

Validation des principes d'architecture



Projet allocation de lits d'hôpital pour les urgences

Table des matières

Introduction.....	3
Résumé des principes.....	4
Principes d'architecture métier.....	6
Principe A1 : Primauté des principes.....	6
Principe A2 : Maximiser les avantages pour l'entreprise.....	6
Principe A3 : Conformité aux lois et aux règlements.....	6
Principe A4 : Adhésion au serment d'Hippocrate à tous les niveaux.....	6
Principes de l'architecture informatique.....	8
Principe B1 : Continuité des activités des systèmes critiques pour les patients.....	8
Principe B2 : Clarté grâce à une séparation fine des préoccupations.....	9
Principe B3 : Intégration et livraison continues.....	10
Principe B4 : Tests automatisés précoces, complets et appropriés.....	10
Principe B5 : Sécurité de type « shift-left ».....	11
Principe B6 : Possibilité grâce à des fonctionnalités déclenchées par les événements.....	11
Méthodologie architecturale et principes de processus.....	13
Principe C1 : Personnalisation de l'ADM TOGAF 9.2.....	13
Principe C2 : Référentiel d'architecture centralisé et organisé comme source de référence.....	13
Principe C3 : Normes ouvertes pour garantir des normes élevées.....	13
Principe C4 : Favoriser une culture de “learning” avec des preuves de concept, des prototypes et des Spike.....	13

Introduction

Ce document a pour objectif de reprendre et de valider les différents principes architecturaux énoncés dans le document « Principes de l'architecture », présent dans le dépôt, dans le sous-dossier Artefact.

Il se focalisera sur les principes vérifiables dans le cadre d'une preuve de concept uniquement, en respectant le principe C4, partie 4 : « *Les principes de base de l'ingénierie, de la livraison et des tests ne doivent pas être assouplis pour l'architecture de la PoC.* », décrite de la manière suivante :

« *La validation de principe doit viser spécifiquement à respecter les principes suivants :*

- *Principe B1 : Continuité des activités des systèmes critiques pour les patients*
- *Principe B2 : Clarté grâce à une séparation fine des préoccupations*
- *Principe B3 : Intégration et livraison continues*
- *Principe B4 : Tests automatisés précoces, complets et appropriés »*

Cependant, une prise en compte des autres principes sera, si possible, effectuée.

Résumé des principes

La liste des principes et leur déclaration sont présents dans le dépôt architectural, dans le sous-dossier artefacts.

- Principes d'architecture métier
 - Principe A1 : Primauté des principes
 - Principe A2 : Maximiser les avantages pour l'entreprise
 - Principe A3 : Conformité aux lois et aux règlements
 - Principe A4 : Adhésion au serment d'Hippocrate à tous les niveaux
- Principes de l'architecture informatique (système, données, solutions, sécurité et opérations)
 - Principe B1 : Continuité des activités des systèmes critiques pour les patients
 - Principe B2 : Clarté grâce à une séparation fine des préoccupations
 - Principe B3 : Intégration et livraison continues
 - Principe B4 : Tests automatisés précoces, complets et appropriés
 - Principe B5 : Sécurité de type « shift-left »
 - Principe B6 : Possibilité d'extension grâce à des fonctionnalités déclenchées par les événements
- Méthodologie architecturale et principes de processus
 - Principe C1 : Personnalisation de l'ADM TOGAF 9.2
 - Principe C2 : Référentiel d'architecture centralisé et organisé comme source de référence
 - Principe C3 : Normes ouvertes pour garantir des normes élevées
 - Principe C4 : Favoriser une culture de “learning” avec des preuves de concept, des prototypes et des Spike
 - Fournir une hypothèse pour chaque preuve de concept
 - Isoler les preuves de concept des données et des systèmes de production
 - Souplesse au niveau de la conformité aux principes, mais prise en compte des conséquences en cas de non-respect de ces mêmes principes

- Les principes d'ingénierie, de livraison et de test de base ne doivent pas être assouplis pour l'architecture des preuves de concepts

Principes d'architecture métier

Principe A1 : Primauté des principes

Déclaration :

« Les principes énoncés ici s'appliquent à tous les membres du Consortium, que nous appellerons collectivement l'entreprise. »

Mise en application :

Les membres du consortium ont accès au code de la PoC ainsi qu'au dépôt architectural, sans distinction d'entités interne autre que celle du consortium.

