

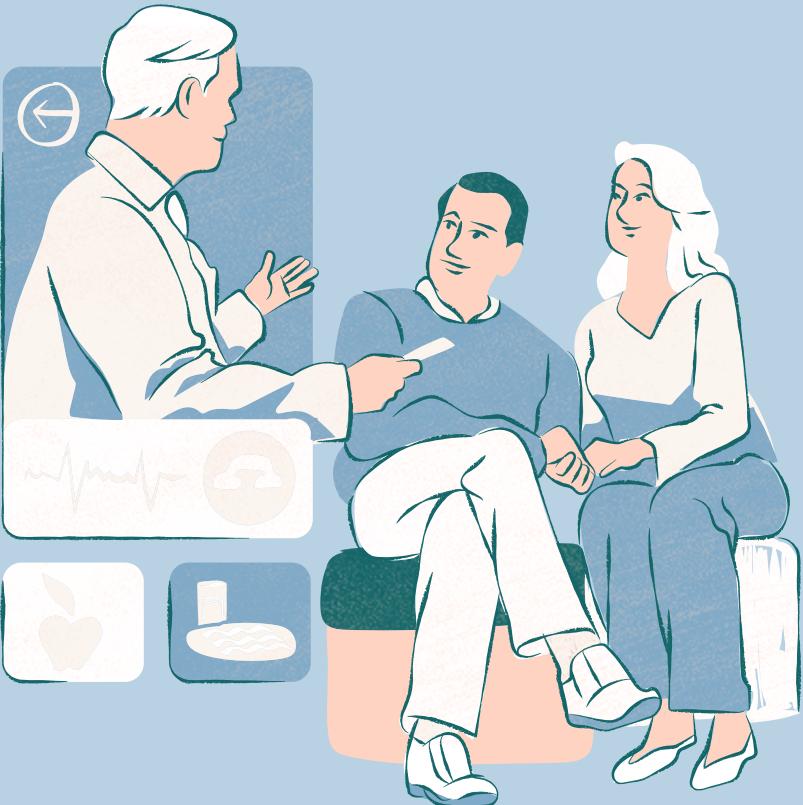
Présentation Architecture Logicielle

Camille Antonios
Amélie Muller
Gauthier Martin
Hajar El Gholabzouri
Jessica Kahungu

Novembre 2025



Introduction

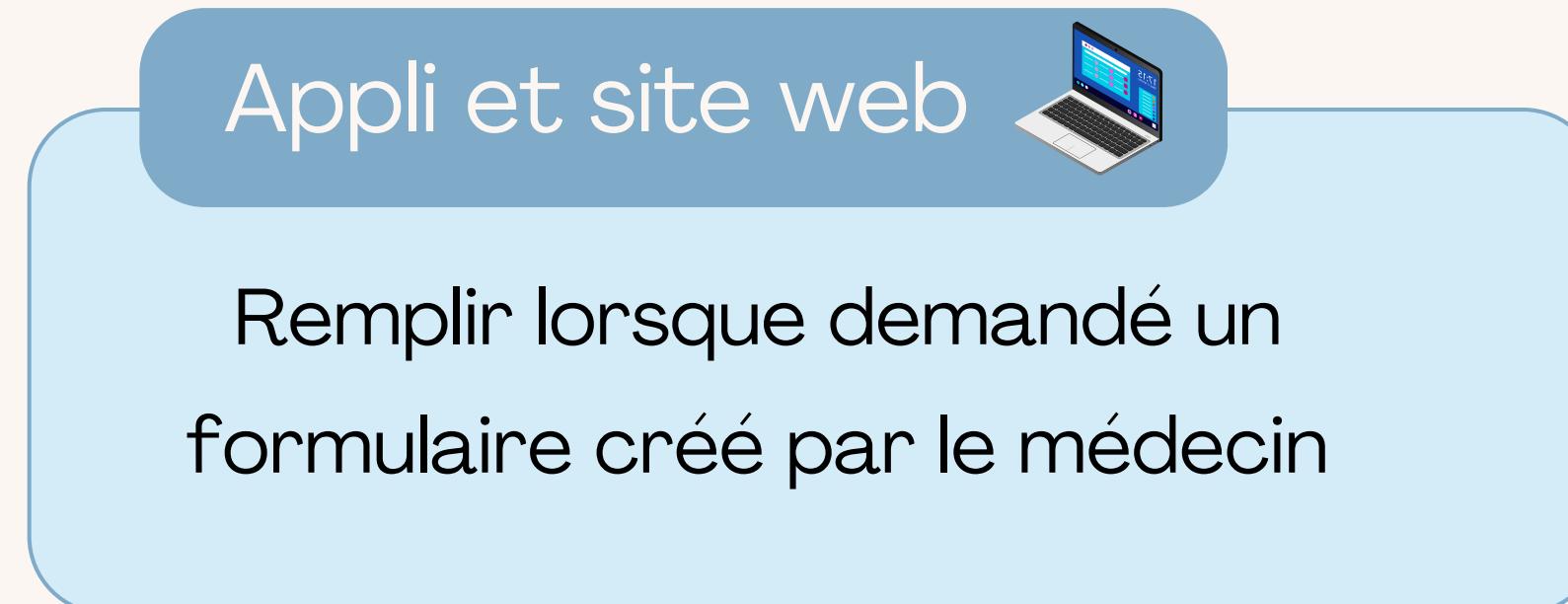
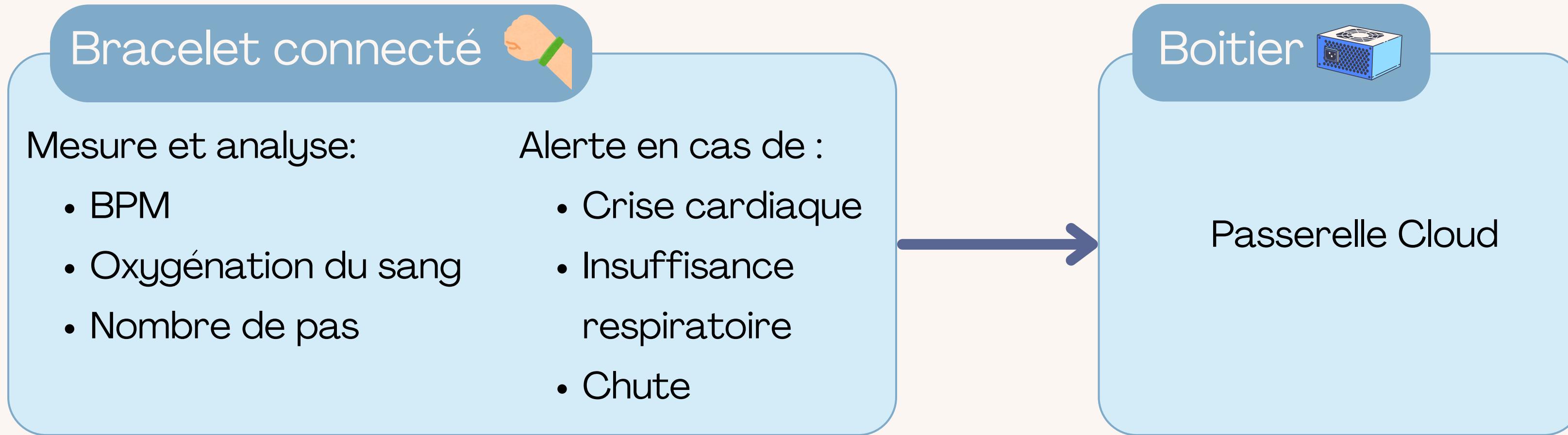


