

Rapport

Membres de l'équipe :

- AMAADOR Omar
 - ASCARI Yannick
 - EL AJI Walid
 - TLILI Abderrahmen
 - NAJAR Sarra
-

Notre objectif métier

- **Maintien à domicile** en sécurité, avec suivi continu **de quoi?**
- **Moins de visites systématiques** de l'infirmier·ère ; interventions **à la demande** (sur alerte ou tendance).
- **Alertes automatiques + rapports clairs** pour patient, famille et soignants.

Permettre le maintien à domicile en toute sécurité grâce à une surveillance continue avec alertes et rapports, afin de réduire les visites systématiques à l'hôpital et intervenir seulement quand c'est nécessaire.

Pack unique (tout inclus)

Élément	Détails
Appareils	Montre (fréquence cardiaque, SpO ₂ , pas), Tensiomètre , Glucomètre , Balance
Applications	Appli mobile patient (Android/iOS), portail web pour soignants et famille
Plateforme	Alertes en temps réel, stockage longue durée, tableaux de bord, génération de rapports, gestion des RDV
Services	Onboarding, support matériel, (option) surveillance 24/7, export PDF/CSV

Utilisation des appareils (cadence)

Appareil	Données	Fréquence d'usage	Remarques
Montre	FC, SpO ₂ , pas	FC en continu ; SpO ₂ ponctuel + la nuit ; pas par minute	Peut fournir des indices d'effort et de sommeil

Tensiomètre	SBP/DBP + FC	1 à 3x / jour, de préférence assis, 5 min au repos	Série de 2–3 mesures, moyenne calculée
Glucomètre	Glycémie	Avant et après repas, + si symptômes	Tag “pré”/“post” repas dans l’appli
Balance	Poids	1x / jour, le matin après toilette	Garder la mesure la plus stable du jour

Données produites

Type	Exemple	Unité	À quoi ça sert
FC (montre)	78	bpm	Repos/effort, tachycardie
SpO ₂ (montre)	96	%	Hypoxie, apnées suspectes
Pas / activité	5 000	pas/jour	Sédentarité, récupération
Tension artérielle	132/85	mmHg	Hypertension, urgence
Glycémie	145	mg/dL	Hypo/hyperglycémie
Poids	78.4	kg	Rétention hydrique, suivi nutrition

Pipeline d’ingestion de données

1) Capture sur l’appareil (montre / tensiomètre / glucomètre / balance)

Transformations

- Horodatage local de la mesure.
- Emballage des métadonnées : batterie, firmware, qualité de mesure (ex. brassard bien serré).

Contrôles

- Plages physiques plausibles (ex. SpO₂ 50–100 %, FC 30–220 bpm).
- Signaux de qualité (mouvement, mauvaise pose, bandelette invalide).
- Reprise automatique (ex. BP répète une mesure si artefact détecté).

2) Téléphone du patient (appli “edge”)

Transformations

- **Normalisation** des unités (kg, mmHg, mg/dL, bpm).

- **Synchronisation du temps** → conversion en UTC.
- **Regroupement** (petits lots de 5–10 s) pour économiser le réseau.
- **Pseudonymisation** : patient → identifiant technique.
- **Compression + chiffrement** avant envoi.

Contrôles

- **Validation de schéma** (champs obligatoires présents).
- **Déduplication** (numéro de séquence / hash).
- **Anti-bruit** : rejet des valeurs hors plage configurée.
- **Tampon hors-ligne** chiffré (≥ 48 h) + reprise automatique.
- Avertissements locaux : batterie faible, Bluetooth perdu, mesure ratée.

3) Transport & courtier de messages (ex. MQTT/HTTPS)

Transformations

- **Routage** par patient/appareil (topics).
- Ajout d'un **horodatage serveur** à la réception.

Contrôles

- Qualité de service (QoS 1/2), ré-essais avec backoff.
- **Authentification** (certificats / OAuth) & **autorisations** par rôles.
- **Limitation de débit** par appareil, journaux d'audit (connexion, erreurs).

4) Traitement en flux (Stream Processing)

Transformations (principales)

- **Rééchantillonnage / agrégation** :
 - **FC** : 1 s → **paquets 1 min** (moyenne, médiane, min, max, écart-type) + **fenêtre glissante 5 min** (moyenne, z-score, lissage EWMA).
 - **SpO₂** : valeurs ponctuelles + **% du temps < 90 %** (fenêtres 5 min et 1 h).
 - **Tension** : moyenne d'une **série de 2–3 mesures**, calcul **MAP** = DBP + (SBP–DBP)/3.
 - **Glycémie** : tag **pré/post repas** ; **Time-In-Range** (70–180 mg/dL) par jour.
 - **Poids** : lissage 3 jours, **variation hebdomadaire (%)**, **IMC** (avec taille enregistrée).
- **Enrichissement** : fuseau horaire, préférences patient, seuils personnalisés.
- **Routage d'événements** : création d'événements **ALERTE/URGENCE** vers le service d'alerting et le **Live Stream**.

Contrôles

- Marquage qualité : `out_of_range`, `mouvement`, `appareil_offline`, `donnée_manquante`.
- Gestion des **retards** (late events) avec watermark ; **idempotence** des écritures.
- Surveillance **SLA** : latence capteur→pipeline, pertes, erreurs.

5) Stockage (time serieeeeeesss)

Transformations

- Écriture double :
 - Brut (rétention courte, ex. 30 jours).
 - Agrégés (1 min / 5 min / 1 h) pour analyses et rapports (réception longue, ex. 5 ans).
- Indexation par patient, type, période ; partitionnement par temps.

Contrôles

- Chiffrement au repos, sauvegardes régulières, vérifications d'intégrité.
- Politiques de **rétention** et **anonymisation** pour usages secondaires.

6) Service de diffusion en direct (Live Streaming)

Transformations

- Publication des **dernières fenêtres** (ex. FC 1 min, SpO₂ récente) et des **événements d'alerte** via WebSocket/SSE.
- **Throttling** côté serveur pour éviter la surcharge (ex. max 1 mise à jour/s par client).

Contrôles

- Authentification/autorisation par rôle (patient, aidant, soignant).
- Reconnexion automatique, limite du nombre de clients.

7) Alertes & escalade (sécurité du patient)

Niveau	Exemple de déclencheur (par défaut)	Action
Info	Tendance FC au-dessus de l'habitude sur 1 h	Notification douce + conseils
Alerte	SpO ₂ < 90 % cumulé ≥ 2 min / 10 min ; SBP ≥ 160 répété ; glycémie > 250 ou < 70	Notification push/SMS au patient + tableau de bord soignant
Urgence	SBP ≥ 180 ou DBP ≥ 120 ; SpO ₂ < 85 % ; glycémie > 300 ou < 54 ; +2 kg en 3 jours (IC)	Appel/alerte prioritaire, contact aidant/plateau, suggestion de RDV

8) Rapports & consommation finale

- **Hebdomadaire (patient/famille)** : résumé simple (tendances, alertes, adhérence).
- **Mensuel (clinicien)** : statistiques détaillées, comparaisons, événements, RDV.
- **Rapport post-événement** : fenêtre ±72 h autour d'une alerte.

Agrégations & rééchantillonnage (à différents niveaux)

Niveau	Ce qui arrive	Rééchantillonnage (sampling)	Agrégations calculées	Pourquoi
Appareil (montre, BP, glucomètre, balance)	Mesures brutes (HR 1 s, SpO ₂ spot, BP séries, glucose ponctuel, poids quotidien)	HR natif (1 Hz) ; le reste "tel quel"	Moyenne très courte côté montre si dispo (anti-bruit)	Limiter le bruit avant envoi
Téléphone (edge)	Paquets de quelques secondes	Regroupement par lot 5–10 s (pas de perte)	Aucune (juste normalisation unités/horodatage)	Économiser batterie/réseau
Pipeline temps réel	Flux normalisé	HR: 1 s → 1 min ; fenêtre glissante 5 min (pas 1 min) • SpO₂: fenêtres 5 min & 1 h • BP: par série • Glucose: par repas/jour • Poids: journalier	HR: mean/median/min/max/std, z-score , EWMA • SpO ₂ : % temps < 90 % • BP: moyenne de série , MAP • Glucose: Time-In-Range (70–180) • Poids: moyenne 3 j , Δ7 j % , IMC	Produire des indicateurs stables pour alertes
Stockage	Bruts + agrégés	Niveaux d'agrégats: 1 min / 5 min / 1 h / jour	Rollups (sum/mean/min/max/count/std, percentiles p10/p90)	Requêtes rapides et rapports
Rapports (hebdo/mensuel)	Séries consolidées	Jour → Semaine → Mois	Moyennes/medians hebdo, boxplots (BP/glucose), tendances, adhérence (%)	Lecture simple & décisions cliniques