Principe A2 : Maximiser les avantages pour l'entreprise

Déclaration :

« Les décisions d'architecture et de conception général sont prises pour fournir un avantage maximum à l'entreprise dans son ensemble, dans le cadre des efforts entrepris pour améliorer les soins dispensés aux patients touchés par ces décisions. »

Mise en application :

Principe A3 : Conformité aux lois et aux règlements

Déclaration :

« Le système d'information, les processus métier et les livrables doivent être conformes à toutes les lois, politiques et réglementations pertinentes. »

Mise en application :

La PoC respecte les lois et règlements en vigueur par la mise en place d'une conformité avec les lois du Règlement Général sur la Protection des Données (RGPD) par le non enregistrement, dans le système d'allocation de lit d'hôpitaux, de toutes données liées au patient.

Principe A4 : Adhésion au serment d'Hippocrate à tous les niveaux

Déclaration :

« En tant qu'entreprise à visé médical dont le but est d'améliorer les soins dispensés aux patients, toutes les décisions organisationnelles doivent adhérer au serment d'Hippocrate (« d'abord ne pas nuire, ensuite soigner ») en ce qui concerne les soins prodigués par tous les membres du Consortium et leur personnel interne. »

Mise en application :

Le projet s'intègre dans une optique d'assurer une meilleure prise en charge du patient. La PoC est, entre autres choses, là pour démontrer le gain de temps de prise en charge du patient par l'obtention d'un hôpital ayant une disponibilité en lit dans la spécialité demandée.

Principes de l'architecture informatique

Principe B1 : Continuité des activités des systèmes critiques pour les patients

Déclaration :

« Les opérations essentielles à la santé des patients, ainsi que les autres pratiques de soin, doivent être assurées malgré les interruptions du système. »

Mise en application :

La PoC répond à ce principe via un plan de test exhaustif, présent dans le dépôt d'architecture, et comprenant entre autres des tests incrémentiels de performance, la possibilité de redondance et de maintenabilité et une première ébauche d'un plan de reprise. <TODO : revoir ?>

Ce plan de tests montre le respect des déclarations d'hypothèse suivantes :

- que plus de 90 % des cas d'urgence sont acheminés vers l'hôpital compétent le plus proche du réseau
- que le temps moyen de traitement d'une urgence passe de 18,25 minutes (valeur actuelle) à 12,00 minutes (valeur souhaitée)
- que 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
- que la mise en œuvre explique les normes qu'elle respecte et pourquoi
- que les instructions pour mettre en production la PoC sont fournies
- que la mise en œuvre est terminée dans le délai imparti.

De plus, des optimisations ont été faites au niveau du code source, avec une parallélisation de l'appel de l'API au niveau du service externe :

Itération	Avec parallélisation	Sans parallélisation
Lancement 1	625	791
Lancement 2	175	298
Lancement 3 et +	80, 79, 82	370, 233, 238

Nous observons une diminution des temps de traitement suite à la parallélisation. Le premier lancement est environ 21 % plus rapide, le second appel 41 %, les derniers appels près de 65 % plus rapide. Ces résultats s'expliquent soit par l'utilisation du cache navigateur, soit d'un cache chez les prestataires de l'API.