Client : le patient

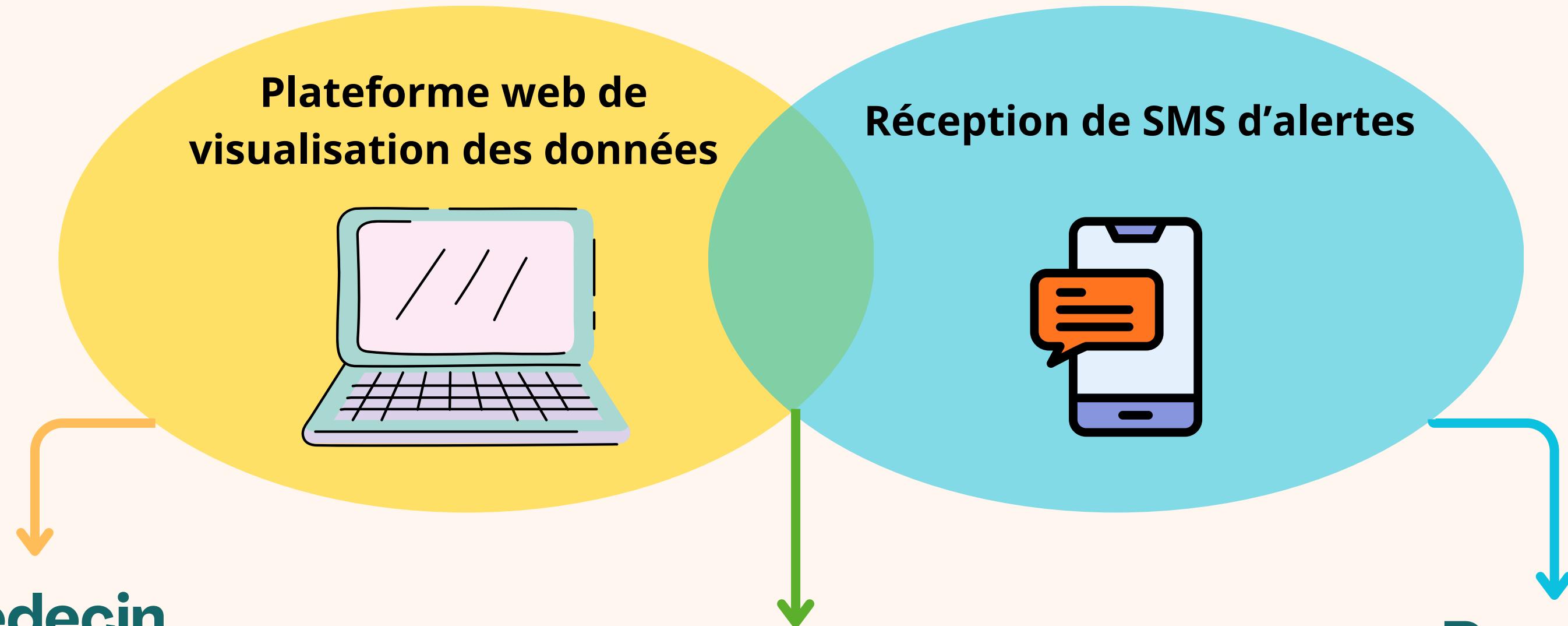
- Solution de facilitation de suivi et de maintien des personnes âgées à domicile dont l'état de santé se dégrade :
 - **Maintien à domicile** : système d'alertes en cas de situations graves
 - **Suivi médical** : collecte automatique de données qui facilite les passages de l'infirmière et le suivi par le médecin
- Prescription du dispositif par le médecin au patient
- Quelques hypothèses :
 - Monitoring uniquement à domicile
 - Vivent seuls, seuls individus monitorés
 - Santé fragile, mais autonomie partielle
 - Visite infirmière quotidienne ou tous les deux jours
 - Connexion Internet et équipement (mobile ou ordinateur)
- Gestion de 150 000 utilisateurs (100 000 patients, 30 000 infirmiers et 20 000 médecins)

Contexte: Solution proposée

Chez le patient :



Contexte: Solution proposée



Médecin

- Accès à une version détaillée de la plateforme
- Réception de rapports automatiques, récapitulant les dernières données mesurées