analyse

Composant	Comment on le voit	Action automatique	Action humaine	Continuité pour l'utilisateur
Montre / Tensiomètre / Glucomètre / Balance	Plus de mesure / batterie faible	Notification “reconnecter / recharger / refaire mesure”	Appel/soutien si >24 h	Données gardées sur le téléphone, envoi dès retour
Téléphone (appli)	Pas de “heartbeat” / app stoppée	Relance en arrière-plan, rappel permissions	Aide au réglage (économie d'énergie, Bluetooth)	Tampon local des mesures
Réseau patient	Envos en échec	Retries + bascule Wi-Fi/4G si possible	—	Tampon → envoi plus tard
Broker (MQTT/HTTPS)	Healthcheck KO	Bascule automatique vers nœud de secours	Intervention SRE	Messages mis en file, puis rejoués
Pipeline streaming	Retard (lag) / job down	Redémarrage auto, envoi en file d'attente	Lecture file d'erreurs & correction	Alertes possibles avec léger retard
Base de données	Erreurs lecture/écriture	Bascule sur réplica, limitation du débit	Ajout de capacité / réparation	UI montre “données partielles”
Alerting	SMS/push en échec	Changement de canal (push↔SMS)	Changer de fournisseur, ajuster quotas	Les urgences passent par le canal alternatif
Portail / App pro	Latence / indispo	Auto-scale, désactiver fonctions lourdes	Rollback si besoin	Mode lecture seule possible

9 Sécurité :

Toutes les communications passent par une **API Gateway** ou un **broker MQTT/HTTPS** sécurisé.

Utilisation obligatoire de **TLS 1.3** pour toutes les transmissions.

Authentification forte via certificats ou jetons.

Rate limiting

Audit logs générés pour chaque connexion et chaque opération critique.

Un **pare-feu applicatif (WAF)** bloque les attaques classiques (injection, XSS, CSRF)

NOTES PROF FROM LAST SESSION

On va s'intéresser à **DDD**.

- Notre **core domain**, c'est la **data collection**.
- Les autres **domaines** sont : **reporting, rendez-vous, urgence**.
- Les autres **sous-domaines** sont : **notifications, dates, suivis, reporting, notification et détection**.
- On doit définir exactement :
 - C'est quoi notre **métier**.
 - C'est quoi le **monitoring** qu'on va offrir :
 - Surveillance médicale continue et intelligente
 - Détection rapide des anomalies
 - Alertes en cas d'urgence
 - Rapports périodiques pour le suivi des patients
 - Qui sont nos **patients**: personnes âgées / médecins / infirmiers
 - On va **monitored quoi, qui, et quand**.
 - Quoi : constantes vitales (définies au dessus(**Données produites**))
 - Qui : les personnes équipées de capteurs connectés
 - Quand : périodiquement(**! interval a définir**)
- Ici, notre **domaine** doit être défini : c'est quoi le **main domain**.
- Un **domaine** est égal à des **intervenants**.

-
- Il faut **définir un domaine**.
 - Après la définition d'un domaine, on commence la **vue service / microservice**.

- Avec ça, il faut faire le **lien entre domaine et service**.
-

- Il faut absolument **définir le pipeline** : Le **pipeline**, c'est le **cycle de vie de nos données**. 
-

- Le **data integration pipeline**, c'est **très important** et c'est **noté**.
- C'est une **partie essentielle du projet**.
- Il faut définir :
 - Les **layers** dans ce data integration pipeline.
 - La **compression** qu'on utilise.
 - Les **filters**.
 - Les **annotations**.
 - Les **entités** qu'on utilise.

-
- Où on **stocke notre data** et comment on l'utilise.
 - Comment on utilise la **data stockée** :
 - Est-ce que ce sont des **streams** ?
 - Est-ce que ce sont des **batchs** ?

-
- On a besoin d'une **couleur par domaine**.
 - On doit absolument **définir le Data Integration Pipeline**.

- On est libre d'utiliser **n'importe quel diagramme — tous les diagrammes** qu'on veut.
-

- Il faut vraiment **expliquer la business value** de chaque partie.
-

- À la fin, on doit **présenter une architecture pipeline**.
- Cette architecture doit **faire apparaître chaque domaine** sur ce pipeline.

Notes in general (le premier prof , pas le prof de DDD)

- Des **remarques données par le prof** :
 - Il faut **absolument clarifier les connexions**.
 - Par exemple : comment **insérer les datas dans la TimeDB** ?
 - Est-ce qu'on utilise des **agrégations**, des **fréquences** ?
 - Est-ce qu'on utilise **REST**, etc.
- Il faut **absolument définir ce qu'est un smartphone** :
 - Est-ce qu'il fait des **traitements** ?
 - Est-ce qu'il fait de **l'agrégation** ?
 - Est-ce qu'il y a des **filtres**, etc.
- Il faut **expliquer ces cinq éléments** dans le **rapport final**, en utilisant des **diagrammes**, des **paragraphes**, etc.

1. Sécurité

- Comment on fait le **cryptage** et l'**encryption** des données.

2. Authentification

- Fournir un **diagramme d'authentification** entre les services.

3. Scaling

- Expliquer **comment on peut scaler la solution**.

4. Maintenance

- Comment on gère le **downtime** et la **maintenance planifiée**.

5. Multi-tenant

- Comment la solution peut assurer une **architecture multi-tenant**.
- Est-ce qu'on utilise une **architecture multi-tenant** ou un **déploiement par client**.

1. Users stories

Le patient

En tant que *patient*, je souhaite:

Pouvoir remplir le formulaire, qui a été créé par le docteur, afin de renseigner mes ressentis, sentiments, état de santé au docteur.

Consulter un rapport (hebdomadaire, journalier) qui ne soit pas trop spécifique et technique, afin de connaître des informations de santé (indications de traitements, santé, etc.)

Avoir un retour sur mon état de santé (bilan) grâce à mon docteur, afin de connaître mon état de santé et savoir quelles actions je dois faire pour me soigner, comme les traitements, repos, hydratation etc.

Déclarer un état d'urgence, afin d'avoir une intervention rapide du médecin.

Être informé des rendez-vous ou interventions prévues, afin de voir l'infirmière et le médecin pour discuter de son état de santé et prendre connaissance des mesures à prendre pour se soigner.

L'infirmier(ère)

En tant qu'infirmier, je souhaite **consulter la liste des patients**, afin de **gérer efficacement leur suivi quotidien**.

En tant qu'infirmier, je souhaite **accéder à l'historique des données cliniques et des capteurs d'un patient**.

En tant qu'infirmier, je souhaite **observer et dialoguer avec le patient**, puis **saisir des observations dans un rapport**

En tant qu'infirmier, je souhaite **être informé par notification lorsqu'une anomalie est détectée** dans les données de santé

En tant qu'infirmier, je souhaite **consulter la liste des rendez-vous planifiés avec mes patients et pouvoir en planifier de nouveaux**

Le docteur

En tant que docteur, je souhaite consulter ma liste de patients, afin de gérer et prioriser mes interventions médicales.

En tant que docteur, je souhaite analyser un bilan de santé pour chaque patient (formulaire du patient, observations de l'infirmier, données capteurs), afin de décider des mesures médicales adaptées (ordonnance, examens, séances).

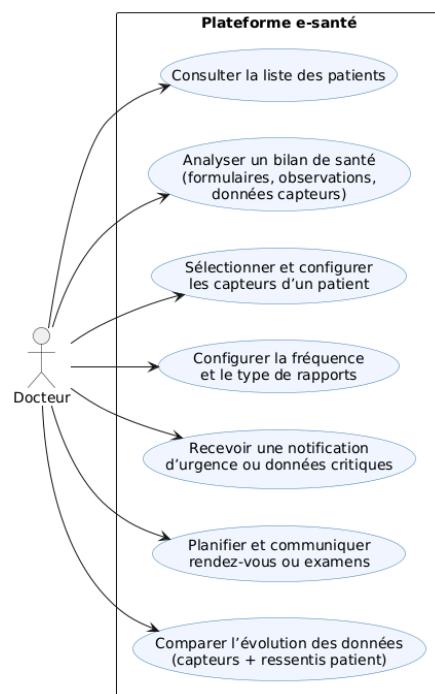
En tant que docteur, je souhaite sélectionner et associer les capteurs pertinents pour chaque patient (selon pathologie et disponibilité), afin de disposer de données objectives pour le suivi.

En tant que docteur, je souhaite configurer la fréquence et le type de rapports générés pour le patient et ses proches, afin d'adapter le niveau de détail au destinataire. (génération automatique)

En tant que docteur, je souhaite recevoir une notification immédiate en cas d'urgence déclarée ou de données critiques détectées, afin de réagir rapidement.