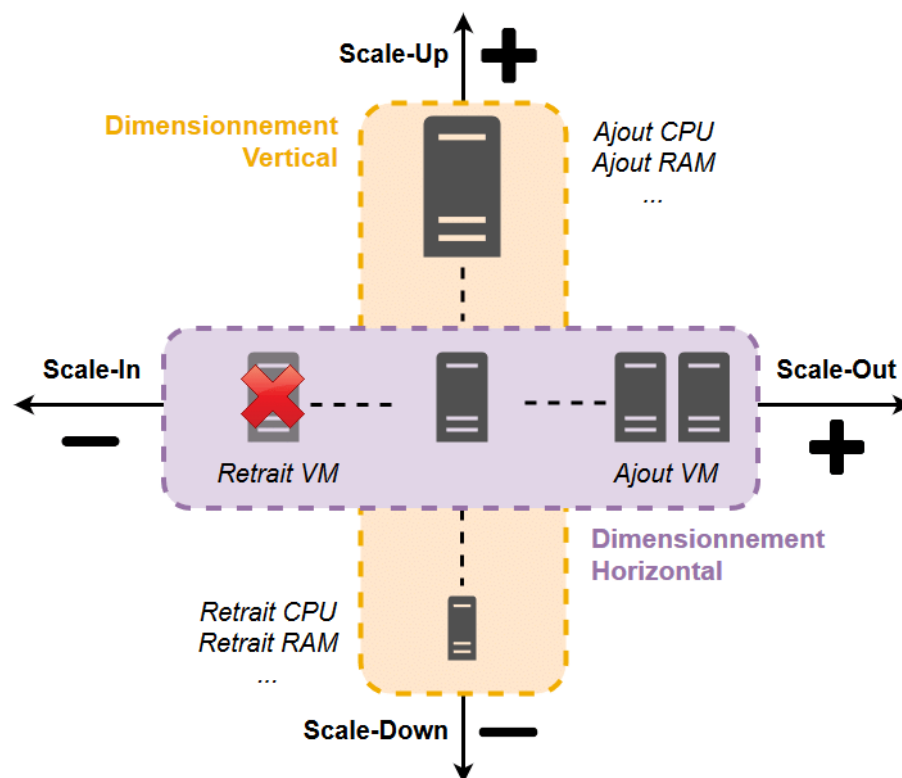
Principe B2 : Clarté grâce à une séparation fine des préoccupations

Déclaration :

« Il faut éviter de regrouper ensemble des responsabilités disparates. Il faut éviter les systèmes centralisés. »

Mise en application :

La PoC utilise SpringBoot pour le développement de micro service. Une architecture en micro service permet de séparer les différents composants applicatifs et de répondre aussi bien a un besoin de décentralisation, les micros-services et leur base de données pouvant être hébergé chez plusieurs prestataires différents, comme à un besoin d'adaptabilité à la charge, les micros-services permettant un dimensionnement horizontal de l'infrastructure. → Parler MVC/multicouche



Principe B3 : Intégration et livraison continues

Déclaration :

« *L'intégration et la livraison continues de petits changements incrémentiels sont favorisées par rapport aux temps de cycle lents et aux intégrations majeures.* »

Mise en application :

La mise en place de pipeline CI/CD est effectué, avec une exécution de tests à chaque merge avec la branche « *main* ». Des journaux d'exécution sont accessibles dans le dépôts Github, dans la partie « *Actions* ».

Un fonctionnement détaillé des pipelines CI/CD est présent dans le document *Exigence de la PoC*, qui peut être retrouvé dans le dépôt d'architecture.

Principe B4 : Tests automatisés précoces, complets et appropriés

Déclaration :

« *Les applications doivent être construites à l'aide de tests automatisés qui garantissent la fiabilité à la fois fonctionnelle et non fonctionnelle de la mise en œuvre.* »

Mise en application :

Le principe des tests automatisé et leur mise en place est décrit de manière plus exhaustive dans le document « **Document de stratégie de tests** » présent sur dans le dépôt d'architecture.