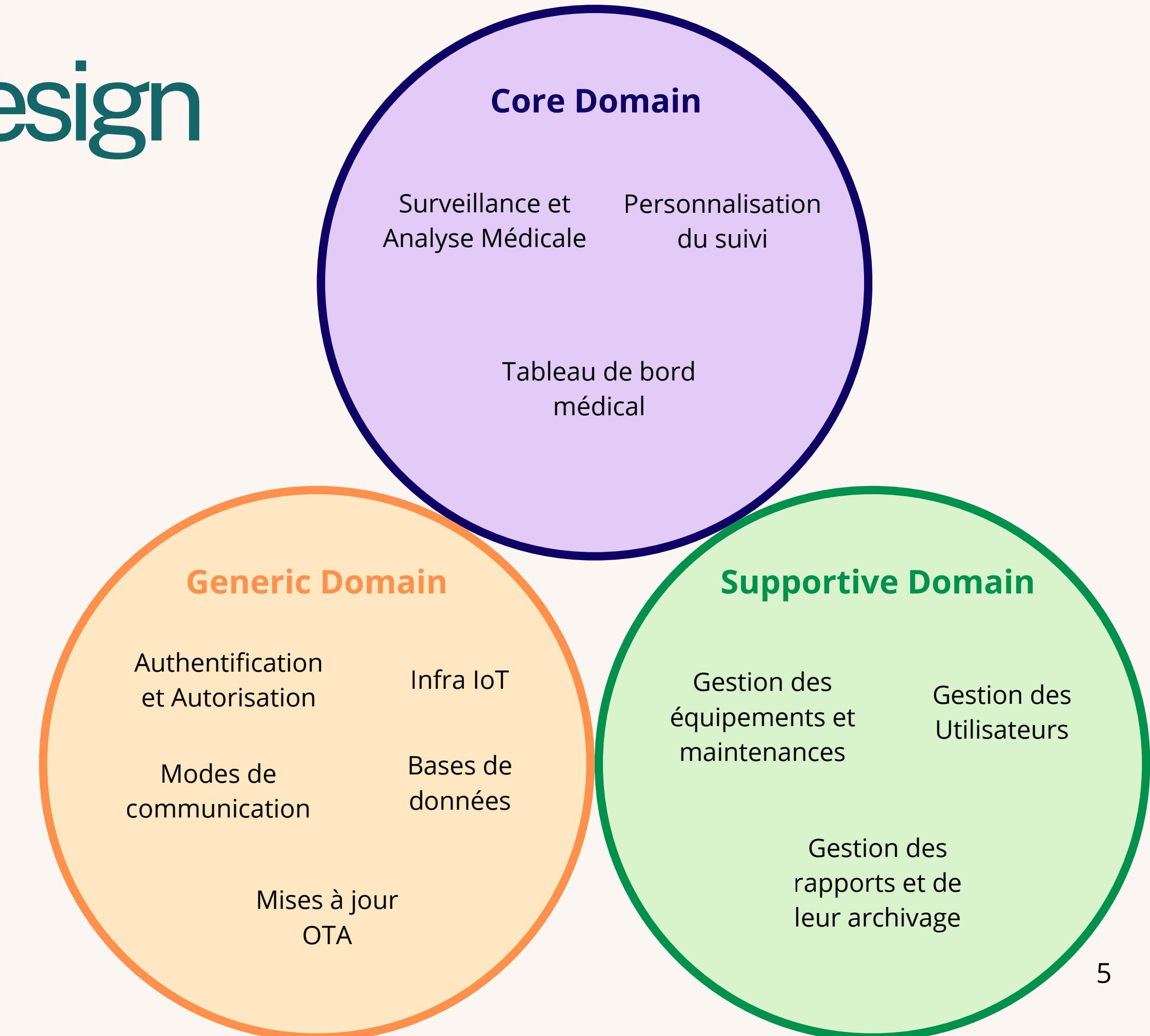
Infirmière

- Accès à une plateforme web simplifiée
- Réception des alertes par SMS

Proche

- Réception des alertes par SMS

Domain Driven Design



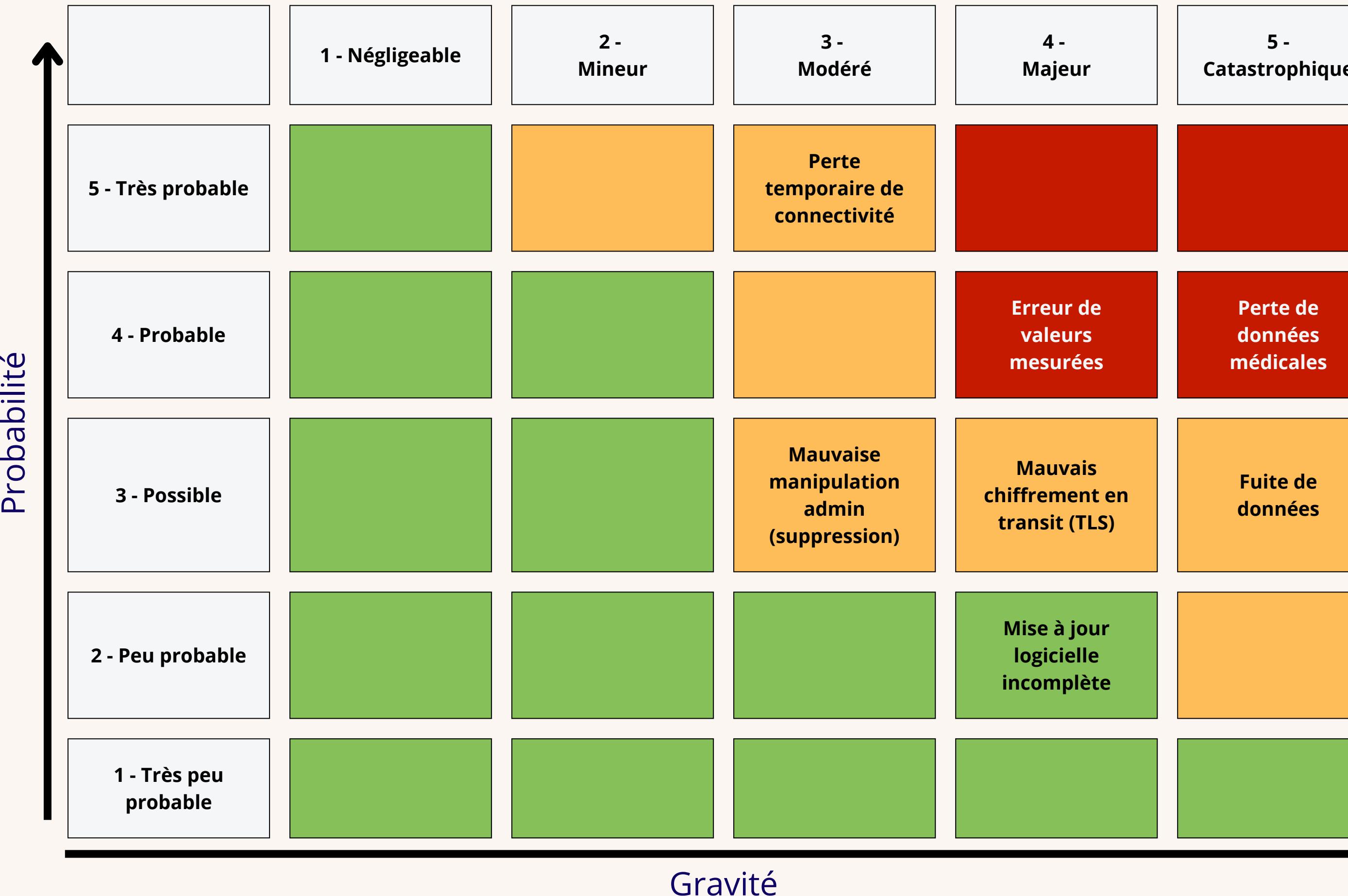
Analyse des risques: Tableau FMEA

Légende

Risque (Failure Effect)	Étape concernée	Mode de défaillance potentiel	Causes possibles	Effet sur le système / l'utilisateur	SEV	OCC	DET	RPN	Contrôles existants	Actions recommandées
Perte de données médicales	Transmission boîtier → Cloud / transmission bracelet → boîtier	Données non envoyées	Panne boîtier, perte réseau, panne électrique	Données manquantes dans la base, absence d'alerte santé	9	6	3	162	Système détecte l'absence de données pendant X temps	Alerter le support + redondance locale (buffer) pour renvoyer lorsque possible
Altération/ corruption de données	Traitement dans backend Kafka / InfluxDB	Données modifiées ou incomplètes	Bug logiciel, mauvaise sérialisation JSON, bug de timestamp	Rapports faussés, fausses alertes	8	3	4	96	Validation du format et checksum avant insertion	Vérification d'intégrité, validation schéma JSON, tests
Fuite de données (Data breach)	Transfert de données/ stockage	Accès non autorisé, exfiltration	Piratage BDD, faille Cloud, compte admin compromis	Violation RGPD, perte de confiance des utilisateurs	10	3	7	210	Données chiffrées, authentification forte	Rotation des clés (de DB par exemple), audit sécurité, tests de pénétration
Erreur de valeur des données mesurées (valeurs aberrantes)	Données acquises par le capteur	Données bruitées ou fausses	Défaillance capteur	Fausse alerte ou absence d'alerte, rapports faussés	8	6	4	192	Détection de valeurs hors plage, nettoyage des données impossibles	Analyse statistique automatique pour détecter une fréquence anormale de valeurs