En tant que docteur, je souhaite planifier et communiquer des rendez-vous ou examens via la plateforme, afin de coordonner efficacement le suivi avec le patient et l'infirmier.

En tant que docteur, je souhaite comparer l'évolution des données capteurs et des ressentis du patient dans le temps, afin de détecter des tendances et ajuster le traitement.



Les proches

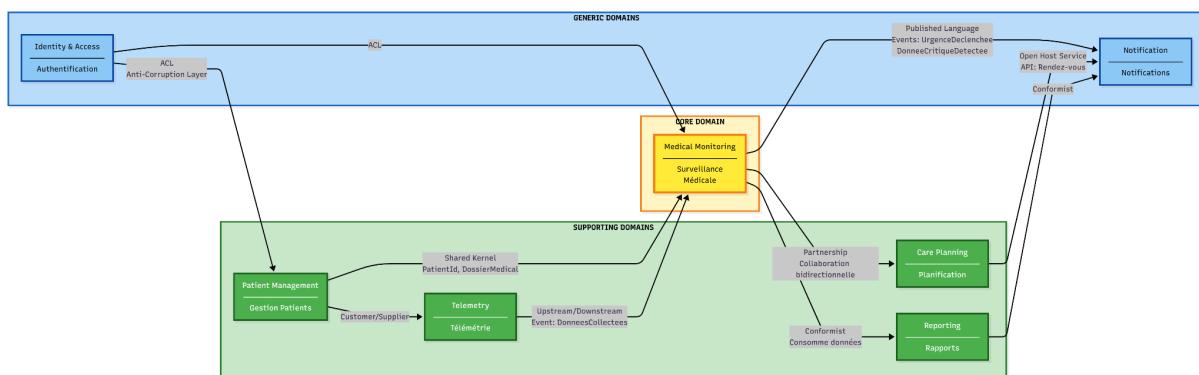
En tant que *proche*, je souhaite:

- recevoir une notification en cas d'urgence (sans trop de détails)
- Être informé des rendez-vous ou interventions prévues
- Recevoir un bilan périodique
- pouvoir noter mes observations sur le patient à travers un formulaire

Administrateur

- **Gérer les comptes et rôles** (patient, proche, infirmier, docteur, concepteur, admin), afin d'appliquer le moindre privilège.

- Identifier le smartphone ou la montre connectée d'un patient et pouvoir connaître la version de son logiciel, et en déclencher la notification de mise à jour.
 - Mettre à disposition l'accès et la gestion des informations des utilisateurs (médecins, infirmiers, patients) avec chiffrement des données de santé des patients, afin d'assurer une exploitation fiable et conforme.
- dique.



1. C'est quoi notre métier

Notre métier est d'assurer le **suivi en temps réel de l'état de santé des résidents d'une maison de retraite**, en combinant **capteurs connectés, analyse intelligente et alertes automatiques**.

L'objectif est de **faciliter le travail du personnel soignant**, de **prévenir les situations à risque** (malaises, chutes, fièvre, troubles respiratoires...) et de **garantir la sécurité et le bien-être des résidents**, tout en réduisant la charge de surveillance manuelle.

En Domain-Driven Design, notre **core domain** est la **collecte et le traitement des données de santé des résidents**.

2. C'est quoi le monitoring qu'on va offrir

Nous offrons un **monitoring médical continu et intelligent**, centré sur la détection précoce des anomalies.

Concrètement :

- **Suivi automatisé** des constantes vitales (fréquence cardiaque, SpO₂, température, activité, sommeil).

- **Analyse de tendances** pour repérer des dégradations progressives (ex. fatigue, inactivité prolongée).
- **Alertes instantanées** en cas d'anomalie (ex. chute, rythme cardiaque trop élevé, hypoxie).
- **Tableaux de bord médicaux** pour le personnel infirmier, avec un **classement par priorité** (normal, à surveiller, urgence).
- **Rapports périodiques** pour les médecins coordinateurs et la direction.

Ce monitoring soutient à la fois la **réactivité médicale** (urgence) et la **prévention à long terme** (tendances).

3. Qui sont nos patients

Nos patients sont les **résidents de la maison de retraite** :

- Principalement des **personnes âgées dépendantes** nécessitant une attention médicale régulière.
- Certains sont atteints de **pathologies chroniques** (hypertension, troubles cardiaques, diabète, Alzheimer...).
- Tous suivis par le **personnel infirmier** de l'établissement et, dans certains cas, par des **médecins externes**.

On distingue donc plusieurs **intervenants métier** :

- Le **résident** (porteur du capteur).
 - Le **personnel soignant** (surveille et intervient).
 - Le **médecin référent** (analyse les rapports).
 - Le **système de monitoring** (collecte, détecte, alerte).
-

4. On va monitorer quoi, qui, et quand

- **Quoi :**

Les données physiologiques et comportementales des résidents :

- Rythme cardiaque, SpO₂, température corporelle.
- Activité physique et sommeil (déplacements, chutes).
- Poids, tension artérielle, et autres mesures ponctuelles.

- **Qui :**

Tous les **résidents équipés de dispositifs connectés** (montre, capteurs BLE dans la chambre...).

Chaque capteur est lié à **un résident identifié**.

- **Quand :**

- **En continu**, pour les signaux vitaux essentiels (rythme, SpO₂).
- **Périodiquement**, pour les mesures planifiées (poids, tension, température). **Immédiatement**, lorsqu'un **événement critique** est détecté (chute, valeur hors seuil).
- **Sur demande**, lors de **contrôles infirmiers ou médicaux**.

En résumé : on monitorer **les résidents, 24h/24, dans leur environnement réel**, pour **prévenir les urgences et aider le personnel à prioriser les interventions**.

avec **tensiomètre connecté + smartwatch + glucomètre + balance connectée**,

Métrique	Description	Fréquence suggérée
Fréquence cardiaque	Battements par minute	Toutes les 10–15 secondes
Pression artérielle	Systolique / Diastolique	Toutes les 1–2 minutes
Glycémie (taux de sucre)	Taux de glucose dans le sang	Toutes les 5–10 minutes
Température corporelle	Température interne du corps	Toutes les 1 minute
Saturation en oxygène (SpO₂)	Pourcentage d'oxygène dans le sang	Toutes les 30–60 secondes
Fréquence respiratoire	Respirations par minute	Toutes les 1 minute
Taux de CO₂	Taux de dioxyde de carbone (sang ou expiré)	Toutes les 2–3 minutes
Poids	Poids corporel	Toutes les 12–24 heures
IMC (Indice de Masse Corporelle)	Calculé à partir du poids et de la taille	Toutes les 12–24 heures
Niveau de stress	Calculé par variabilité cardiaque ou capteurs	Toutes les 1 heure
Qualité du sommeil	Phases du sommeil, repos	Toutes les 1 heure ou au réveil
Niveau d'activité	Pas, mouvement, posture, etc.	Toutes les 5 minutes

ECG / EKG	Signal électrique du cœur (brut)	Toutes les quelques millisecondes
Niveau d'hydratation	Taux d'eau dans le corps	Toutes les 1–2 heures
Température de la peau	Température de surface	Toutes les 2–5 minutes
Température ambiante	Température de l'environnement	Toutes les 5–10 minutes
Signal oxymétrique brut	Forme d'onde du capteur d'oxygène	Toutes les 1 seconde

Hypothèse:

- On suppose que le système est destiné pour une maison de retraite
- On suppose que c'est l'admin qui crée les comptes et affecte les priviléges selon les rôles (médecin, infirmier, patient, proche,...)
- On suppose que les droits sont gérée d'une manière hiérarchique, cad l'admin peut gérer tous les comptes, le médecin gère les compte des infirmiers et les patients ...

