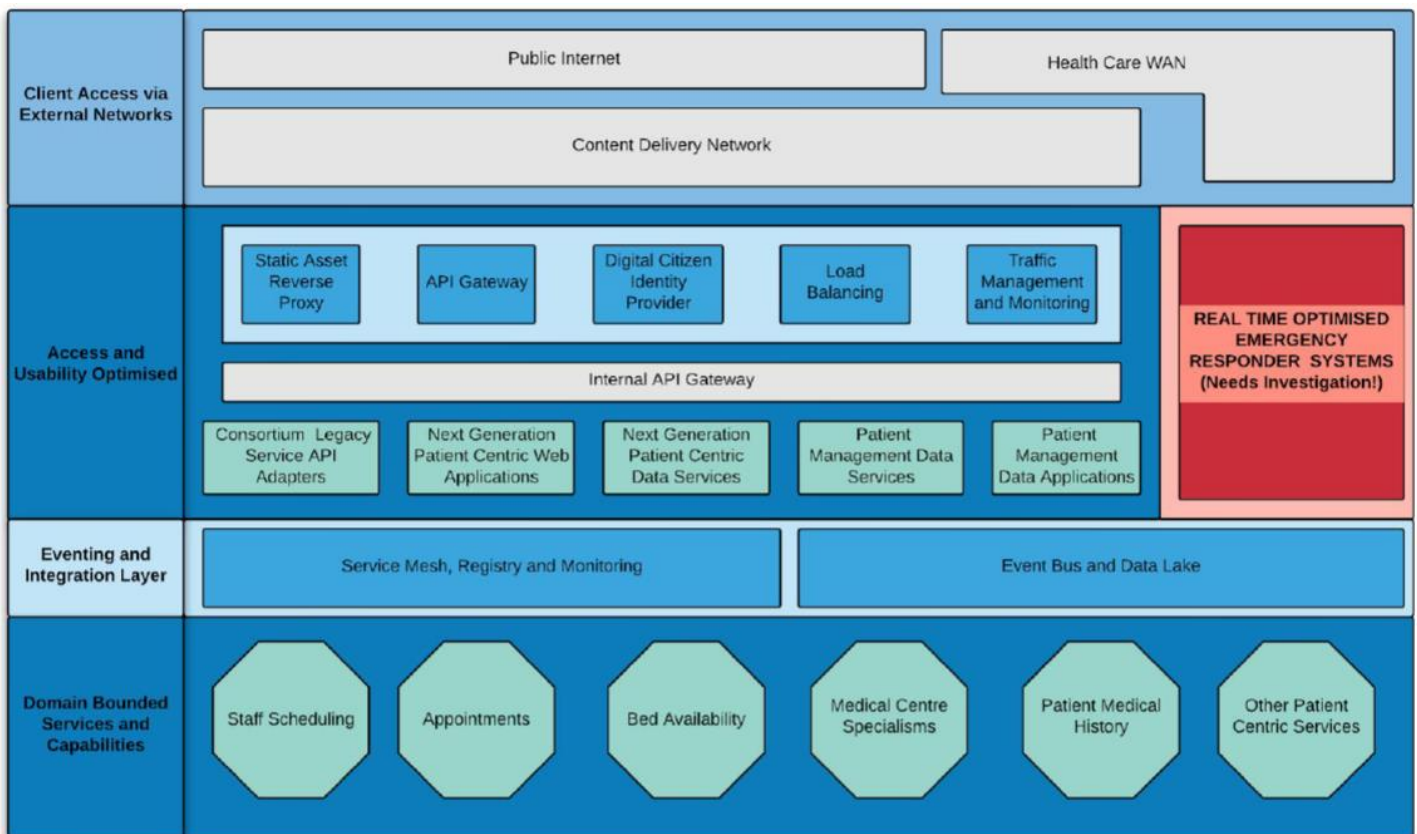
A. Analyse de l'architecture cible

A partir de l'ensemble des éléments que nous avons traité, l'architecture micro-service est la plus adaptée pour atteindre les objectifs.

