

# Principes d'Architecture

LABOUCHE Bastien

29 mars 2023



# Sommaire

<b>1 Informations sur le document</b>	<b>3</b>
<b>2 introduction</b>	<b>4</b>
<b>3 Principes "Business"</b>	<b>5</b>
<b>4 Principes "Donnée"</b>	<b>5</b>
<b>5 Principes "Application"</b>	<b>5</b>
5.1 Principe B1 : Continuité des activités des systèmes critiques pour les patients . . . . .	5
5.1.1 Déclaration . . . . .	5
5.1.2 Raisonnement . . . . .	5
5.1.3 Implications . . . . .	5
5.2 Principe B2 : Clarté grâce à une séparation fine des préoccupations . . . . .	5
5.2.1 Déclaration . . . . .	5
5.2.2 Raisonnement . . . . .	5
5.2.3 Implications . . . . .	6
5.3 Principe B3 : Intégration et livraison continues . . . . .	6
5.3.1 Déclaration . . . . .	6
5.3.2 Raisonnement . . . . .	6
5.3.3 Implications . . . . .	6
5.4 Principe B4 : Tests automatisés précoces, complets et appropriés . . . . .	6
5.4.1 Déclaration . . . . .	6
5.4.2 Raisonnement . . . . .	6
5.4.3 Implications . . . . .	7
5.5 Principe C4 : Favoriser une culture de "learning" avec des preuves de concept, des proto- types et des Spike . . . . .	7
5.5.1 Déclaration . . . . .	7
5.5.2 Raisonnement . . . . .	7
5.5.3 Implications . . . . .	7
<b>6 Principes "Technologie"</b>	<b>8</b>
<b>7 Signatures</b>	<b>9</b>



# 1 Informations sur le document

<b>Nom du projet :</b>	MedHead - Preuve de concept Système d'intervention d'urgence		
<b>Préparé par :</b>	Bastien Labouche	<b>Version du document :</b>	1.0
<b>Titre :</b>	Principes d'Architecture	<b>Date de publication :</b>	29 mars 2023
<b>Relu par :</b>	N/A	<b>Date de relecture :</b>	N/A

## Liste de distribution

<b>De :</b>	<b>Date :</b>	<b>Téléphone/Fax/Email</b>
Bastien		

<b>Pour :</b>	<b>Action</b>	<b>Date :</b>	<b>Téléphone/Fax/Email</b>
Kara Trace	Relecture		
Anika Hansen	Relecture		
Ashley Ketchum	Relecture		
Chris Pike	Relecture		

## Historique des versions

<b>Version</b>	<b>Date</b>	<b>Révisé par</b>	<b>Description</b>	<b>Nom du fichier</b>
1.0	29 mars 2023		Première version	



## 2 introduction

Ce document est le catalogue des exigences pour la preuve de concept du système d'intervention d'urgence en temps réel. Ce document liste et précise les exigences, hypothèses et contraintes spécifiques à la preuve de concept du système d'intervention d'urgence en temps réel.