Fonctionnalités systèmes :

- Générer les rapports automatiquement pour les patients - les proches

Risque	Probabilit é	Impact	Niveau de risque	Mesures d'atténuation
Défaillance des capteurs (panne, données erronées) (plutot pas de données)	Moyenne	Élevé	Élevé	Vérification régulière, alertes automatiques, redondance des capteurs
Perte ou vol de données sensibles (RGPD)	Faible	Très élevé	Élevé	Chiffrement, anonymisation, politique RGPD stricte, audits réguliers
Non-détection d'une urgence critique===(absence de réception d'alerte)	Faible	Très élevé	Élevé	Système d'alerte redondant (capteurs + bouton SOS), surveillance en continu
Indisponibilité de la plateforme (panne serveur, maintenance)	Moyenne	Élevé	Élevé	Infrastructure scalable, sauvegardes, CI/CD, supervision

Manque d'engagement du patient (oubli de remplissage, fatigue numérique)=	Élevée	Faible	Moyen	Notifications, rappels, interface utilisateur intuitive
Non-réception des alertes ou messages importants	Moyenne	Moyen	Moyen	Multicanal (SMS/email/app), système d'accusé de réception, relance automatique
Coupeure du courant	faible (géographique)	Élevée	Moyen	Ajouter des groupes électrogènes, détecter les pannes, et déployer plusieurs instances de l'application sur au moins 2 noeuds de Serveur (par ex: une instance de secours dans un DC à Antibes et un serveur prod à Cannes)
Coupeure de réseau (<i>internet</i>)	Moyenne	Élevée	Moyen	Mettre en place un Retry on Error, relancer la requête jusqu'à ce que ça marche, ajouter aussi un cache local qui permet de stocker les données critiques non-sauvées. Créer une file d'attente (Queue), pour ajouter les opérations critiques - non sauvées.
Faille de sécurité	Plutôt élevée	Critique	Élevée	Mettre en place un système de security checker avec la CI/CD, comme Sysdig . Permettant d'éviter de déployer quand il y a des failles dans le code.
Charge Max des serveurs (Pic de requêtes)	Moyen	Moyen	Élevée	On peut mettre en place des systèmes de sécurité, permettant de filtrer les requêtes (par les robots par ex.). Avoir un auto-scaling au niveau du déploiement, qui permet d'avoir plus de ressources quand il y a beaucoup de requêtes.
Obsolescence de certaines bibliothèques ou composants	Faible	Moyen	Moyen	Versioning ou mise à jour régulière (màj auto), Dépendre le moins de dépendances qui sont mal maintenus et pas très connus.

Priorités : scenarios fournisseurs , hypotheses , protocoles utilisés par les capteurs

Architecture technique

Maison de retraite (Edge)

Capteurs connectés ()

Besoins

- Mesure continue et automatique des données de santé (FC, SpO₂, glycémie, tension, poids).
- Détection rapide d'événements anormaux (chute, hypoglycémie, hypertension).
- Dispositif non invasif, adapté à des personnes âgées.
- Transmission fiable et sécurisée des données.

Contextetdedup (seq/hash)

- Les patients sont équipés d'une **smartwatch** et éventuellement d'autres capteurs médicaux connectés (tensiomètre, glucomètre, balance).
- Ces capteurs communiquent en **Bluetooth Low Energy (BLE)** avec un smartphone passerelle

Justification

- Avantages montre connectée :
 - Portée 24/7, quasi-transparente pour le patient.
 - Un seul appareil portable, pas de câblage ni de contraintes lourdes.
 - Multi-capteurs intégrés (FC, SpO₂, mouvement, stress, sommeil).
- Inconvénients montre connectée :
 - Autonomie limitée (recharge quotidienne).
 - Précision parfois inférieure à des capteurs spécialisés.

Technologies

- *** pour Android.
- *** pour iOS.
- Frameworks santé (Google Health Platform, Apple HealthKit).

Alternatives

- Capteurs spécialisés par métrique (ex. glucomètre médical, tensiomètre).

- Avantages : précision médicale, autonomie élevée.
- Inconvénients : multiplication des appareils, adoption difficile par des personnes âgées.

Micro-services applicatifs

Besoins

- Modularité (patient, infirmier, docteur, proches, admin).
- Évolutivité.
- Gestion des rôles et droits.

Contexte

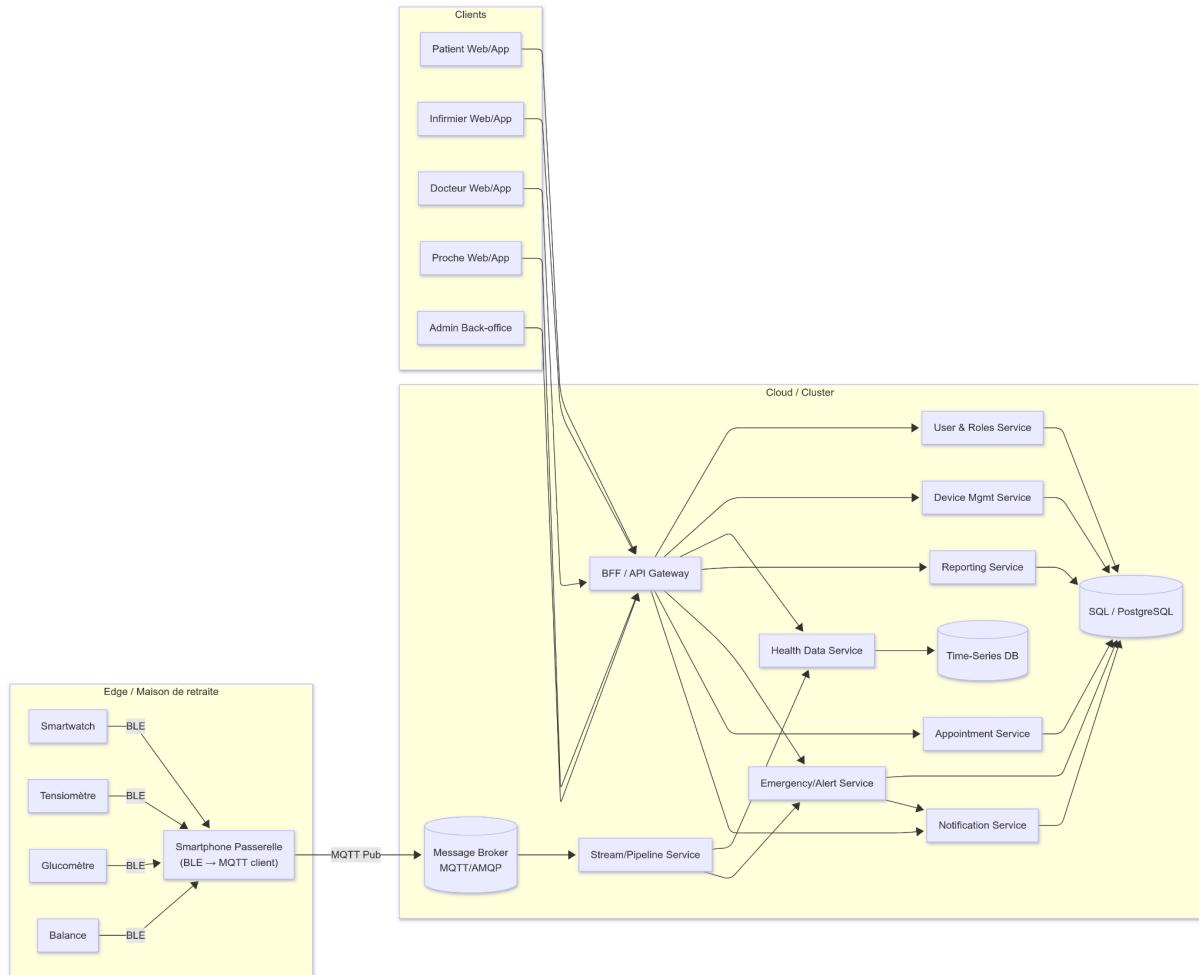
- Chaque fonctionnalité est isolée dans un **micro-service** :
 - Gestion des utilisateurs et rôles.
 - Gestion des données santé.
 - Génération de rapports.
 - Gestion des alertes/urgences.
 - Gestion des rendez-vous.
 - Notification multicanale.

Justification

- Avantages micro-services : évolutifs, résilients, déploiement indépendant.
- Inconvénients : complexité DevOps (CI/CD, monitoring).

Technologies

- **Java Spring Boot / Node.js / Go** (selon équipe).
- **PostgreSQL** pour les données structurées.
- **InfluxDB** pour données temporelles.
- ******* pour l'orchestration.

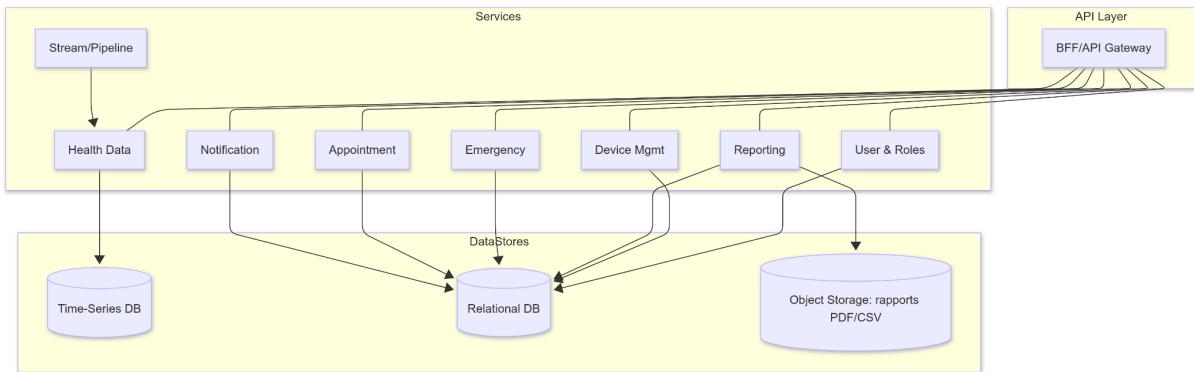


- un seul box
- webapp / front avant gateway
- c quoi un smartphone ?
- Type de liaison
-

Flowchart d'architecture système e-santé:

- **API Gateway** expose des endpoints adaptés à chaque rôle (façade unique).
- **Broker** central pour télémétrie et événements (découplage temporel/spatial/synchronisation).
- **Pipeline** (stream) applique nettoyage/normalisation/agrégation et publie des événements métiers (p. ex. *HealthStatusUpdated*, *EmergencyTriggered*).