En effet, selon les exigences techniques énoncées, à savoir que la solution doit permettre des améliorations et modifications futures du système. Cette architecture a un couplage faible qui permettra par la suite de pouvoir faire des évolutions sans impacter toute la solution.

Les micro-services développés sont parfaitement autonomes. En effet, chaque micro-service a sa propre base de données et pour faciliter la communication les uns avec les autres, chacun à son propre API REST.

Architecture proposée par l'entreprise :



L'architecture est divisée en quatre niveaux :

- **Accès client via des réseaux externes** : les clients externes (patients) et le personnel médical accèdent à la plateforme via cette couche.
- **Couche optimisée pour l'accès et l'utilisation** : ce niveau fournit des applications et des services hautement optimisés pour l'accès aux données et leur utilisation.

- **Systemes d'intervention d'urgence optimisés en temps réel** : les systèmes hautement optimisés pour des cas d'utilisation en temps réel et des niveaux élevés de tolérance aux pannes seront hébergés dans cette couche.
- **Couche d'événements et d'intégration** : comprends les bus d'évènement, monitoring, service Mesh, les services métiers de base...

1. Micro-service mise en place dans le PoC du système d'intervention d'urgence

Nous nous sommes concentrés sur deux micro-services :

| Micro-service | Objectifs |
|-----------------------------------|--|
| Gestion des hôpitaux | - Rechercher un hôpital en fonction de sa localisation et de sa spécialisation |
| Gestion des réservations des lits | - A partir de l'hôpital réserver un lit disponible selon la disponibilité des lits et en fonction de la spécialisation |

Nous avons développé et mise en place des services pour faciliter la configuration des micro-services :

| Services | Objectifs |
|-----------------|---|
| Configuration | Ce service permet de centraliser toutes les configurations nécessaires pour l'ensemble des micro-services. |
| Enregistrement | Ce service permet d'enregistrer la localisation de chaque micro-service. C'est un annuaire qui publie l'ensemble des informations des micro-services (nom, adresse IP, numéro de port). |
| API Gateway | C'est le point d'entrée unique pour les API et micro-services. Il joue le rôle d'intermédiaire entre l'application et les micro-services pour faciliter et surveiller les échanges |
| Application web | L'interface web qui permet aux clients de gérer la gestion des hôpitaux plus facilement |

Pour plus de détail, voir le document description de tous les éléments de base réutilisable (IV- description des micros-services).

V. Architecture SI (phase C) et architecture Techniques (Phase D)

Sur l'architecture des systèmes d'information, il est nécessaire de décrire les nouveaux composants du SI, leur intégration avec les applications existantes, la gestion, le traitement et la localisation des données. En effet, il faut décrire les modes de communication, les interfaces, les responsabilités de chaque composant.

Sur l'architecture technique, il est nécessaire de définir les choix technologiques retenus, les plateformes nécessaires à la mise en œuvre et au déploiement des nouveaux composants du SI.

A. L'architecture du PoC mise en place



Pour plus de détail , voir le document description de tous les éléments de base réutilisable (II- Architecture et communications des micros-services).

B. Les différentes fonctionnalités des micros-services

Pour plus de détail, voir le document description de tous les éléments de base réutilisable (IV-Description des micros-services).

| Fonctionnalité | Description |
|---|---|
| Saisir le lieu de l'incident d'urgence | Le personnel médical peut saisir le lieu de l'incident (en cas d'urgence) |
| Rechercher un hôpital en fonction de la spécialisation et du lieu de l'incident | En fonction de la localisation et de la spécialisation, l'hôpital le plus proche de l'incident doit être proposé (en fonction de la disponibilité des lits) |
| Vérifier si dans l'hôpital en question il y a un lit disponible | En fonction de la localisation et de la spécialisation, vérifier si l'hôpital possède des lits disponibles |
| Réservez un lit d'hôpital au nom du patient selon la disponibilité | Si un lit est disponible, un événement doit être déclenché pour réserver un lit |