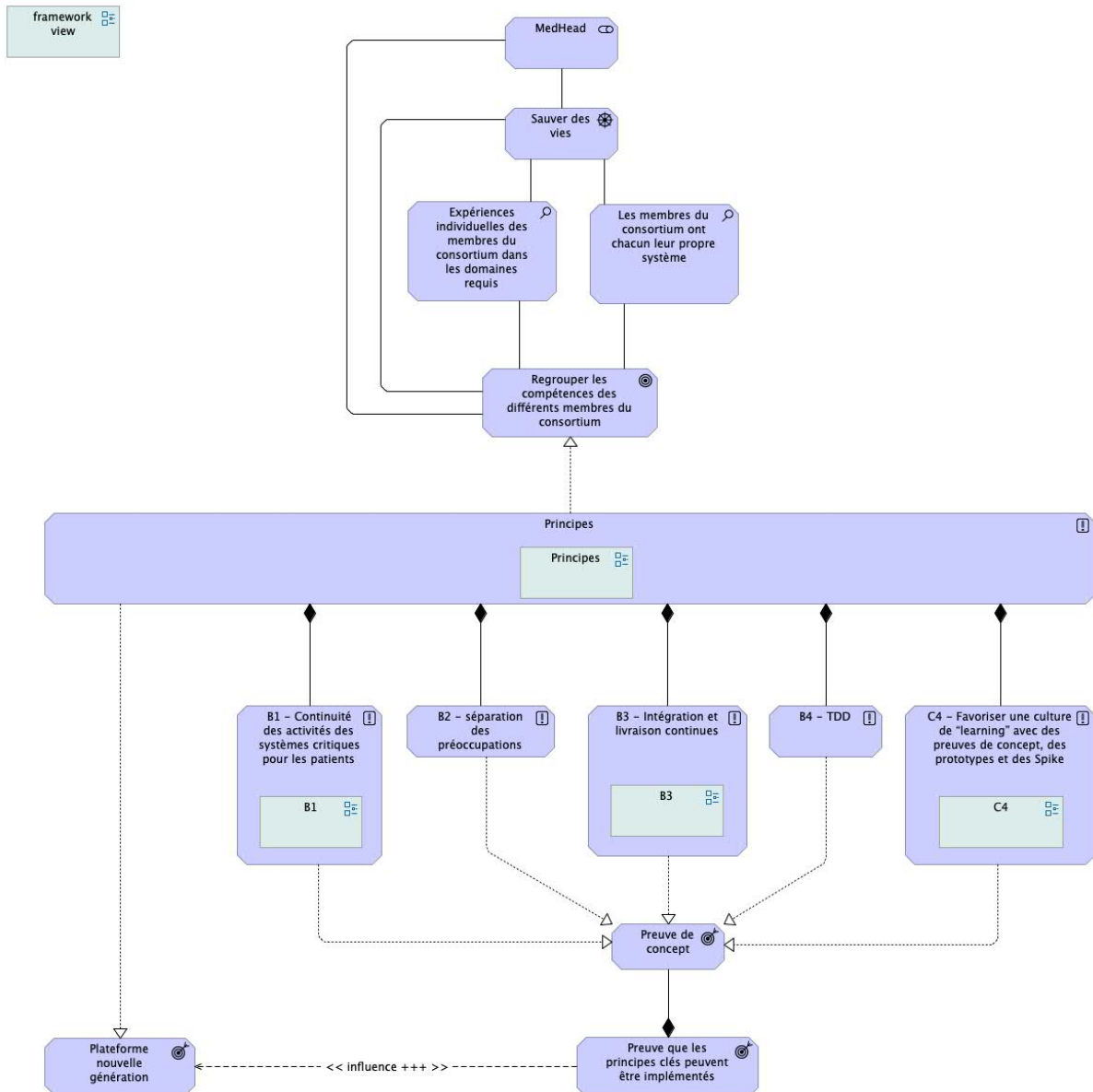


Figure 1: Alignement Stratégique

La figure ci-dessus, l'alignement stratégique, montre les principes d'architecture à l'origine de la preuve de concept.



### 3 Principes "Business"

### 4 Principes "Donnée"

### 5 Principes "Application"

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

##### 5.1.1 Déclaration

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

##### 5.1.2 Raisonnement

Étant donné que les soins aux patients sont considérés comme une priorité, tous les systèmes critiques doivent être construits conformément aux principes de tolérance aux pannes, de telle sorte que la priorité soit accordée à la fiabilité de ces systèmes tout au long de leur conception, de leur déploiement, de leur développement et de leur utilisation. Les partenaires médicaux, les fonctions métiers et techniques de l'entreprise doivent être en mesure de remplir leurs tâches indépendamment des événements externes. Les pannes matérielles, les attaques ciblées, les catastrophes naturelles et la corruption des données ne doivent pas perturber ou à arrêter les activités de l'entreprise

##### 5.1.3 Implications

La dépendance vis-à-vis des applications système partagées exige que les risques d'interruption des activités soient établis à l'avance et traités lorsqu'ils se présentent. La gestion comprend, sans s'y limiter :

- Principes SRE (Site Reliability Engineering) qui surveillent et mesurent en continu les SLI cibles (Service Level Indicators).
- Examens périodiques de la santé et des risques du système.
- Tests incrémentiels de performances, de vulnérabilité et d'exposition pour chaque incrément de la plateforme technique.
- Services critiques conçus pour assurer la continuité des fonctions de l'entreprise grâce à des capacités redondantes ou alternatives.
- La récupérabilité, la redondance et la maintenabilité doivent être prises en compte au moment de la conception.
- Les demandes doivent être évaluées selon leur criticité et leur impact sur la mission de l'entreprise, laquelle est d'assurer les soins aux patients.
- Des plans de reprise doivent exister pour tous les systèmes critiques.