Vue d'implémentation (composants et contrats)

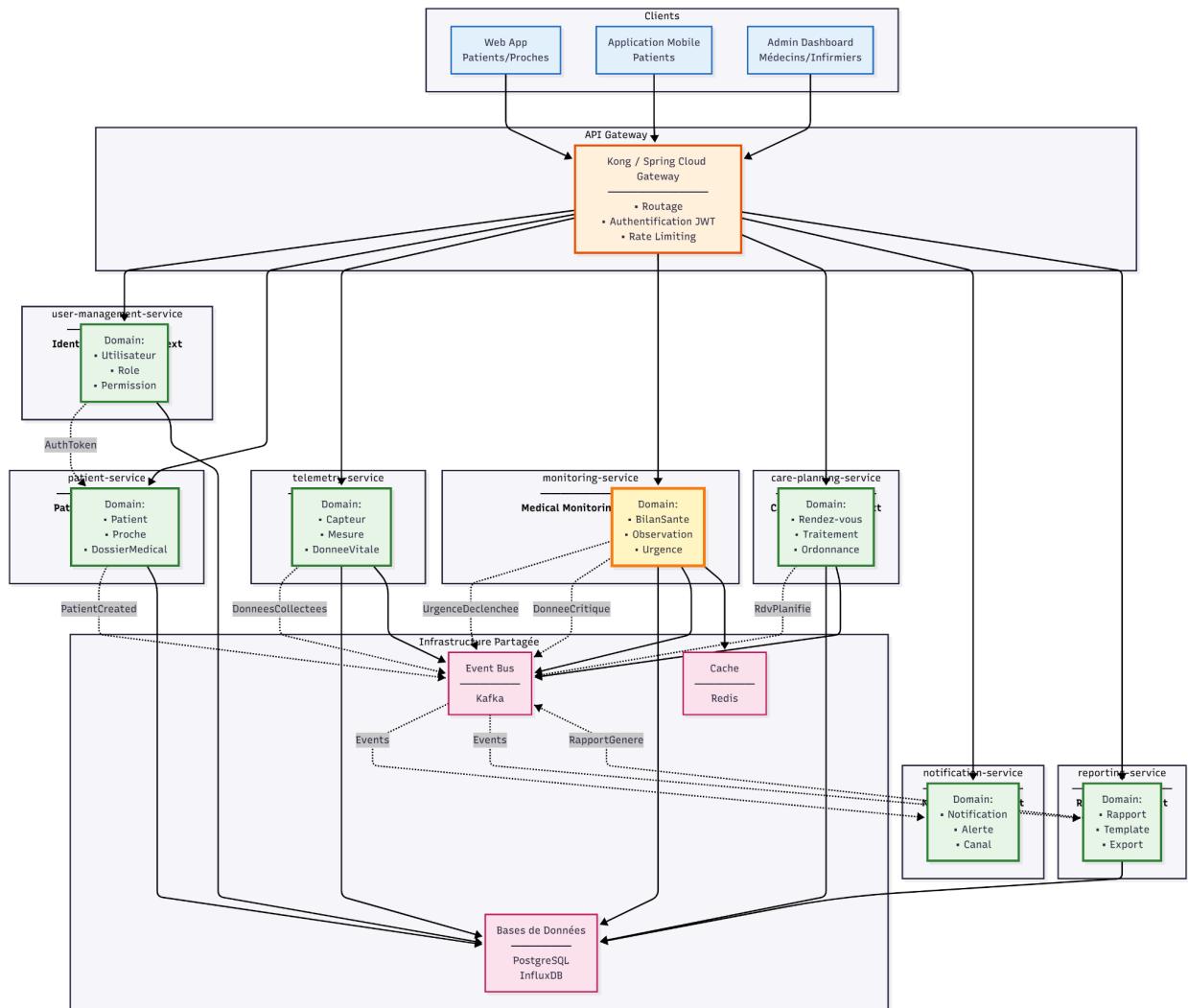


Contrats (exemples)

- `POST /api/patient/forms/{formId}/responses` (Patient)
- `GET /api/patient/{id}/summary?period=weekly` (Patient/Proche selon ACL)
- `GET /api/nurse/patients` (Infirmier)
- `GET /api/nurse/patient/{id}/signals?from=...&to=...` (Infirmier/Docteur)
- `POST /api/doctor/patient/{id}/plan` (Docteur)
- `POST /api/doctor/patient/{id}/alert-rule` (Docteur)
- `POST /api/alert/sos` (Watch/Phone → EMS)
- `POST /api/admin/users` (Admin)

Scénario de démo :

Pour ce POC, l'objectif est de démontrer une chaîne fonctionnelle essentielle de la plateforme e-santé. Le scénario couvrira l'inscription d'un nouveau patient, gérée par un administrateur qui crée les comptes. Ensuite, nous simulons les données provenant de 2 ou 3 capteurs connectés qui transmettent leurs mesures à un smartphone servant de passerelle. Un utilisateur, tel qu'un médecin, pourra visualiser ces informations sur un Dashboard et le système intégrera un mécanisme de notification pour déclencher une alerte en cas de détection de données critiques. Finalement on aura la génération automatique d'un rapport de santé pérío



Bounded Contexts

Pour votre projet, je vous suggère ces contextes :

1. Gestion des Patients (Patient Management)

- Entités : Patient, Proche, DossierMedical
- Responsabilités : inscription, profil, relations familiales

2. Télémétrie & Capteurs (Telemetry)

- Entités : Capteur, Mesure, DonneeVitale
- Responsabilités : collecte, validation, stockage des données capteurs

3. Surveillance Médicale (Medical Monitoring)

- Entités : BilanSante, Observation, Urgence

- Responsabilités : analyse des données, détection d'anomalies, alertes

4. Planification & Suivi (Care Planning)

- Entités : Rendez-vous, Traitement, Ordonnance
- Responsabilités : planification des soins, prescriptions

5. Notification & Alertes (Notification)

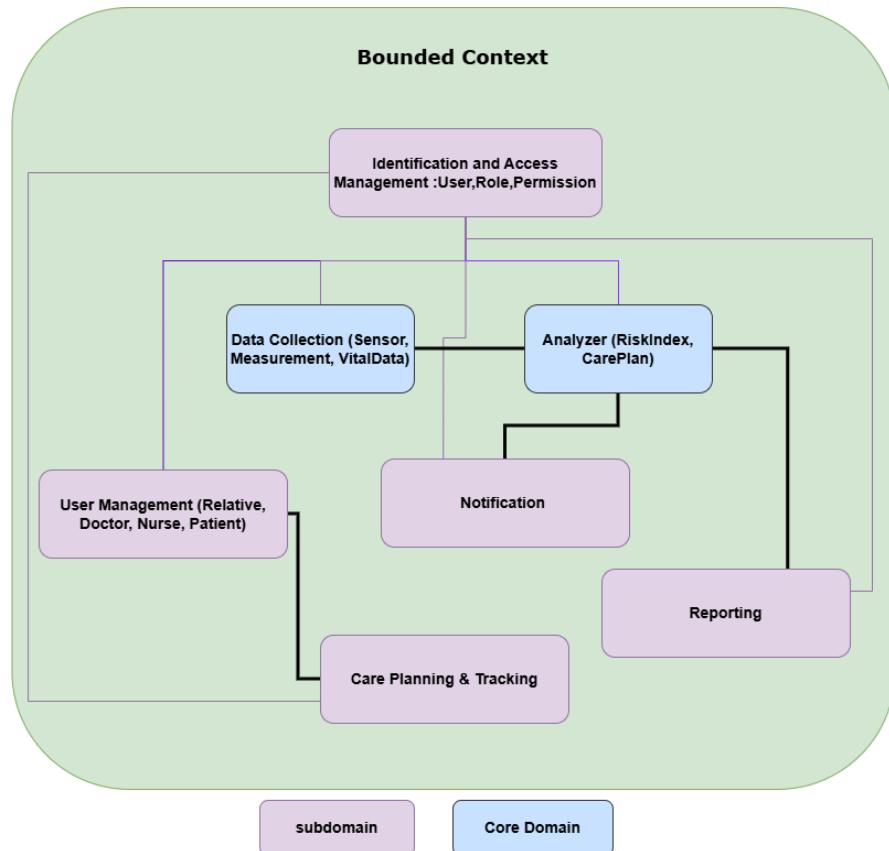
- Value Objects : Notification, Alerte
- Responsabilités : envoi multi-canal (SMS, email, push)