C. Les User Story

| Fonctionnalités | Descriptions |
|---|---|
| Chercher et suggérer l'hôpital le plus proche de l'incident | <i>Etant donné qu'il y a un incident</i> <i>Quand le lieu de l'incident est renseigné</i> <i>Et que la pathologie est sélectionnée</i> <i>Alors le système m'affiche l'hôpital de la ville sélectionné</i> |
| | <i>Etant donné qu'il y a un incident</i> <i>Quand le lieu de l'incident est renseigné</i> <i>Et que la pathologie n'est pas sélectionnée</i> <i>Alors un message d'erreur s'affiche</i> |
| Réserver un lit d'hôpital selon la disponibilité | <i>Etant donné qu'il y a un incident</i> <i>Quand il y a un hôpital qui est proposé</i> <i>Et qu'on a saisi le nom et prénom du patient</i> <i>Alors le système déclenche un événement pour réserver un lit en fonction du nom du patient</i> |
| | <i>Etant donné qu'il y a un incident</i> <i>Quand il y a un hôpital qui est proposé</i> <i>Et qu'il n'y a pas de place disponible sur les hôpitaux les plus proches de l'incident</i> <i>Alors un message en rouge s'affiche « il n'y a pas de lit disponible dans cet hôpital »</i> |

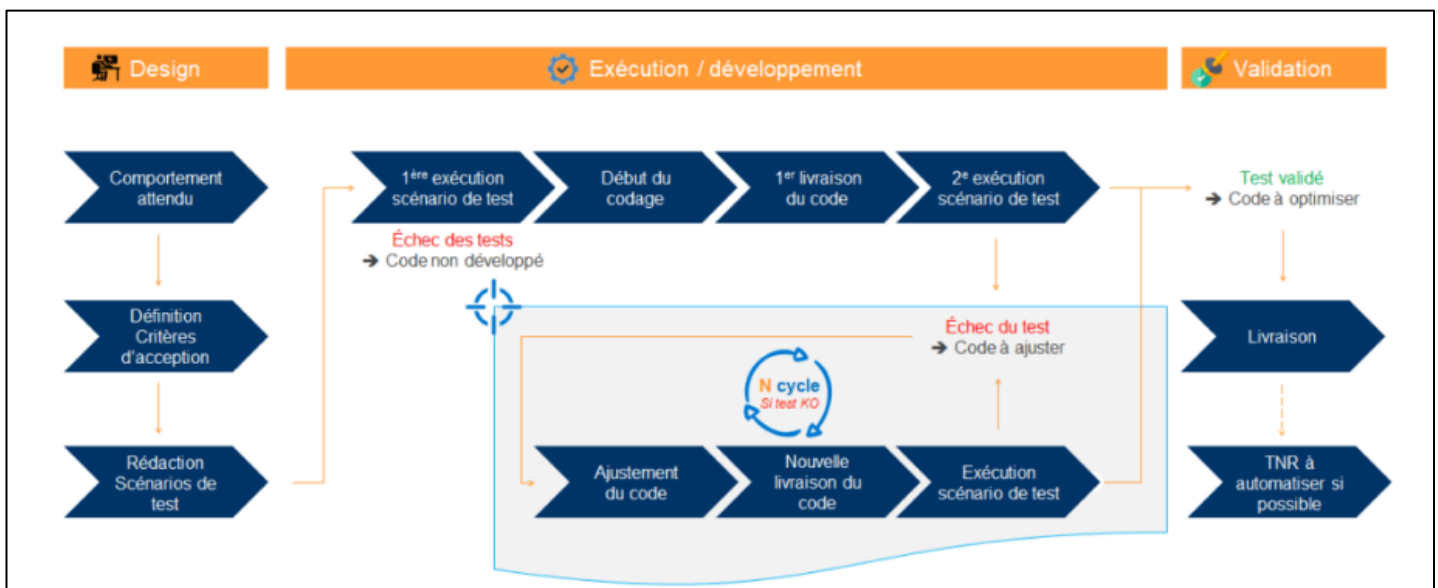
D. Les outils et dépendances utilisés

| | Outils/technologies |
|--|-----------------------------|
| Développement backend et frontend du PoC | IDE Eclipse |
| | Java |
| | Maven |
| | Spring boot |
| | JPA |
| | Proxy |
| | Spring api Gateway |
| | Thymleaf |
| Base de données | SQL Server |
| Test du PoC | JUnit |
| | Jenkins |
| Documentations | Swagger (documentation API) |
| | Jacoco (rapport des tests) |

VI. Passage en revue des opportunités et des solutions (Phase E) et planning de migration (Phase F)

Une fois l'architecture effectivement décrite et ciblée, les phases E et F sont le moment de définir concrètement les étapes à venir. Quels sont les changements apportés par cette refonte architecturale ?