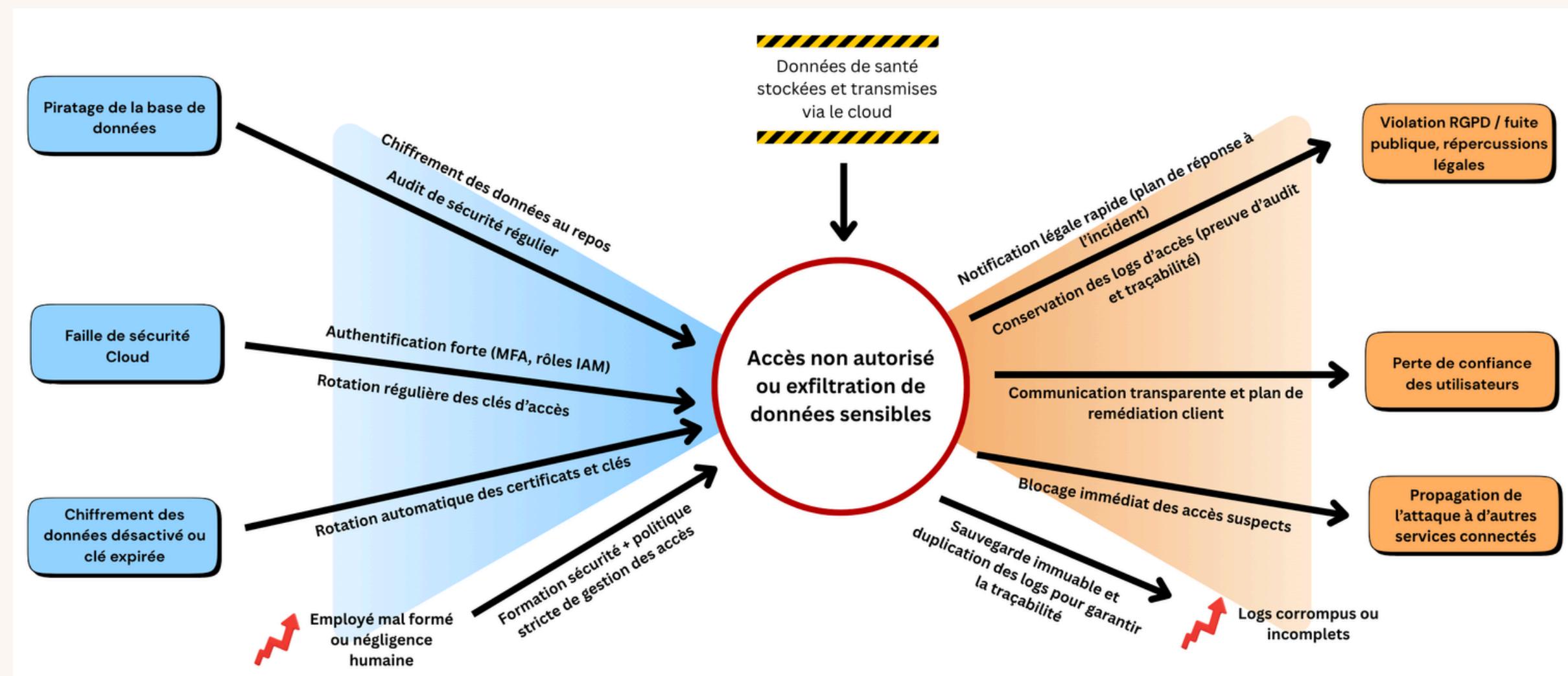
- SEV (Severity) : gravité de l'impact si le risque se produit (1 = faible, 10 = critique).
- OCC (Occurrence) : probabilité d'apparition du risque sur la durée de vie du système (1 = rare, 10 = fréquent).
- DET (Detection) : probabilité de détection avant l'impact utilisateur (1 = facilement détectable, 10 = difficile à détecter).
- RPN (Risk Priority Number) = $SEV \times OCC \times DET$ (plus il est élevé, plus une action est prioritaire).
- Actions recommandées : mesures préventives ou correctives à mettre en œuvre pour réduire la probabilité ou la gravité du risque.

Analyse des risques : Matrice de risques



Analyse des risques: fuite de données sensibles

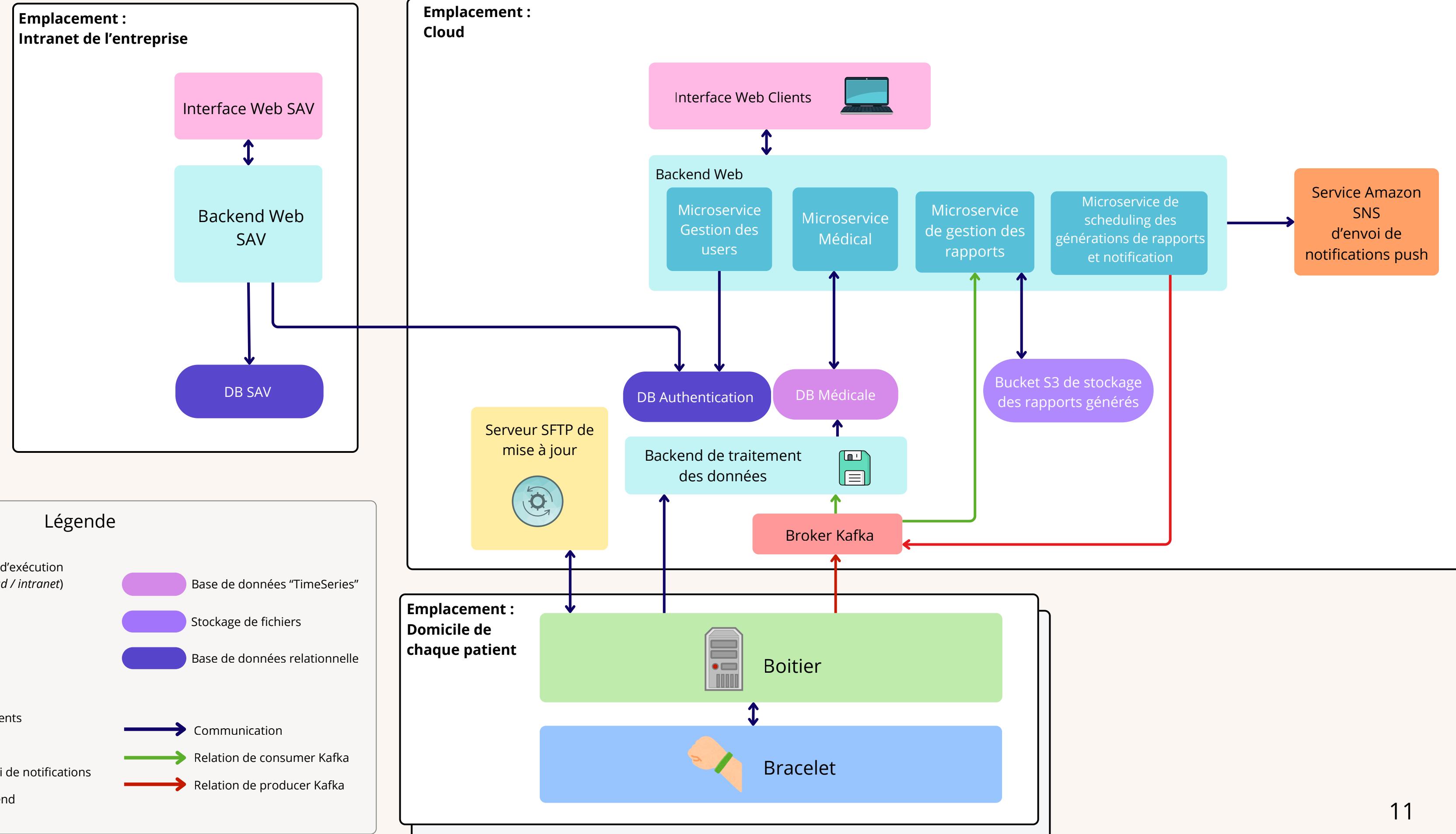
Risque (failure effect)	Etape concernée	Mode de défaillance potentiel	Causes possibles	Effet sur le système/ utilisateur	SEV	OCC	DET	RPN	Contrôles existants	Actions recommandées
Fuite de données	Transfert de données / stockage	Accès non autorisé aux données	Piratage BDD, faille Cloud...	Violation RGPD, perte de confiance	10	3	3	90	Données chiffrées, authentification forte	Rotation des clés DB, audits sécurité, tests de pénétration