6. Gestion des Utilisateurs (Identity & Access)

- Entités : Utilisateur, Rôle, Permission
- Responsabilités : authentification, autorisation, RBAC

7. Rapports (Reporting)

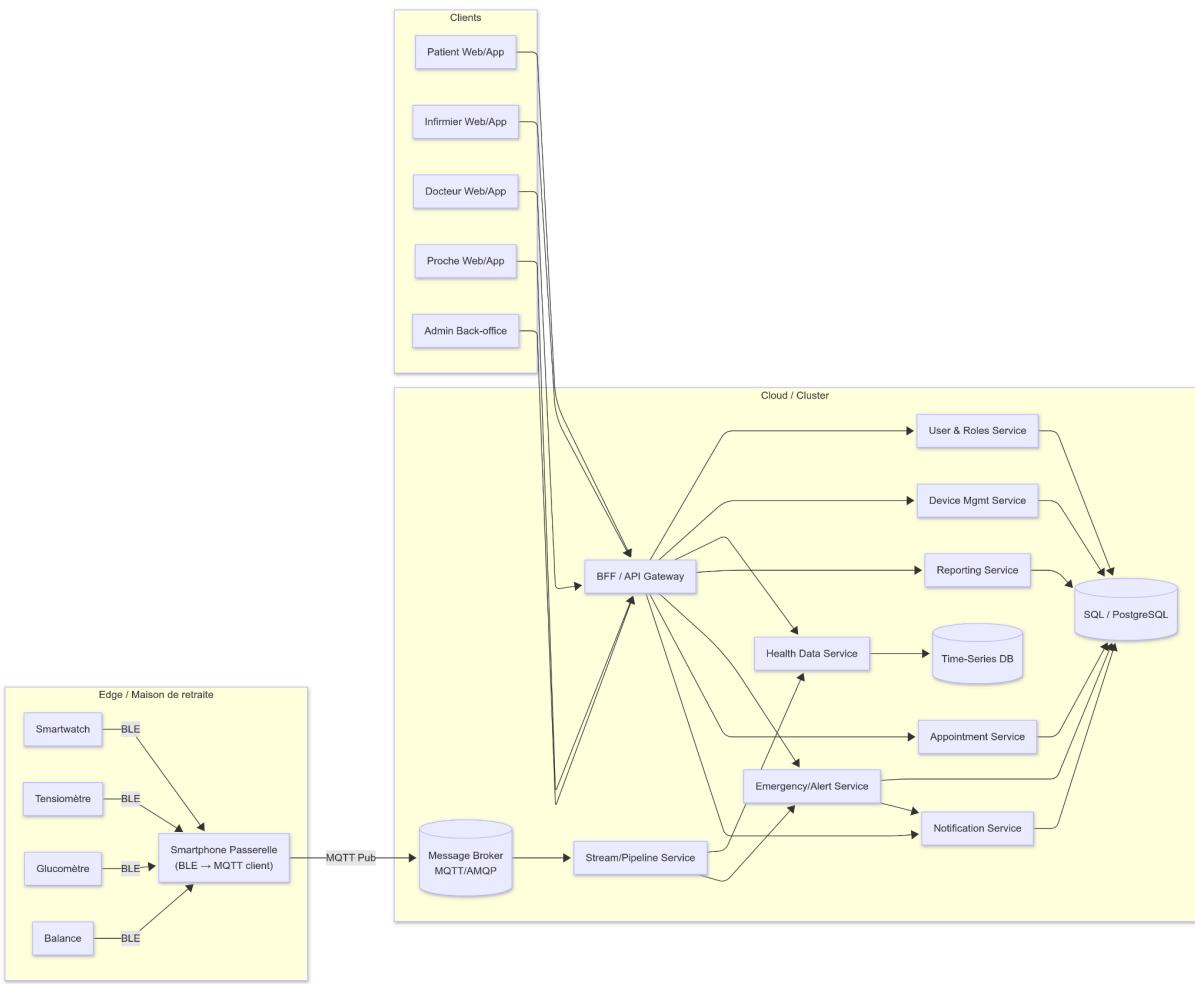
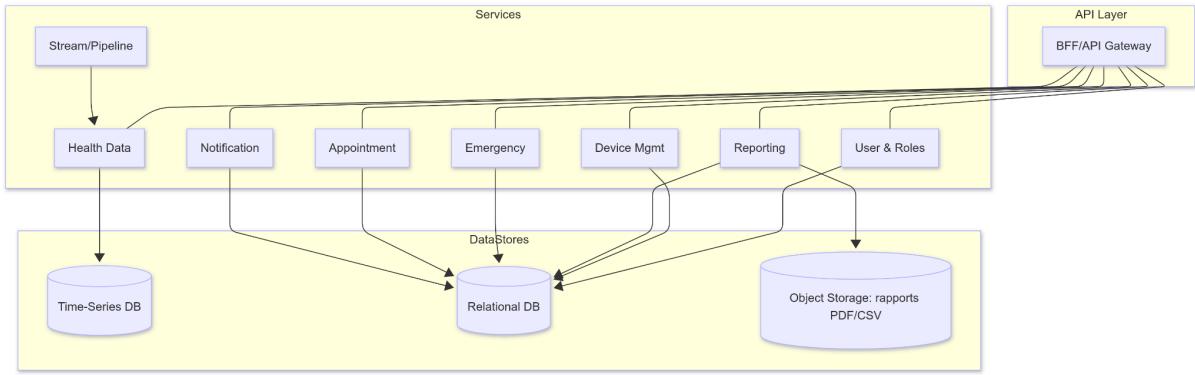
- Entités : Rapport, Template
- Responsabilités : génération automatique de bilans



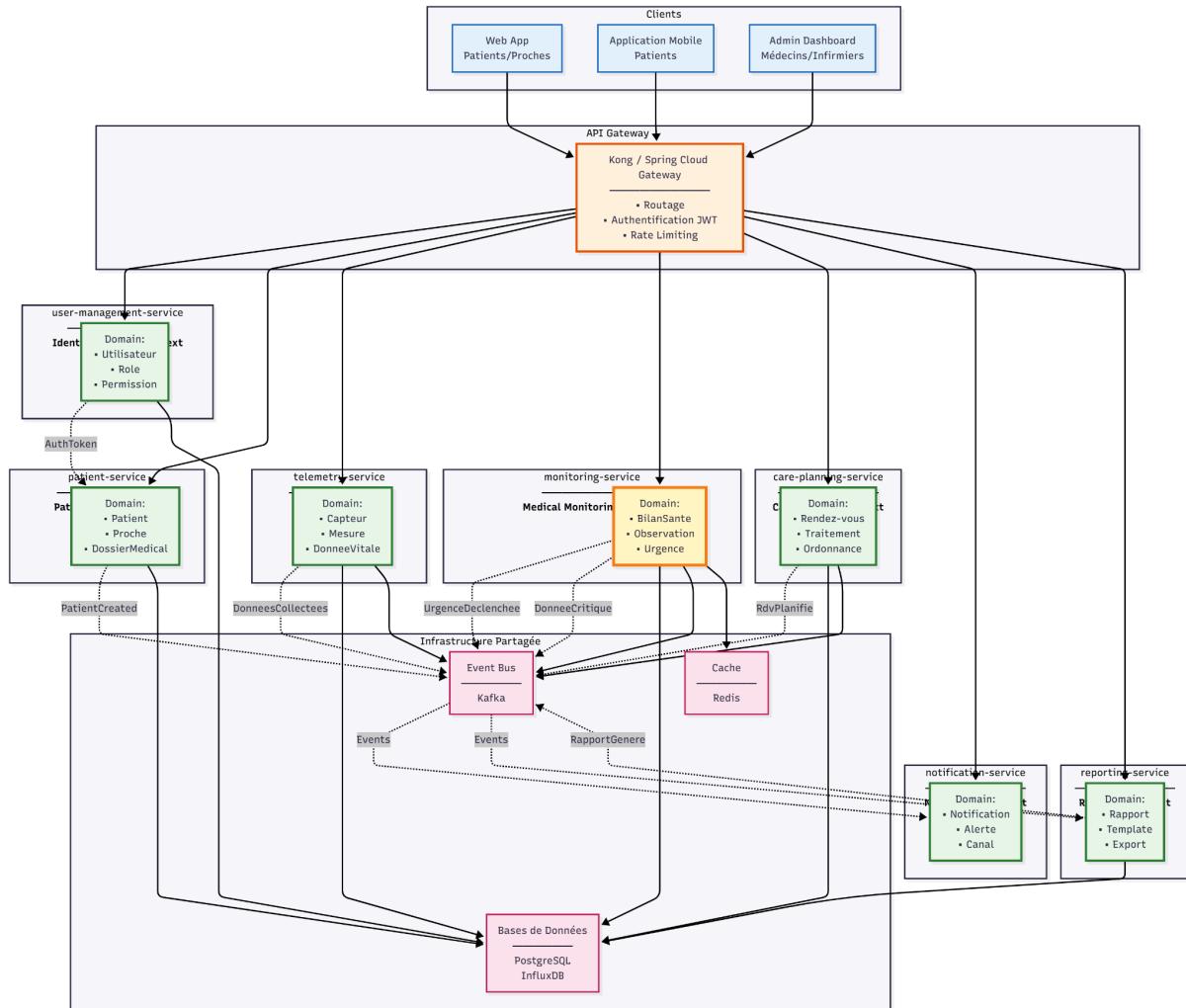
User Stories

Diagrams

Ancienne version - séance 1/10/2025 et 10/10/2025 (nécessite des améliorations)

