



Enoncé des travaux d'architecture

Table des Matières

1. Information sur le document	3
2. Liste de distribution	3
3. Historique des versions du document	3
4. Information sur le projet	3
4.1. Résumé du projet	3
4.2 Processus du projet	3
4.3 Jalons du projet	3
4.4 Conditions d'utilisation	4
4.5 Hypothèses générales	4
5. Découpage des phases	4
5.1 Phase n° 1 : Etude et développement de l'API-REST	4
5.1.1 Etude des BDDs	4
5.1.2 Outils	4
5.2 Phase n° 2 : Etude et développement de l'outil de test à distance	5
5.2.1 Tests de connections	5
5.1.2 Outils	5
5.3 Phase n° 3 : Le pipeline CI/CD	5
5.2.1 Tests de connections	5
5.1.2 Outils	5
6. Etude et méthodologie Togaf	5
6.1 Architecture cible	5
6.2 Phase cyclique ADM du Projet	6
6.2 Contenu de l'architecture du Projet	6

Table des figures

Figure 1 : Information sur le document	3
Figure 2 : Liste de distribution	3
Figure 3 : Historique des versions du document	3
Figure 4 : Exemple d'hôpitaux	4

1. Information sur le document

Ce document a pour vocation de lister les hypothèses de développement d'une preuve de concept pour le sous système d'intervention d'urgence.

Nom du projet	Preuve du concept
Préparé par :	Frédéric Pichot
N° de version du document :	0.1
Titre :	Énoncé des travaux d'architecture
Date de version du document :	16/12/2021

Figure 1: Information sur le document

2. Liste de distribution

De	Date	Phone/Fax/Email
Ash Kara	27/10/2021	Email
Chris Pike	27/10/2021	Chat

Figure 2 : Liste de distribution

3. Historique des versions du document

Numéro de Version	Date de la version	Revu par	Description	Nom du fichier
1	27/10/2021	Frédéric Pichot	Création du document	Énoncé des travaux d'architecture

Figure 3 : Historique des versions du document

4. Information sur le projet

4.1. Résumé du projet

Lors de la recherche d'un lieu où amener un malade dont la pathologie nécessite de trouver rapidement une place dans la limite d'un périmètre, les outils actuels sont imprécis ou trop lents.

L'envergure de la transformation et la nécessiter de maintenir l'existant en état, le groupe de recherche a décidé de développer une Preuve de Concept (PoC), qui aidera à valider le lancement conséquent de la transformation de l'outil actuel.

4.2 Processus du projet

Le processus du projet tend à prouver que l'idée du système se montre robuste et solide, même lors des fortes sollicitations. Il permet des évolutions aux grés des besoins, mais elles seront automatiquement vérifiées et testées de façon régulière avant qu'elles modifie l'équipement en fonction.

4.3 Jalons du projet

Dans un délais de trois mois, il devra être développé une API REST déposée chez un hébergeur dédié au versionning (Github, Gitlab...). Elle sera dotée d'un ensemble de tests, unitaire et d'intégration couvrant tous les comportements clés du système.

Dans un second temps, l'équipe devra coder un outil de tests, surveillé lui-même par un ensemble d'auto évaluation, qui sera destiné à questionner l'API.

Pour finir il sera confié les cycles d'auto-évaluation permanents et séquentiels des deux projets, par l'intermédiaire d'un pipeline CI/CD.

4.4 Conditions d'utilisation

La PoC a comme utilité c'elle de démontrer physiquement la pertinence des hypothèses et de leurs faisabilité. Elle devra être utilisée dans un cadre de démonstration ou à comme support pédagogique. Elle n'a pas la pertinence de remplacer l'équipement présent usité à des même fins.

Pour autant, un ensemble de tests et de class pourront être totalement intégré dans une épreuve fonctionnelle.

4.5 Hypothèses générales

La plupart des hypothèses sont issues du travail de recherche et de collecte que la situation de défaut exigeait.