Impacts architecturaux liés aux risques

Risque prioritaire	Décision d'architecture prise
Retard de traitement des données critiques	Quand le bracelet détecte crise/chute, il envoie au boîtier (contenant une carte SIM) qui expédie immédiatement le SMS aux proches (sans passer par le cloud)
Interruption/dégradation du service d'ingestion des données par pics de requêtes synchrones	Adoption de Kafka pour de l'ingestion asynchrone (mécanisme de file d'attente, retries...)
Erreur de valeurs mesurées	Contrôle de plausibilité de données et filtrage des outliers côté backend

Architecture

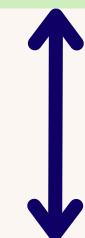


Architecture : IoT



Boitier

- Capteur BLE et bouton d'appairage
- Ecran E-paper
- Carte SIM et Modem GSM
- Nano-ordinateur
- Port Ethernet et alimentation USB-C



Communication BLE



Bracelet

- Capteurs :
 - Capteur IMU : pas, chutes
 - Capteur PPG : BPM et SpO2
- Ecran E-paper
- Capteur BLE et bouton d'appairage
- Port USB-C de recharge

- Taches : FreeRTOS
 - Mesures périodiques
 - Détection de situations de danger

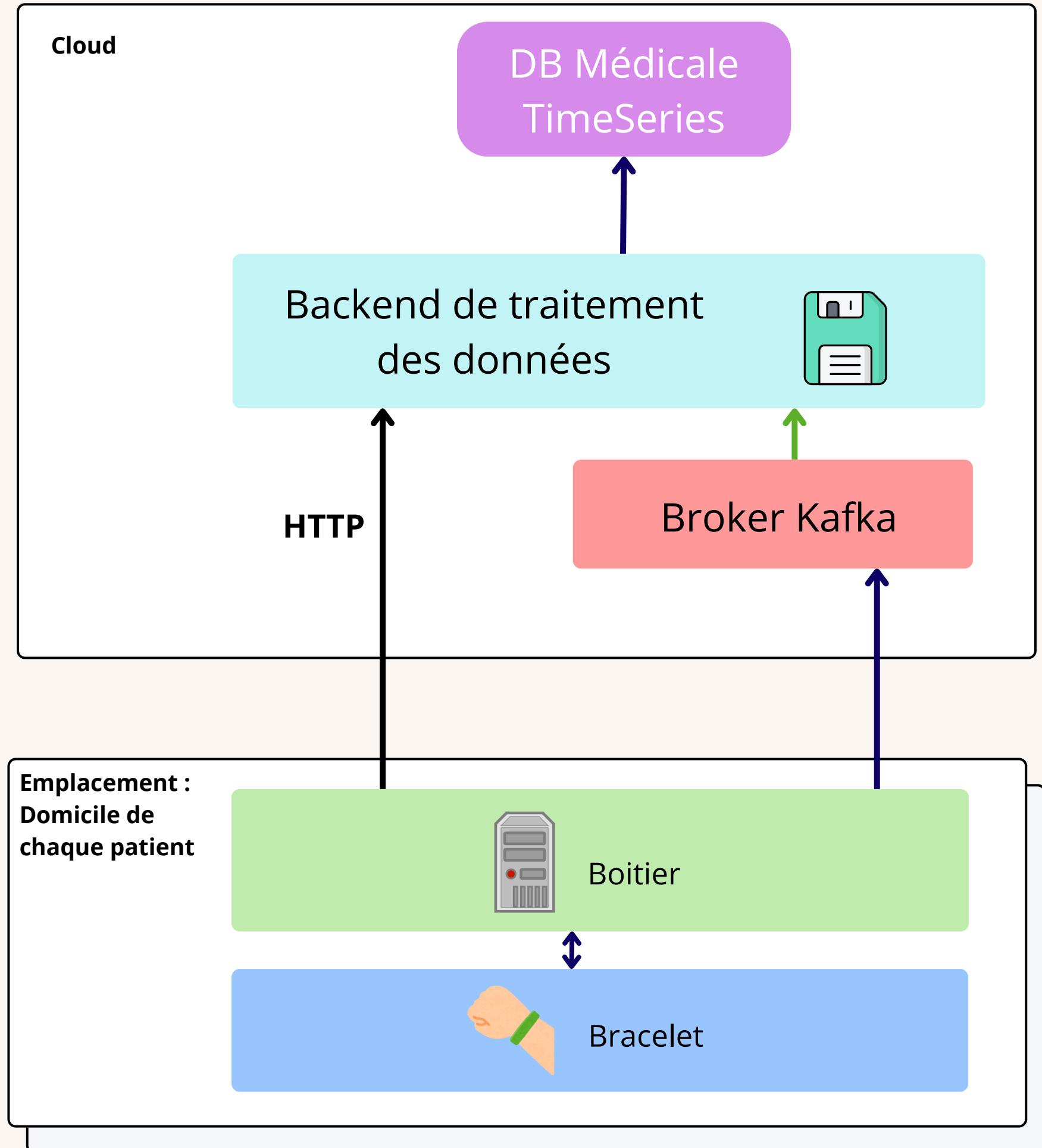
IoT device

IoT gateway

Architecture :