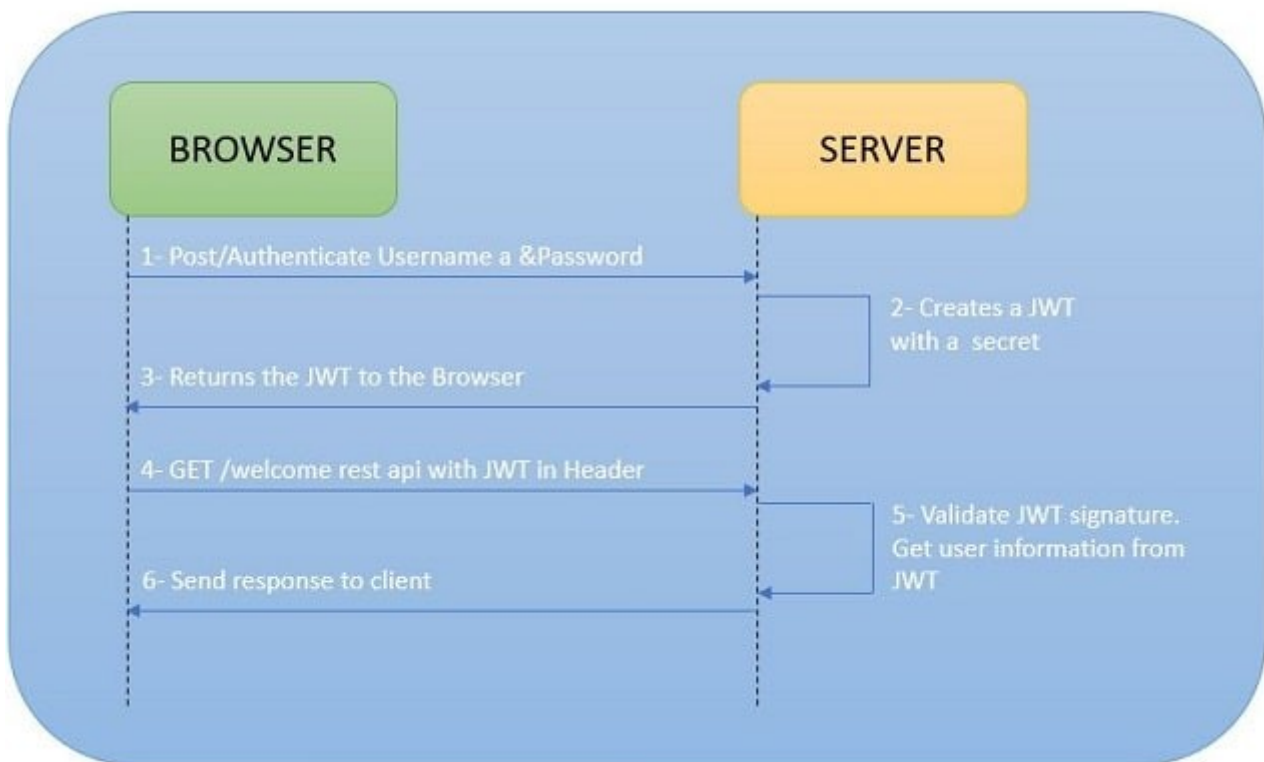
Principe B5 : Sécurité de type « shift-left »

Déclaration :

« *Le risque global de sécurité de la plateforme est réduit en spécifiant et en respectant les exigences de sécurité dès le début de chaque incrément.* »

Mise en application :

La mise en application des principes « shift left » de sécurité n'a pas été gérée dans le POC. En effet, les différents mots de passes utilisés ou clé d'API sont stockés en clair dans le fichier `application.properties`. Hors preuve de concept, une authentification forte est à prévoir. De plus, l'usage de technologies comme Auth2 ou des tokens JWT fourni par Spring Boot Secure vont permettre d'améliorer la sécurité dès le développement du projet. → TODO : HTTPS/protocoles sécurisés