A. La phase de test



Nous avons automatisé tous les tests qui seront développés dans ce projet.

Les différentes étapes à réaliser :

- L'écriture de scénario
- Développement des tests (test unitaire et d'intégration)
- Exécution des tests
- Livraison

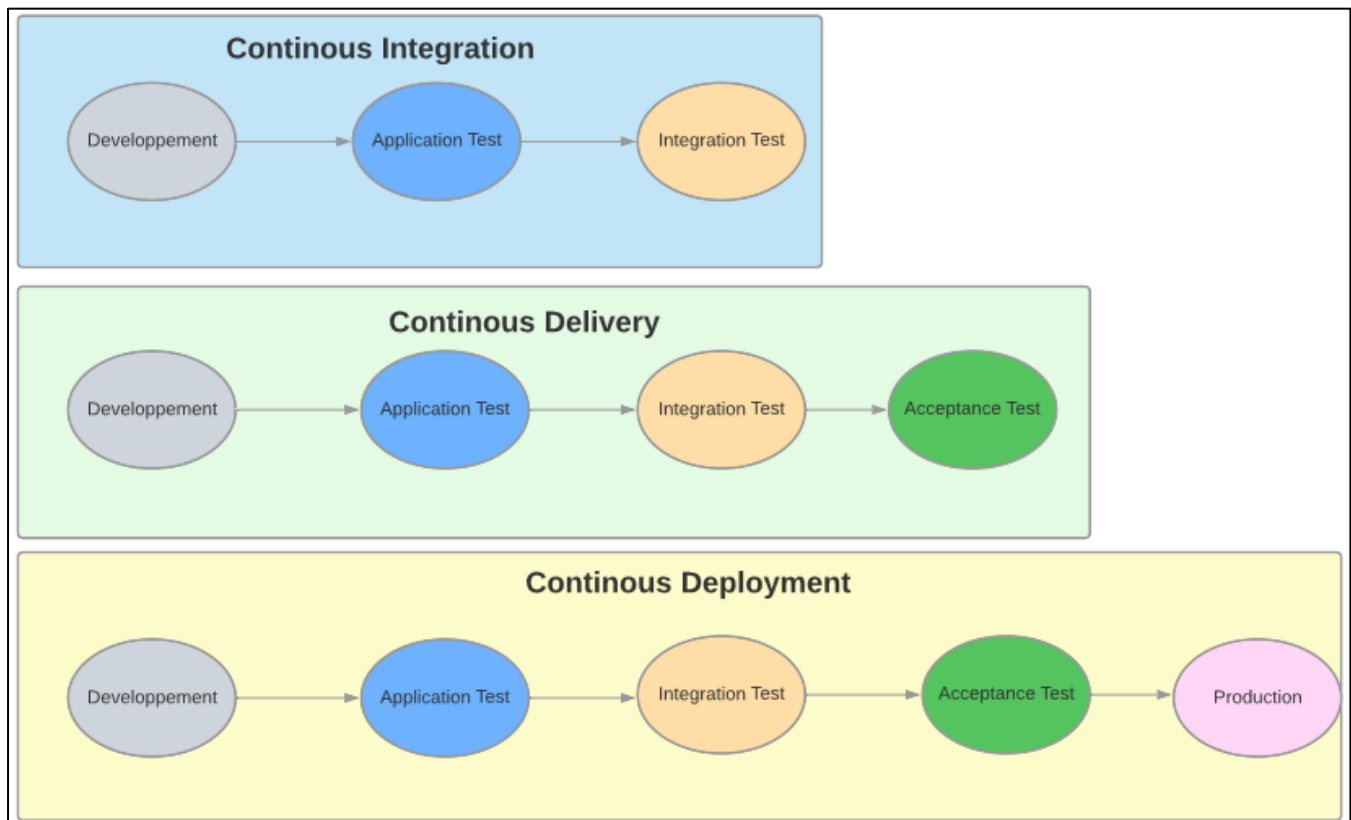
B. Les différents tests automatisés mis en place pour ce PoC