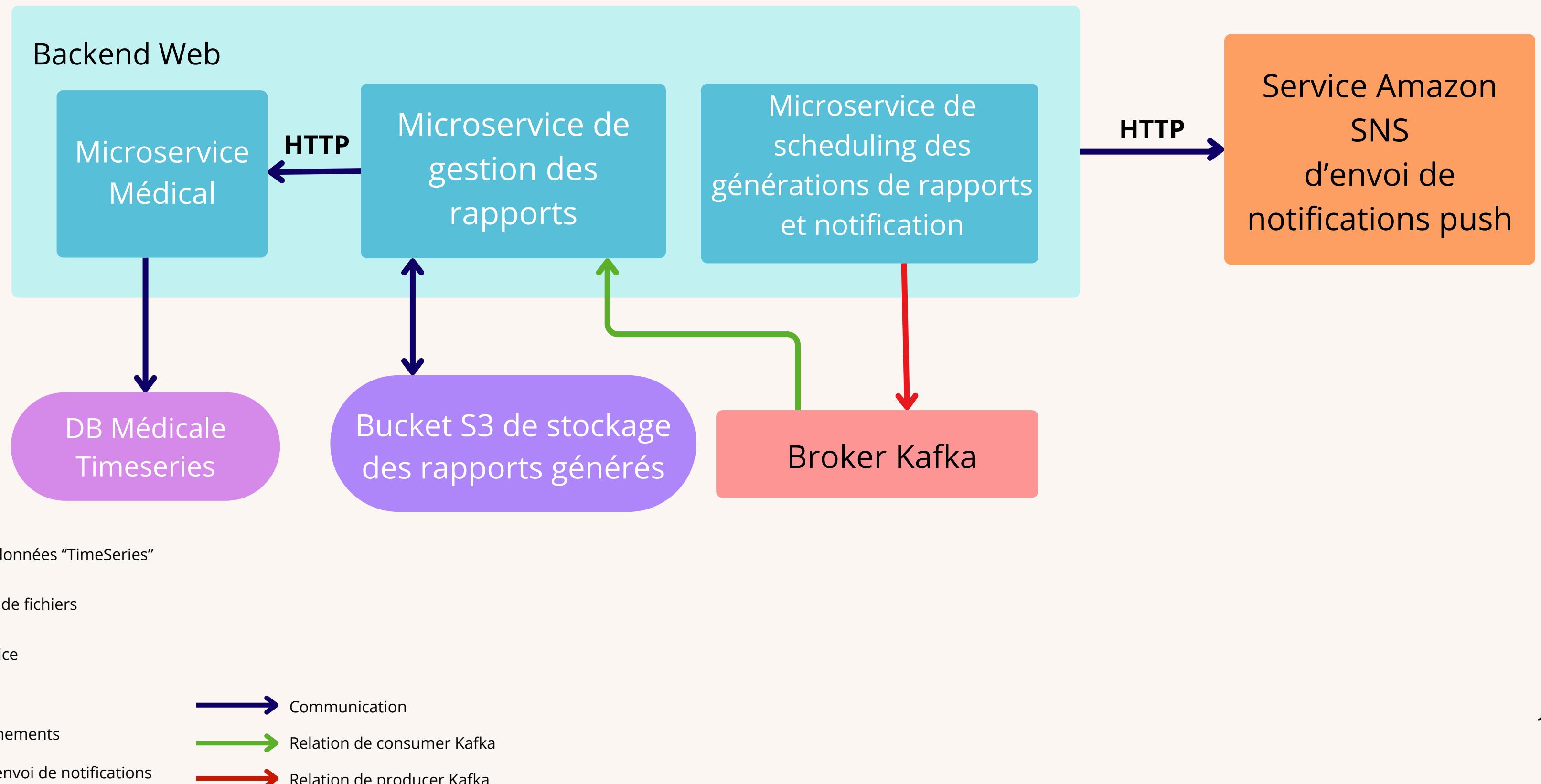
Transmission et stockage des données médicales

- Emplacement d'exécution (*domicile / cloud / intranet*)
- IoT device
- IoT gateway
- Backend
- Bus d'évènements
- Base de données "TimeSeries"
- Communication
- Relation de consumer Kafka



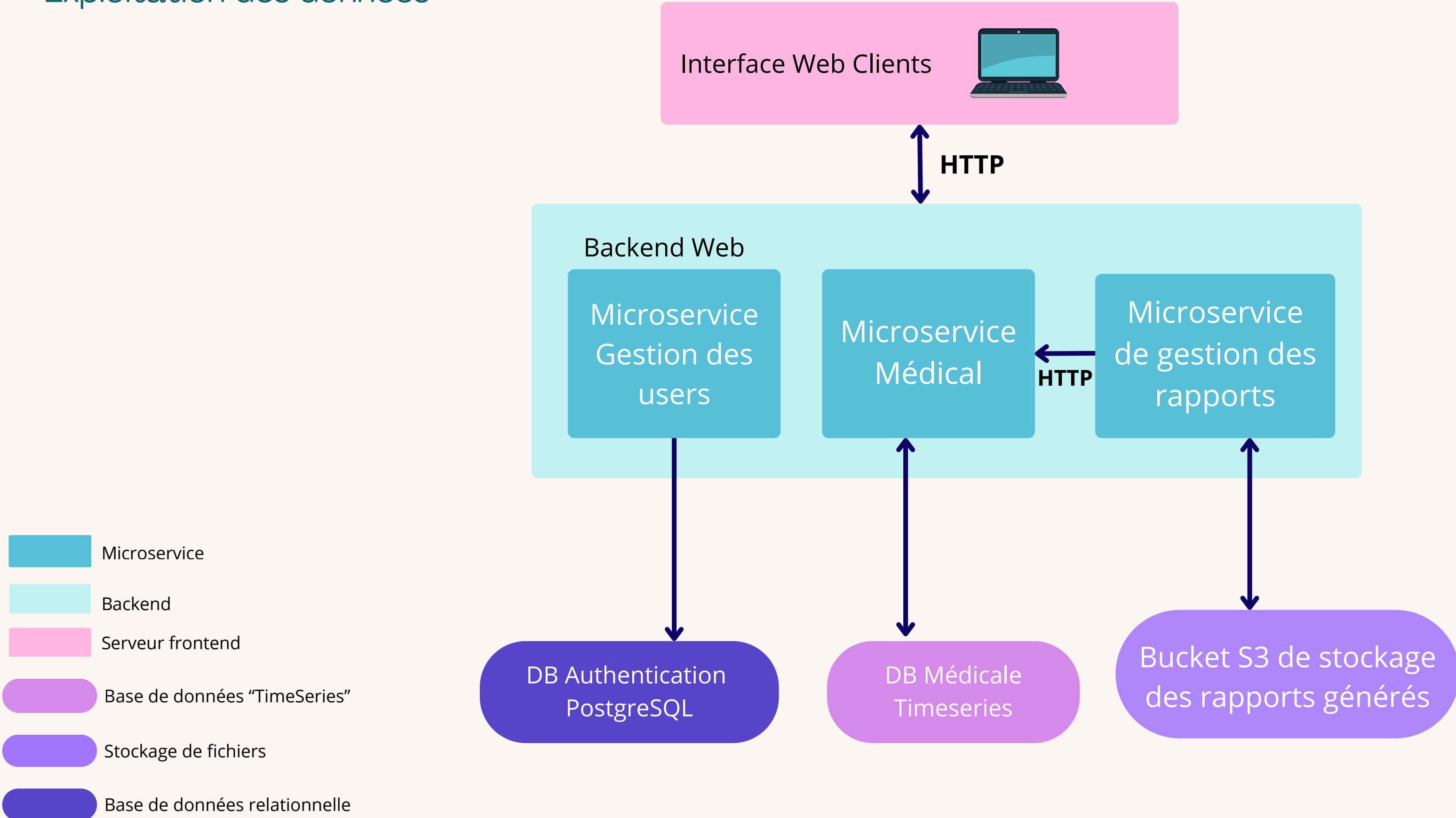
Architecture :

