



ENONCE DES TRAVAUX D'ARCHITECTURE TOGAF

Projet PoC Sous-Système

Intervention Urgence

Table des matières

1 METHODE TOGAF	3
2 OBJECTIFS ET EXIGENCES	5
2 ANALYSES ET DIAGNOSTICS	6
3 PASSAGE EN REVUE DES OPPORTUNITÉS ET DES SOLUTIONS (PHASE E) ET PLANNING DE MIGRATION (PHASE F)	9
4 MISE EN ŒUVRE DE LA GOUVERNANCE (PHASE G)	10
4-1 Suivi du projet	10
4-2 Point journalier (Daily scrum)	11
4-3 Point hebdomadaire	11

1 METHODE 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.

1-1 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 TOGAF](#)

- 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

1-2 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).

1-3 Architecture métier (Phase B)

Analyse de l'architecture de base

A-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.

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.

2 OBJECTIFS ET EXIGENCES

Les 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.

2 ANALYSES ET DIAGNOSTICS

2-1 Exigences / technologies manquant dans la solution Solutions initiale

Exigences, Technologie manquante dans la solution actuelle	Solutions apportées
Gestion de montée en charges : - Obtenir un temps de réponse de moins de 200 millisecondes avec une charge de travail allant jusqu'à 800 requêtes par seconde)	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
Gestion des évolutions futures de la solution - Permettre des améliorations et modifications futures du système	Une architecture avec un faible couplage - Une indépendance des modules (autonomie) - Une possibilité de réutilisations des modules développés - Une architecture facile à maintenir

2-3 Analyse de l'architecture cible

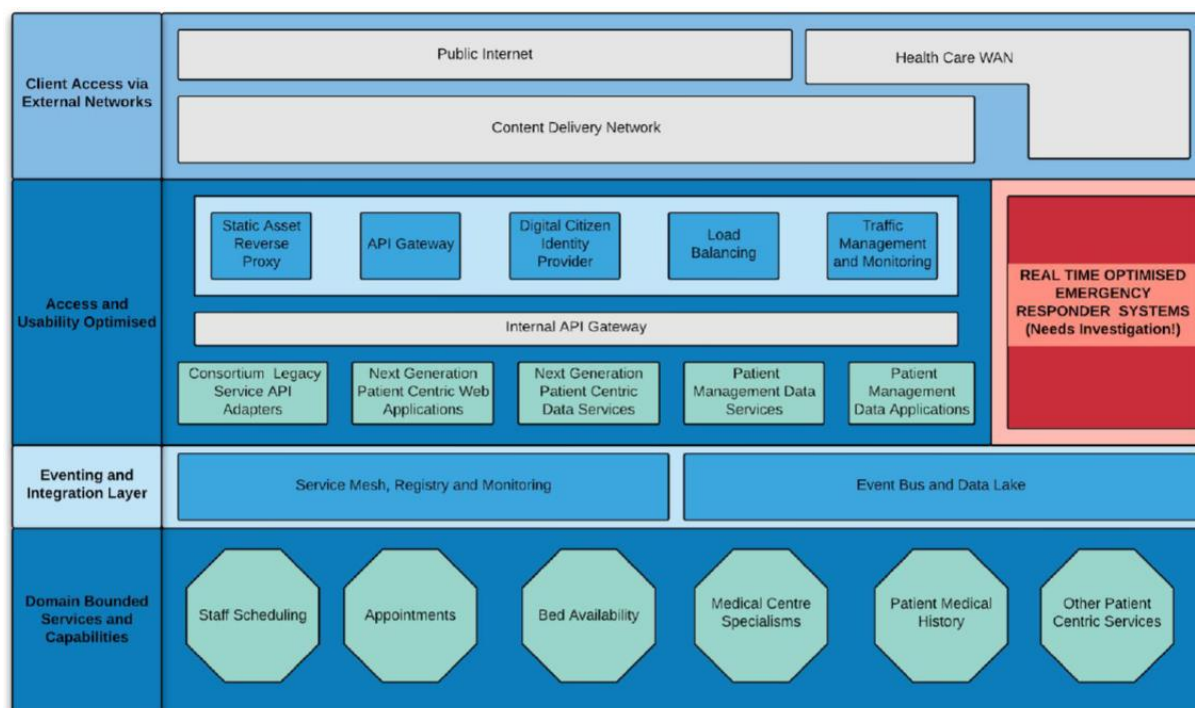


Figure N ° 1représentant l'architecture proposée par l'entreprise

A partir de l'ensemble des éléments que nous avons traité, l'architecture microservice 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 micros-services développés sont parfaitement autonomes. En effet, chaque micro-service à sa propre base de données et pour faciliter la communication les uns avec les autres, chacun à son propre API REST.

2-4 Micros-service mise en place dans le PoC du système d'intervention d'urgence

Services	Objectifs
Application web	L'interface web qui permet aux clients de gérer la gestion des hôpitaux plus facilement
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 microservices pour faciliter et surveiller les échanges
Configuration	Ce service permet de centraliser toutes les configurations nécessaires pour l'ensemble des microservices
Configuration	Ce service permet de centraliser toutes les configurations nécessaires pour l'ensemble des microservices.

2-5 Les différentes fonctionnalités des Micro-services

Fonctionnalités	Description
Saisir le lieu de l'incident	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

2-6 Les Users stories

Fonctionnalités	Descriptions	
Chercher et suggérer l'hôpital le plus proche de lieu de l'incident	Étant donné qu'il y a un incident	Étant donné qu'il y a un incident
	Quand le lieu de l'incident est renseigné	Quand le lieu de l'incident est renseigné
	Et que la pathologie est sélectionnée	Et que la pathologie n'est pas sélectionnée
	Alors le système m'affiche	Alors un message d'erreur s'affiche

	l'hôpital de la ville sélectionné	
Réserver un lit d'hôpital selon la disponibilité	<p>Etant donné qu'il y a un incident</p> <p>Quand il y a un hôpital qui est proposé</p> <p>Et qu'on a saisi le nom et telephone du patient</p> <p>Alors le système déclenche un évènement pour réserver un lit en fonction du nom du patient Et son telephone</p>	<p>Etant donné qu'il y a un incident</p> <p>Quand il y a un hôpital qui est proposé</p> <p>Et qu' il n'y a pas de place disponible sur les hôpitaux les plus proches de l'incident</p> <p>Alors un message en rouge s'affiche « il n'y a pas de lit disponible dans cet hôpital »</p>

3 PASSAGE EN REVUE DES OPPORTUNITÉS ET DES SOLUTIONS (PHASE E) ET PLANNING DE MIGRATION (PHASE F)

3-1 La phase de test

Nous avons automatisé tous les tests qui seront développer dans ce projet. Les différentes étapes à réaliser :

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

3-2 Les différents tests automatisés misent 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 chque mise à jour du code.

Le 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.

4 MISE EN ŒUVRE DE LA GOUVERNANCE (PHASE G)

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

4-1 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.

4-2 Point journalier (Daily scrum)

- 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.

4-3 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.