Elles peuvent-être retrouvées dans le document rédigé par Ms. Chris Pike et Ash Kara sous l'intitulé « Hypothèse », dont voici un aperçu.

Les exigences suivantes ont été convenues lors de la définition de cette hypothèse :

- Fournir une API RESTful qui tient les intervenants médicaux informés en temps réel sur : le lieu où se rendre et ce qu'ils doivent faire.
- S'assurer que toutes les données du patient sont correctement protégées.
- S'assurer que la PoC est entièrement validée 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.
- S'assure que la PoC peut 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.
- S'assurer que les équipes de développement chargées de cette PoC sont en mesure de l'utiliser comme un jeu de modules de construction pour d'autres modules.

5. Découpage des phases

La domiciliation du projet sera sous « localhost:9010 ».

La sauvegarde des deux éléments sera hébergée sur GitHub.

5.1 Phase n° 1 : Etude et développement de l'API-REST

5.1.1 Etude des BDDs

Plusieurs tables seront présentes dans la version finale (hôpitaux, patients, spécialités) pour des raisons de planning il sera retenu l'emploi de la première qui sera nommé « Hospital ». Pour des raisons de taille et de vélocité elle sera une table éphémère de type « H2 ».

Celle-ci sera dotée de champs latitude et longitude qui seront utilisés pour calculer la distance entre le patient et les hôpitaux pour estimer celui le plus à même de le recevoir.

Pour neutraliser l'aspect « Spécialités », il sera enregistrer plusieurs occurrence du même hôpital.

5.1.2 Outils

Le langage utilisé sera JAVA, le codage se fera dans l'environnement Spring-boot. Le cadre du développement sera sous le model Maven.

5.2 Phase n° 2 : Etude et développement de l'outil de test à distance

L'enregistrement du projet sera sous « localhost:9010 ».

Le versionning des deux éléments sera hébergé sur GitHub.

5.2.1 Tests de connexions

L'utilisation du protocole « HTTPClient », nécessitera une version postérieure à JAVA 8.

5.1.2 Outils

Le langage utilisé sera JAVA 16, le codage se fera dans l'environnement Spring-boot. Le cadre du développement sera sous le model Maven.

5.3 Phase n° 3 : Le pipeline CI/CD

La sauvegarde des deux éléments seront hébergés sur GitHub.

5.2.1 Tests de connexions

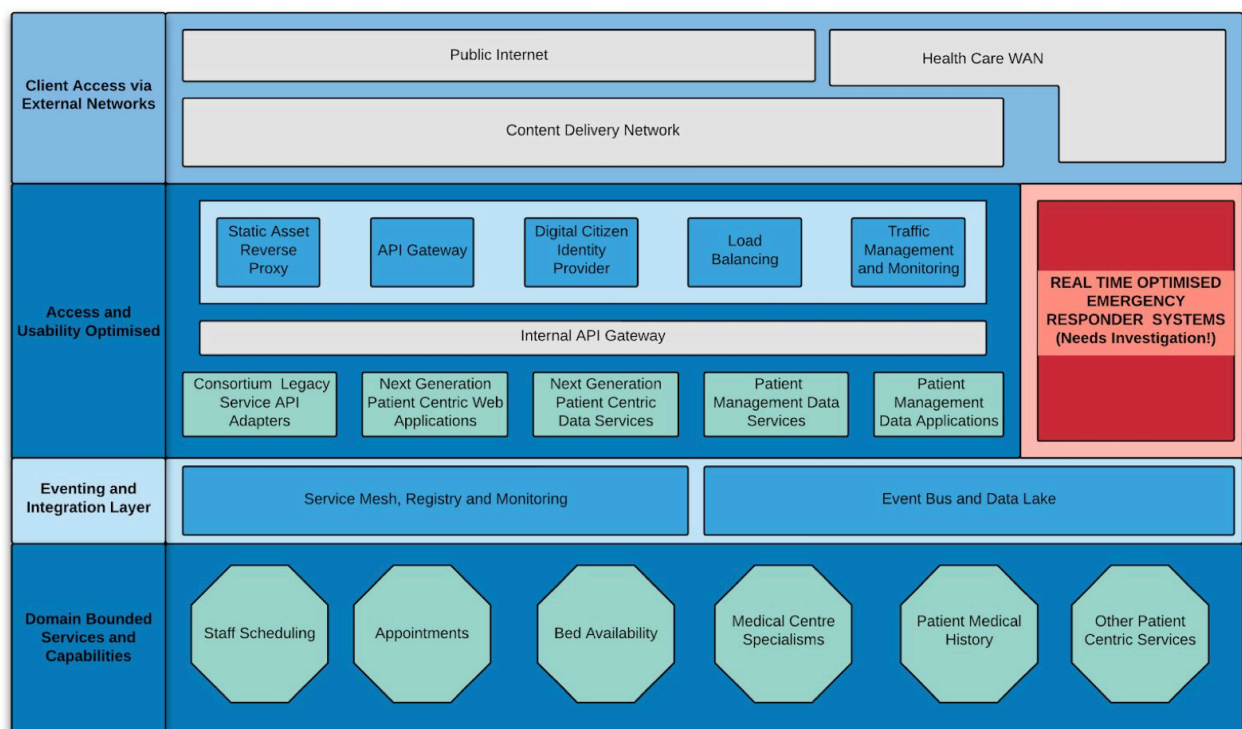
L'utilisé le protocole « HTTPClient », nécessitera d'utiliser une version postérieure à JAVA 8.

