

Exigences pour le développement d'une preuve de concept pour le système d'intervention d'urgence en temps réel

Déclaration

Nous pensons que la mise en œuvre d'une preuve de concept pour le système d'intervention d'urgence en temps réel par l'équipe d'architecture métier du consortium MedHead permettra :

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

Nous saurons que nous avons réussi si :

- plus de 90 % des cas d'urgence sont acheminés vers l'hôpital compétent le plus proche du réseau ;
- le temps moyen de traitement d'une urgence passe de 18,25 minutes (valeur actuelle) à 12 minutes (valeur souhaitée) ;
- nous obtenons un temps de réponse de moins de 200 millisecondes avec une charge de travail allant jusqu'à 800 requêtes par seconde, par instance de service ;
- la mise en œuvre respecte les normes imposées ;
- la mise en œuvre est terminée dans le délai imparti.

Scénario

Le système d'intervention d'urgence en temps réel est destiné à suggérer l'hôpital le plus proche offrant un lit disponible et possédant la spécialisation attendue (voir les [Données de référence sur les spécialités](#)) sur la base d'une banque de données d'informations récentes sur les hôpitaux.

Par exemple, SUPPOSONS trois hôpitaux, comme suit :

Hôpital	Lits disponibles	Spécialisations
Hôpital Fred Brooks	2	Cardiologie, immunologie
Hôpital Julia Crusher	0	Cardiologie
Hôpital Beverly Bashir	5	Immunologie, neuropathologie, diagnostic

ET un patient nécessitant des soins en cardiologie,

QUAND un patient demande des soins en cardiologie ET que l'urgence est localisée près de l'hôpital Fred Brooks,

ALORS l'hôpital Fred Brooks devrait être proposé,

ET un événement devrait être publié pour réserver un lit.

Exigences

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

- Fournir une API RESTful qui renvoie le lieu où se rendre :
 - La technologie Java est imposée par le consortium,
 - L'API doit pouvoir s'inscrire à terme dans une architecture microservice ;
- Fournir une interface graphique qui consomme l'API :
 - Une simple page permettant de sélectionner une spécialité et de saisir la localisation est suffisante,
 - Le consortium impose d'utiliser l'un des frameworks Javascript/Typescript courant du marché : Angular, React, VueJS ;
- S'assurer que toutes les données du patient sont correctement protégées ;
- S'assurer que votre PoC est entièrement validée avec des tests reflétant la pyramide de tests (tests unitaires, d'intégration et E2E) :
 - L'API doit être éprouvée avec des tests de stress pour garantir la continuité de l'activité en cas de pic d'utilisation ;

- S'assurer 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) ;
- S'assurer que les équipes de développement chargées de cette PoC sont en mesure de l'utiliser comme un bloc de construction pour d'autres modules ou tout du moins comme un modèle à suivre ;
- Le code est versionné à l'aide d'un workflow Git adapté ;
- La documentation technique de la PoC sera formalisée dans un fichier readme.md respectant le format markdown et contiendra au minimum :
 - les instructions pour l'exécution des tests,
 - le fonctionnement du pipeline,
 - le workflow Git retenu (ce dernier sera détaillé pour qu'il soit réutilisable par les équipes) ;
- Un document de reporting sera rédigé indiquant :
 - une justification des technologies utilisées,
 - une justification du respect des normes et des principes (exemple : norme RGPD, principe d'architecture microservice, etc.),
 - les résultats et les enseignements de la PoC.