#### 5.2 Principe B2 : Clarté grâce à une séparation fine des préoccupations

##### 5.2.1 Déclaration

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

##### 5.2.2 Raisonnement

Par entropie naturelle, les architectures complexes ont tendance à évoluer au fil du temps vers des réseaux régis par des dépendances complexes et difficiles à définir, et des responsabilités mal placées. Les composants d'une telle architecture sont souvent étroitement et fortement couplés.



Cela peut, au fil du temps, entraîner une perte des fonctions de l'architecture qui limite l'agilité d'une plateforme à répondre à l'évolution des besoins de l'entreprise ou des patients. Il faut connaître les limites du système. Il faut rendre le système transparents, c'est à dire :

- tout est bien découpé et on sait ce qu'il fait exactement
- on connaît les dépendances entre chaque fonction

### 5.2.3 Implications

Les décisions architecturales doivent suivre les principes et les meilleures pratiques de la conception pilotée par le domaine et des architectures de microservices. Cela implique un partenariat actif entre les équipes techniques et métier pour fournir des capacités à l'entreprise en utilisant un modèle partagé et un langage qui reflète le domaine des soins aux patients. Les dépendances étroites entre les capacités techniques doivent être identifiées et, dans la mesure du possible, doivent apporter une réponse aux situations problématiques traitées dans le contexte métier et dans le monde réel. Les solutions techniques doivent toutes être justifiées et modélisées en fonction de leur contribution globale aux scénarios de soins aux patients.

## 5.3 Principe B3 : Intégration et livraison continues

### 5.3.1 Déclaration

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

### 5.3.2 Raisonnement

L'intégration continue de petites fonctions et pipelines jusqu'à la production réduit les risques et permet d'avoir un retour précoce au sein des grandes équipes en cas de problèmes d'intégration. Une cadence de livraison rapide et régulière encourage également les équipes à réduire les risques en proposant des tests plus approfondis et de meilleurs résultats.

### 5.3.3 Implications

Les pipelines CI/CD doivent être facilement (ou automatiquement) déclenchés par des événements appropriés dépendant de l'état du code poussé sur le répertoire. Pour faciliter cela, les points suivants sont également à considérer :

- Les fonctionnalités doivent être clairement traçables dans le contrôle de version en utilisant des techniques d'étiquetage appropriées.
- Les exécutions CI/CD doivent être liées à une livraison de fonctionnalité donnée.
- Les exécutions CI/CD génèrent des journaux ou des sorties clairs qui peuvent être analysés pour isoler les builds en échec ou les erreurs dans les étapes de build, de test et de livraison.

## 5.4 Principe B4 : Tests automatisés précoces, complets et appropriés

### 5.4.1 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.

### 5.4.2 Raisonnement

Les bogues logiciels sont inévitables et peuvent être causés par des erreurs de code ou d'analyse. Des tests précoces garantissent que le logiciel est construit selon les spécifications et que chaque spécification est validée avant d'investir dans de mauvaises solutions.



### 5.4.3 Implications

Ce principe encourage l'utilisation de techniques de développement dirigé par des tests (TDD pour Test-Driven Development en anglais). Afin de valider rapidement les exigences, il est recommandé d'utiliser le langage du domaine métier lors des tests.

Les premières exigences devraient être rédigées sous une forme qui facilite les tests.

Implications : Les équipes devraient suivre la pyramide des tests et mettre en œuvre un niveau de test approprié pour chacune des catégories de tests suivantes

- Unitaire
- Intégration
- E2E

Lorsque les services sont interdépendants, il est également conseillé d'envisager des tests centrés sur le consommateur.