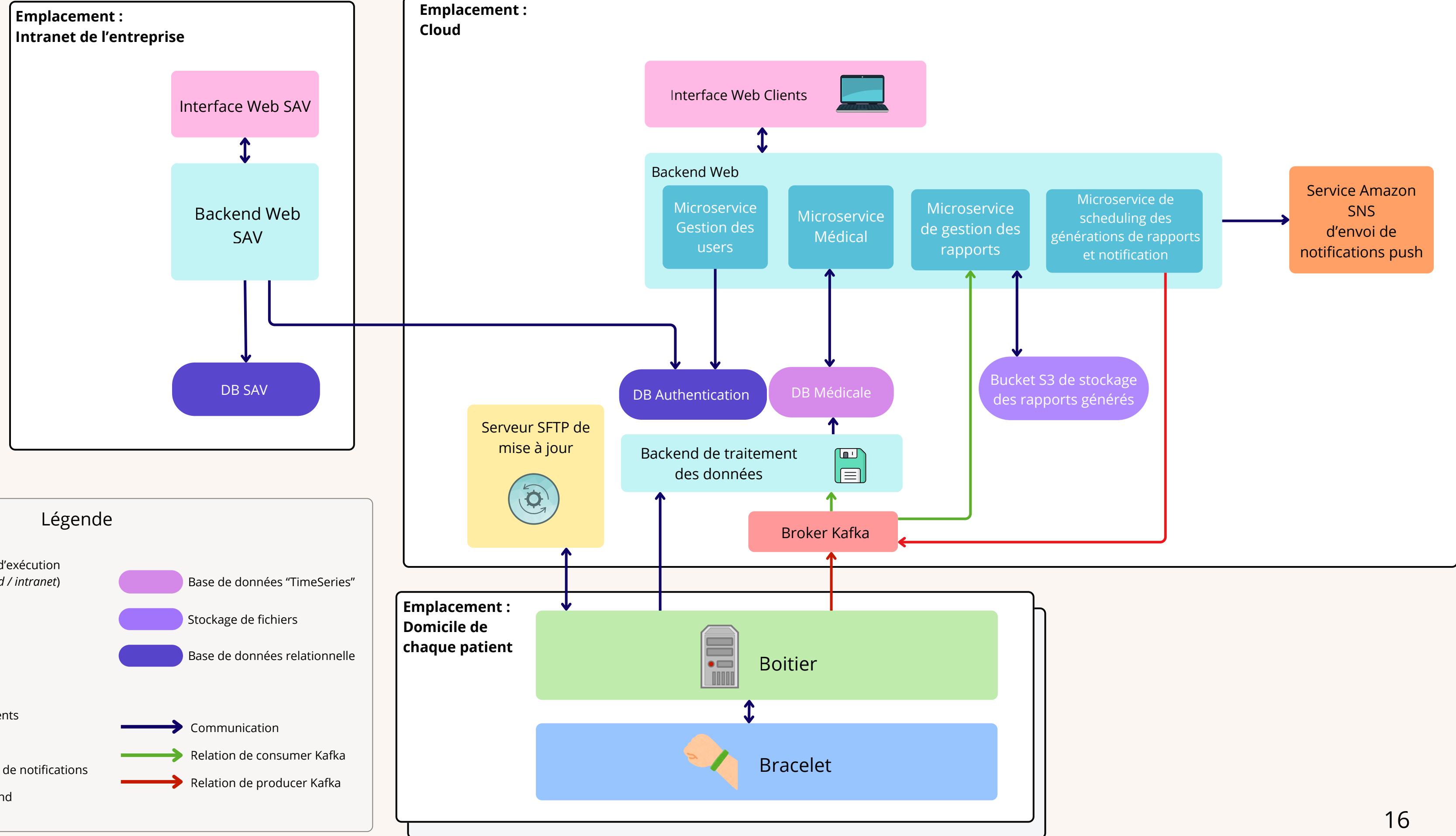
Génération automatique des rapports



Architecture :

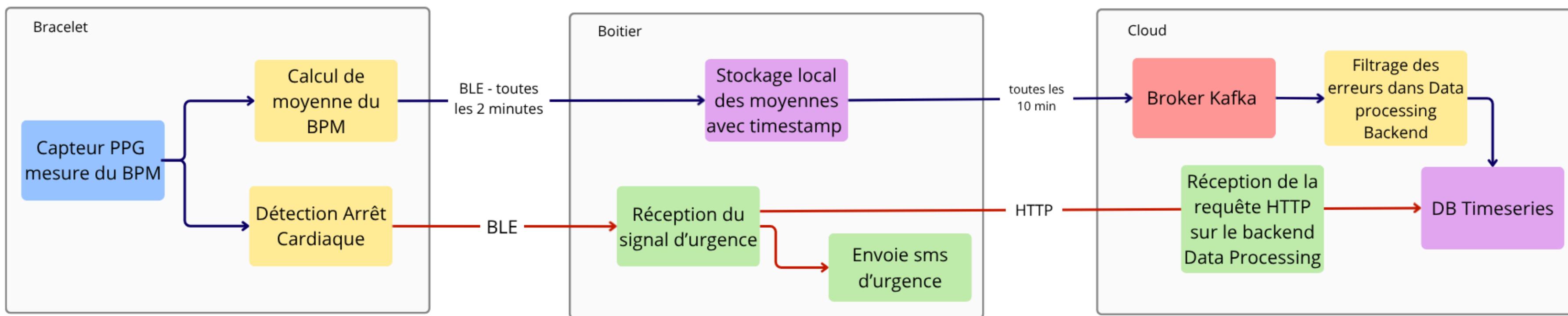
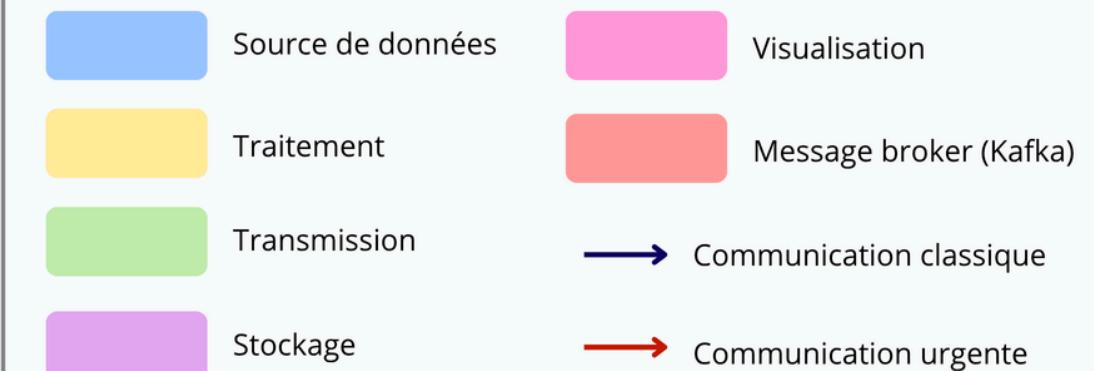
Exploitation des données