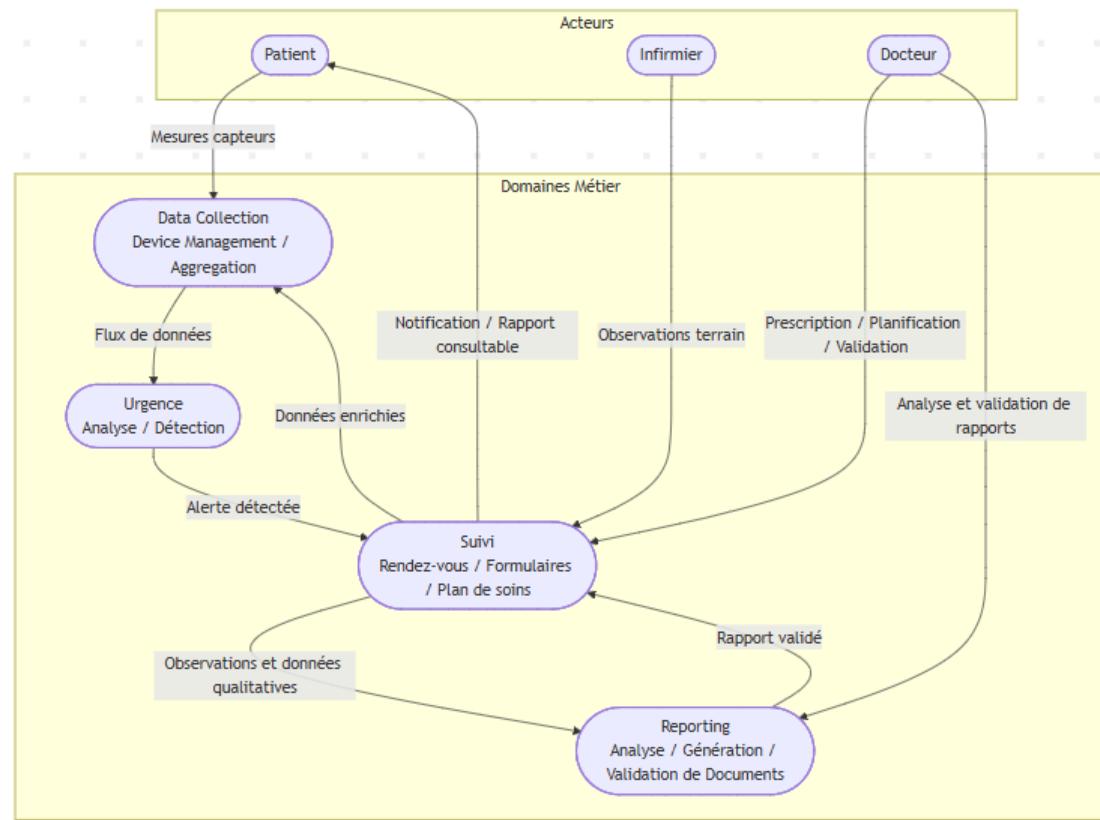


- un seul box
- webapp / front avant gateway
- c quoi un smartphone ?
- Type de liaison



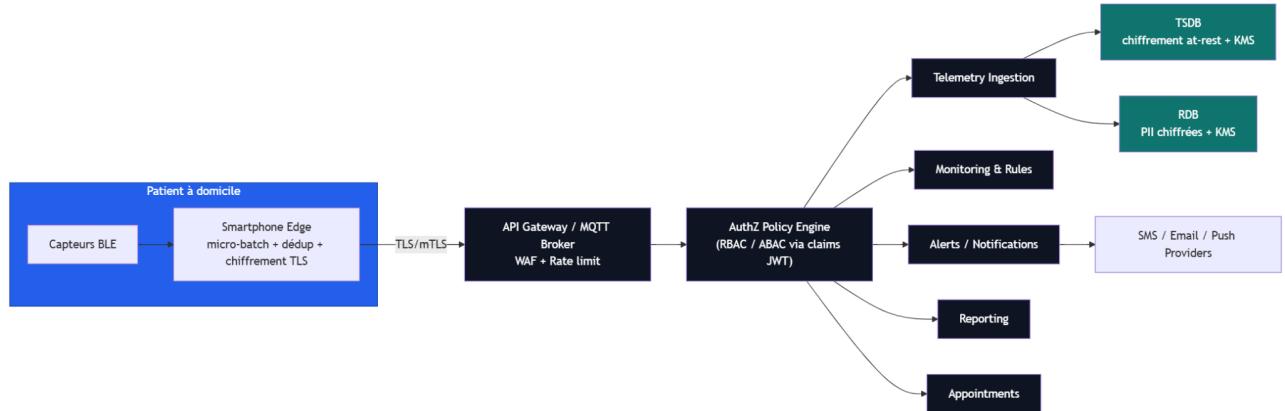
Séance 15/10/2025 :

1. DDD avec exemples de flow avec les acteurs.



Domaine	Responsable principal	Type d'activité
Data Collection	Système	Collecte automatique des données
Urgence	Système	Réaction temps réel (alertes)
Suivi	Docteur & Infirmier	Coordination humaine, planification, communication
Reporting	Docteur	Validation et génération de rapports

Diagramme de Sécurité



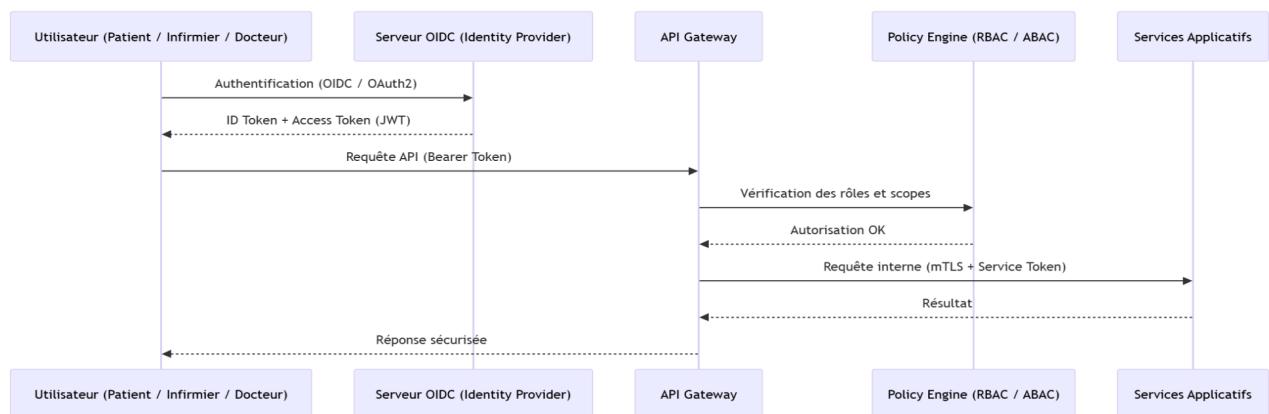
Ce diagramme illustre la **sécurité de bout en bout** du système.

Les données issues des capteurs sont chiffrées dès le smartphone (*Edge*), transmises via **TLS/mTLS** vers le cloud, et protégées par un **API Gateway** intégrant des règles WAF et de limitation de débit.

Les services cloud sont accessibles uniquement via des **politiques d'autorisation (RBAC/ABAC)** fondées sur les rôles et identités JWT.

Les données sensibles (PII, mesures médicales) sont **chiffrées au repos** dans les bases (TSDB/RDB) via un **KMS**, garantissant confidentialité et conformité RGPD.