5.1.2 Outils

Le langage utilisé sera JAVA 16, le codage se fera dans l'environnement Spring-boot. Le cadre du développement sera sous le model Maven.

6. Etude et méthodologie Togaf

6.1 Architecture cible



6.2 Phase cyclique ADM du Projet

Il est possible que le premier tour de roue arrive à une étape intermédiaire, dans ce cas il sera nécessaire d'effectuer une seconde itération en débutant de la situation d'arrivée de la précédente.

Phase ADM	Entrée/Sortie	Activités
Préliminaire & gestion des exigences		
A. Vision de l'Architecture	Entrant	Arrêter la liste des hypothèses et ordonnancer les en fonction de l'avancement de la roue ADM
B. Architecture du Business C. Architectures des systèmes d'Information D. Architecture Technique	Entrant	Développer les architectures dans les quatre domaines : <ul style="list-style-type: none"> • Métier • Système d'information - Application • Système d'information - Données • Technologique
E. Opportunités et Solutions	Entrant	Mise en regard les hypothèses et les tests
F. Planification de la migration	Entrant/Sortant	Liste et ordonnancement des modules de construction de la PoC, mise en production du pipeline CI/CD
G. Gouvernance de la mise en oeuvre		Evaluation de la PoC en veillant à ce quelle s'inscrive dans la conformité de l'architecture
H. Gestion du changement d'architecture	Entrant	Les mises à jour seront accompagnées d'un historique d'intégration et de leur approbation

6.2 Contenu de l'architecture du Projet

L'approche Togaf est une base de travail complet pour appréhender les problématiques diverses, cependant, il sera nécessaire de sélectionner des artefacts en adéquation avec le projet.

Phase d'existence	Entrée/Sortie	Commentaires
A. Vision de l'Architecture	Entrant	Défini les principes d'architecture, inhérents aux hypothèses
B. Architecture du Business C. Architectures des systèmes d'Information D. Architecture Technique	Entrant	Mettre à jour, si nécessaire, les quatre domaines en tenant compte du document de définition de l'architecture: <ul style="list-style-type: none"> • Métier • Système d'information - Application • Système d'information - Données • Technologique (voir hypothèse du PoC)
E. Opportunités et Solutions	Entrant	Exploration et révision si nécessaire
F. Planification de la migration	Entrant/Sortant	Le projet terminé contiendra un API-Rest fonctionnelle, testée et éprouvée par un ensemble d'outils de tests. Une documentation détaillée sera envisagée, intégrée dans la feuille de route avant la mise en production.