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

### 5.5.1 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.

### 5.5.2 Raisonnement

Le Consortium encourage collectivement l'utilisation de validations de principes, de Spike et de prototypes, ainsi que d'autres moyens d'enquête pour atteindre un état d'échec sans danger dans les zones où les informations disponibles sont insuffisantes pour comprendre le risque lié à la prise de décisions de conception ou de mise en œuvre spécifiques au niveau de la production. Le coût de l'investissement dans les efforts d'apprentissage pour réduire les risques est encouragé dans toute l'entreprise afin de protéger les intérêts des patients, des partenaires et de l'entreprise elle-même.

### 5.5.3 Implications

Les partenaires du Consortium conviennent collectivement de stimuler une culture de prise de décision fondée sur des preuves et centrée sur l'apprentissage. Ce faisant, les exceptions et considérations suivantes devraient s'appliquer :

1. Fournir une hypothèse pour chaque apprentissage
  - Toutes les implémentations liées à l'apprentissage doivent être accompagnées d'une hypothèse définissant l'apprentissage souhaité et permettant de mesurer si ce résultat d'apprentissage a été atteint.
2. Isoler les preuve de concept des données et des systèmes de production
  - Des mesures doivent être prises pour atténuer ou éliminer le risque d'impact sur les patients lorsqu'il existe un risque de nuire au patient ou à l'entreprise. Par exemple, l'apprentissage peut être mené de manière isolée dans un environnement artificiel afin d'éviter l'impact sur les systèmes de production. Utiliser des données factices ou anonymisées
  - Les données des patients utilisées pour les activités d'apprentissage à haut risque doivent être protégées, afin d'éviter un impact sur la sécurité des données ou les soins aux patients. Les PoC devrait utiliser des données anonymisées ou factices lorsque cela est possible.
3. Assouplir la conformité, mais tenir compte des conséquences



- Les normes de gouvernance et les niveaux de conformité peuvent être assouplis lorsque des mesures sont prises pour protéger les systèmes de production et les données des patients. Les PoC isolées des données réelles des patients et des systèmes de production ne sont pas régies par des normes externes ou une quelconque gouvernance d'entreprise en matière de séparabilité.
  - Lorsque les normes et la gouvernance ne sont pas pleinement respectées, les responsables de la mise en œuvre et les concepteurs devraient réfléchir à la manière dont ces prototypes ou ces mises en œuvre centrés sur l'apprentissage peuvent fournir des leçons dans les mises en œuvre finales en production.
  - Il est fortement déconseillé de produire directement des prototypes. Il convient plutôt de veiller à ce que les conceptions tiennent compte des effets secondaires de la production qui peuvent invalider tout apprentissage. Par exemple, l'omission de problèmes de sécurité ou la mauvaise estimation du volume de données attendu peut entraîner des problèmes de performances qui invalident les apprentissages tirés d'un tel prototype non évolutif.
  - Les tests de performance des prototypes et des implémentations centrées sur l'apprentissage devraient valider les algorithmes clés faisant partie de cette échelle d'apprentissage.
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.
- 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
5. Plans de test comme outils de communication des exigences
- Les livrables avec des plans de test autodocumentés sont préférables aux plans de test documentés en externe.
  - La preuve de concept doit comporter des plans de test décrivant comment le produit doit se comporter.
  - Les plans de test doivent utiliser des scénarios BDD (behaviour-driven development - voir C3) pour décrire les critères d'acceptation métier qui sont dans la portée.
  - Les plans de test doivent utiliser le langage commun de l'entreprise et être compréhensibles par les partenaires techniques et non techniques.
6. Tester les rapports d'exécution pour documenter le comportement pris en charge Pour prendre en charge la visibilité des comportements attendus, l'apprentissage continu et la transparence concernant l'état du logiciel :
- Les PoC devraient avoir des pipelines CI qui exécutent des tests et produisent des rapports d'exécution des tests
  - Les environnements CI doivent permettre aux propriétaires des logiciels d'inspecter les exécutions passées et les dégradations de la build qui peuvent affecter les l'hypothèse, en ligne avec le principe B3.

## 6 Principes "Technologie"





## 7 Signatures

LABOUCHE Bastien