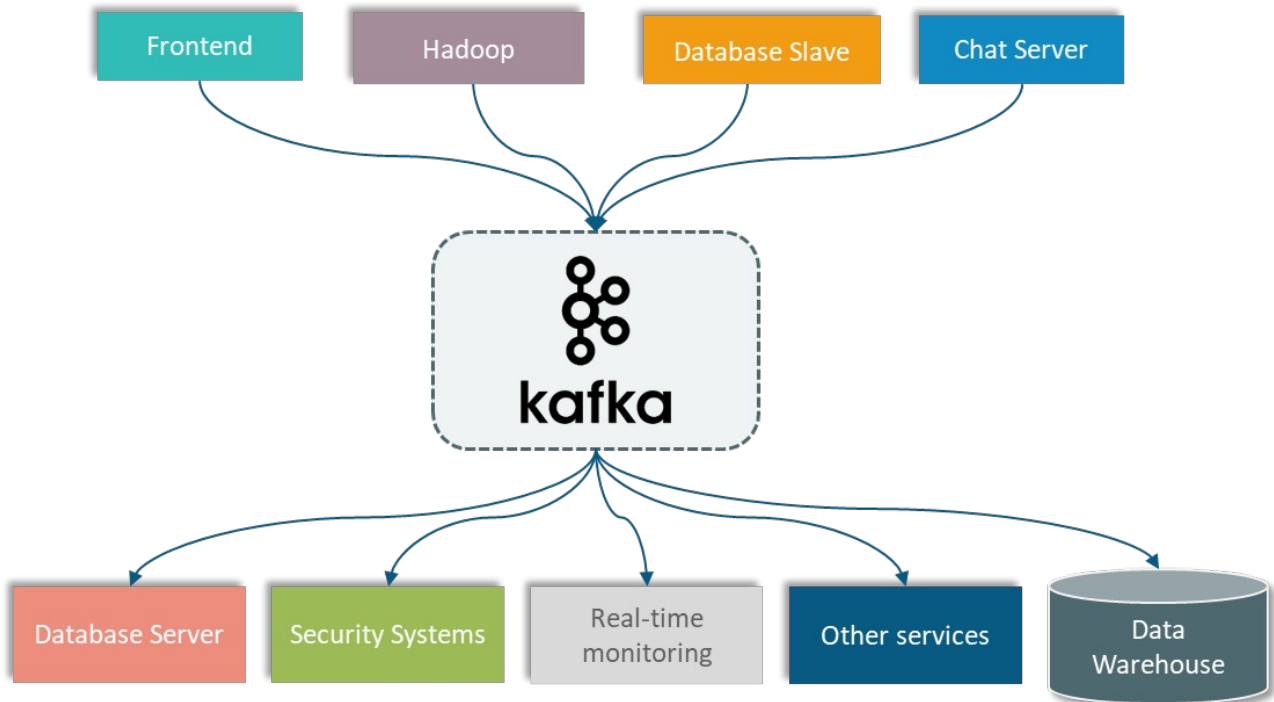
Principe B6 : Possibilité grâce à des fonctionnalités déclenchées par les événements

Déclaration :

« Tous les composants techniques doivent être conçus pour publier en continu les événements métiers, dont l'apparition déclenche d'autres fonctions métiers. »

Mise en application :

La fonctionnalité d'extension grâce à des événements n'a pas été incluse dans la preuve de concept. <TODO : Voir pour mettre en place un bus ?> En effet, la PoC ne prévoit pas de gestion événementielle. Dans le cas de micro-services, l'utilisation d'un système d'évènement basé sur des messages, comme pourrait le faire Apache Kafka, permettrait cette gestion extensions. De plus, l'intégration d'Apache Kafka dans le code est possible sans gros changement dans le développement.



Méthodologie architecturale et principes de processus

Principe C1 : Personnalisation de l'ADM TOGAF 9.2

Déclaration

« L'architecture métier sera façonnée par la personnalisation et l'amélioration continue d'un cadre d'architecture adapté à partir de l'ADM de TOGAF 9.2. »

Mise en application :

Principe C2 : Référentiel d'architecture centralisé et organisé comme source de référence

Déclaration

« Toutes les informations pertinentes sur le plan architectural devraient être disponibles dans un répertoire d'architecture central géré en permanence par la fonction d'architecture métier, qui en sera responsable. »

Mise en application :

Un dépôt d'architecture est présent pour tous les membres du projet à l'URL suivante :

https://github.com/etcomment/OCR_P11_repo_architectural

Du fait de l'utilisation d'un repository Git, celui-ci peut être cloné pour permettre des modifications à l'avenir.

Principe C3 : Normes ouvertes pour garantir des normes élevées

Déclaration

« L'application de normes ouvertes et de meilleures pratiques convenues peut soutenir l'organisation en lui apportant les connaissances et l'expertise du secteur. »

Mise en application :

Le développement d'une application sous forme de micros-services, et le choix fait d'une architecture modulaire permet de gérer une ouverture au niveau du code et permet l'application de normes élevées si besoin.

Principe C4 : Favoriser une culture de “learning” avec des preuves de concept, des prototypes et des Spike

Déclaration :

« L'entreprise encourage les implémentations centrées sur l'apprentissage qui réduisent les risques, valident les hypothèses et investissent dans l'apprentissage nécessaire pour faire évoluer la plateforme de manière responsable. »

Mise en application :