Les tests automatisés seront à réaliser à chaque évolution de l'outil. Ils seront maintenus et lancés dans le cadre de notre organisation CI/CD à chaque mise à jour du code.

Ce plan de test va couvrir ces catégories de tests :

- **L'intégration continue** : consiste à intégrer des modifications de code dans un dépôt plusieurs fois par jour.

- **Le déploiement continu** : la **livraison** continue automatise les intégrations de code, tandis que le **déploiement** continu publie automatiquement les versions finales aux utilisateurs finaux.



Pour plus de détails consulter le document « Stratégie de test ».

VII. Mise en œuvre de la gouvernance (Phase G)

Cette phase va permettre de définir et organiser le suivi des projets par l'architecture.

A. Suivi du projet

Nous avons travaillé sur le modèle agile (Scrum). Cependant, nous envisageons de faire des réunions quotidiennes lors de la mise en place de l'application. Nous serons donc amenés à solliciter l'équipe métier afin de vous faire des démonstrations des applications en cours de mise en place, afin d'avoir un retour des utilisateurs. Nous avons utilisé l'outil Jira pour la gestion des sprints.

1. Point journalier (Daily)

Nous avons fait des réunions quotidiennes afin de faire un point sur l'avancée des tâches de chacun. Cette réunion permettait d'aborder les difficultés rencontrées, les points de blocages et l'avancement concernant chaque tâche. Il permettait de suivre et de structurer le projet.

2. Point hebdomadaire

Une seconde réunion sera réalisée une fois par semaine. Cette réunion rassemble toute l'équipe du projet. Cette réunion nous permet de faire un point sur l'avancée du projet, ainsi de pouvoir respecter les délais du projet.

D'autres réunions sont réalisées à la fin des sprints par exemple pour faire des démonstrations des applications en cours de mise en place, afin d'avoir un retour concret. Cela permet d'ajuster à temps les fonctionnalités du projet.

Extrait du Backlog de l'outil Jira :

The screenshot displays the Jira Software interface for a project named 'PoCMedHead+' under the 'Service gestion-hopital' project. The interface is organized into three main columns representing the workflow stages: 'A FAIRE 2 TICKETS', 'EN COURS 2 TICKETS', and 'FINI 4 TICKETS'. Each ticket card includes a description, a status (checked), and a priority (POC-2, POC-5, POC-6, POC-7, POC-1, POC-3, POC-4, POC-8). The 'FINI' column also shows a green checkmark indicating completion. The left sidebar contains navigation options for 'PLANIFICATION' (Feuille de route, Backlog, Tableau) and 'DÉVELOPPEMENT' (Code, Pages de projet, Ajouter un raccourci, Paramètres du projet). The top navigation bar includes 'Votre travail', 'Projets', 'Filtres', 'Tableaux de bord', 'Personnes', 'Appli', and a 'Créer' button. A search bar is located in the top right corner.

| Statut | Description | POC |
|----------|---|-------|
| A FAIRE | Fonctionnalité réduire le nombre de lit | POC-2 |
| A FAIRE | Tester la fonctionnalité réduire le nombre de lit | POC-5 |
| EN COURS | Fonctionnalité réserver un lit | POC-6 |
| EN COURS | Tester la fonctionnalité réserver un lit | POC-7 |
| FINI | Fonctionnalité rechercher un hôpital | POC-1 |
| FINI | Fonctionnalité afficher les spécialisations | POC-3 |
| FINI | Tester la fonctionnalité de recherche d'hôpital | POC-4 |
| FINI | dév service configuration | POC-8 |