Diagramme d'authentification



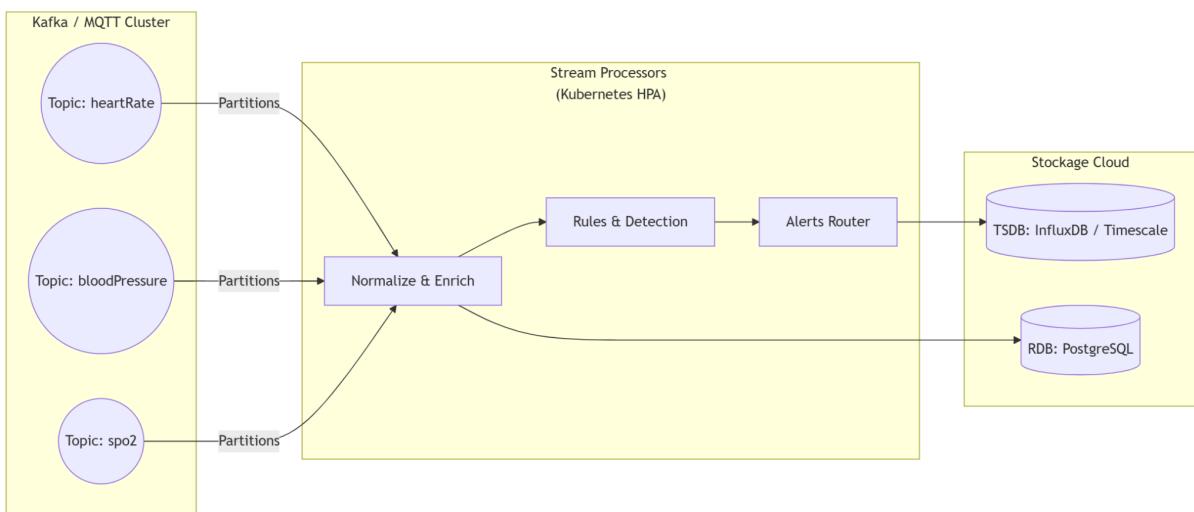
L'authentification repose sur le protocole **OIDC / OAuth2**.

Chaque utilisateur (patient, infirmier, docteur) s'identifie auprès d'un **Identity Provider**, qui renvoie un **token JWT** contenant son rôle et son *tenantId*.

Ce token est ensuite transmis à l'API Gateway, où un moteur de politiques (**Policy Engine**) valide les droits avant tout accès.

Les communications inter-services utilisent des **tokens techniques et mTLS**, assurant une séparation stricte des responsabilités et une authentification de machine à machine sécurisée.

Diagramme de Scaling



L'architecture est pensée pour être **hautement scalable** grâce à une approche événementielle.

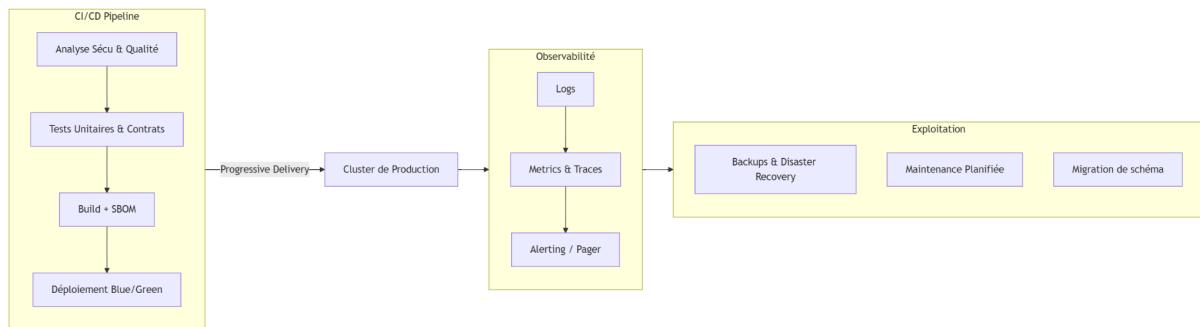
Les flux de données sont distribués par **Kafka** ou **MQTT**, partitionnés par *patientId* afin de paralléliser les traitements.

Les **processseurs de flux** (containers Kubernetes) se répliquent automatiquement selon la charge via **Horizontal Pod Autoscaler (HPA)**.

Les bases TSDB et SQL absorbent efficacement les écritures séquentielles et permettent un accès rapide aux données agrégées.

Ce modèle garantit la **montée en charge horizontale** sans dégradation de performance, même avec des milliers de patients connectés simultanément.

Diagramme de Maintenance



La maintenance du système repose sur une intégration continue robuste.

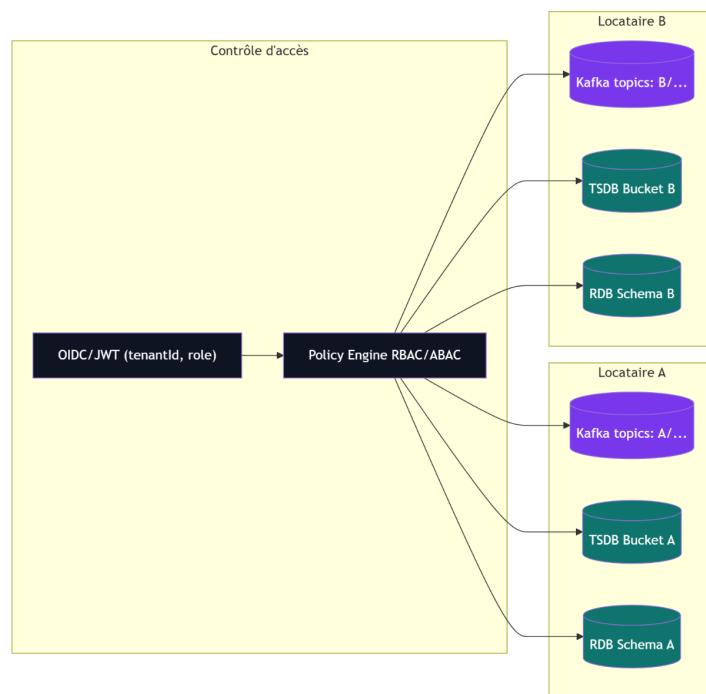
Chaque mise à jour passe par un pipeline **CI/CD** avec analyse de sécurité, tests automatiques et déploiement progressif **Blue/Green**, garantissant une **disponibilité continue**.

L'observabilité (logs, métriques, traces) permet de détecter les anomalies avant impact utilisateur.

Des **backups chiffrés**, des **tests de reprise après sinistre (DR)** et des **fenêtres de maintenance planifiées** assurent la fiabilité à long terme de la plateforme.

L'objectif est un système **auto-surveillé et tolérant aux pannes**.

Diagramme Multi-Tenant



Le système est **multi-tenant**, conçu pour desservir plusieurs structures de soins (cliniques, associations, réseaux infirmiers).

Chaque locataire dispose de ses propres **topics Kafka**, **buckets TSDB** et **schémas SQL**, assurant une isolation logique des données.

Les requêtes et autorisations sont filtrées par le **claim tenantId** dans le token JWT.

Le moteur de politiques (**Policy Engine**) garantit qu'un utilisateur n'accède qu'aux données de son propre tenant.

Cette approche réduit les coûts tout en préservant la **sécurité inter-locataires** et la **scalabilité** de la plateforme.

Note prof - 15/10/2025

ATTENTION :

- Ça ne doit pas être un EHPAD, *les personnes âgées restent chez elles.*
 - **Contexte : Une solution d'e-santé.**
 - Aider les personnes âgées à rester à la maison.
 - Partager des informations avec certaines personnes.
- Spécifier objectif métier pour les infirmiers, docteurs ...etc
- types d'alerte à traiter (notification etc ..)
- Définir la valeur ajoutée :
 - ex : éviter d'aller à l'hôpital
- Spécifier les hypothèses pour le bon fonctionnement du système:
 - exemple : Si on récupère des données du Smart Watch, on doit fournir ça avec la solution, idem pour les autres capteurs.
 - Est-ce qu'on va fournir les capteurs ou intégrer l'existant avec le système ?
- Edge chez le patient
- Définir les topics (topic par métrique / capteur)
- Scénario démo validé
- Pour les capteurs il faut préciser à quel patient ils appartiennent les données : Pour cela utiliser un modèle hiérarchique ex : data/<patient_id>/<sensor_id>/<metric>
- Pour Kafka: il a mentionné d'utiliser une DB qui va avec Kafka la plupart de temps dans plusieurs architectures
- Système d'authentification et sécurisation des endpoints au fonction de qui a droit de faire quoi
- Penser à la notion de maintenance : Installer et Maintenir la solution