Flux de données BPM

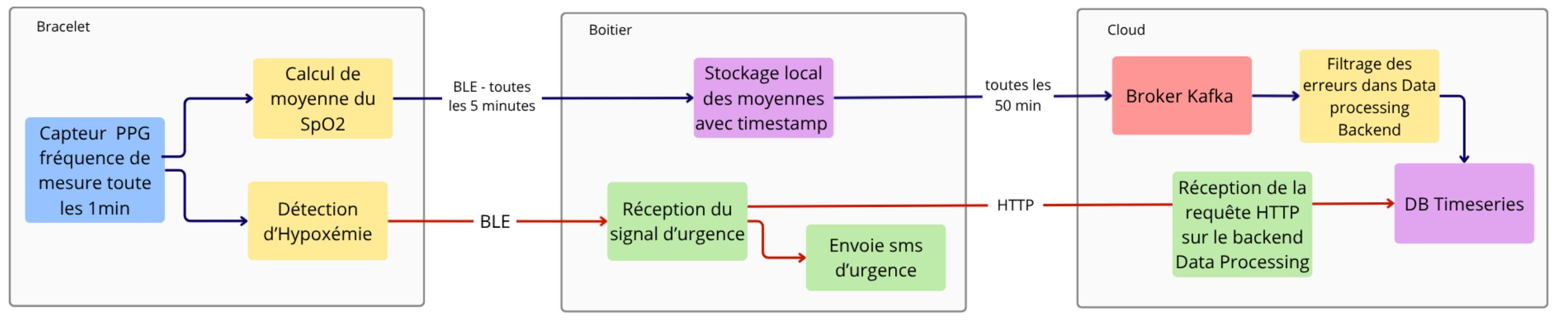
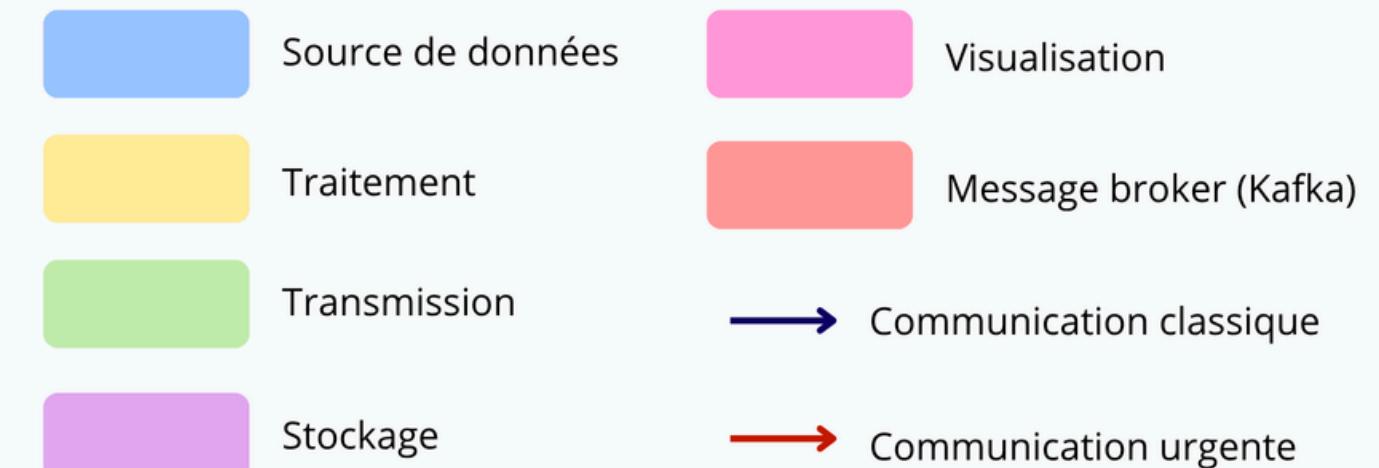
Légende des diagrammes de data pipelines



Flux de données

SpO2

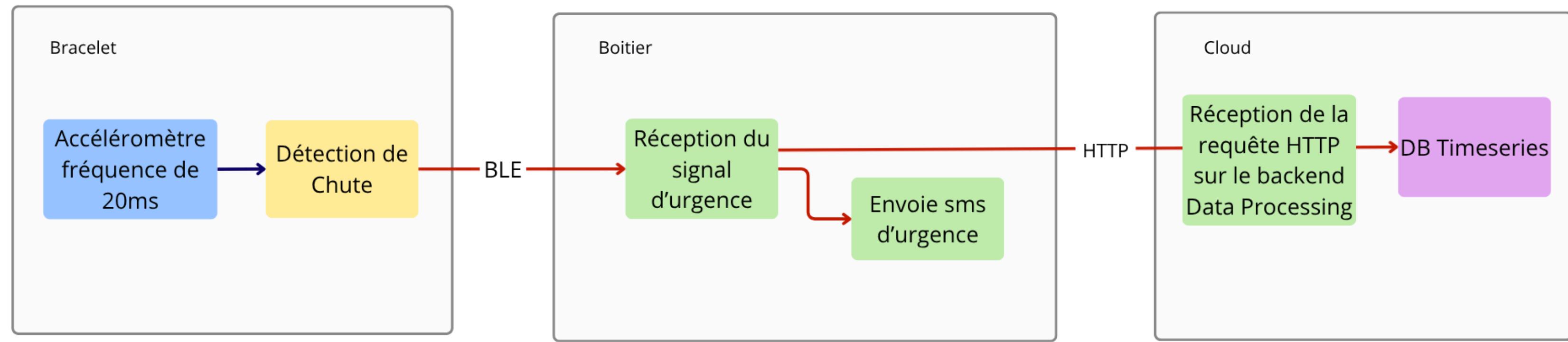
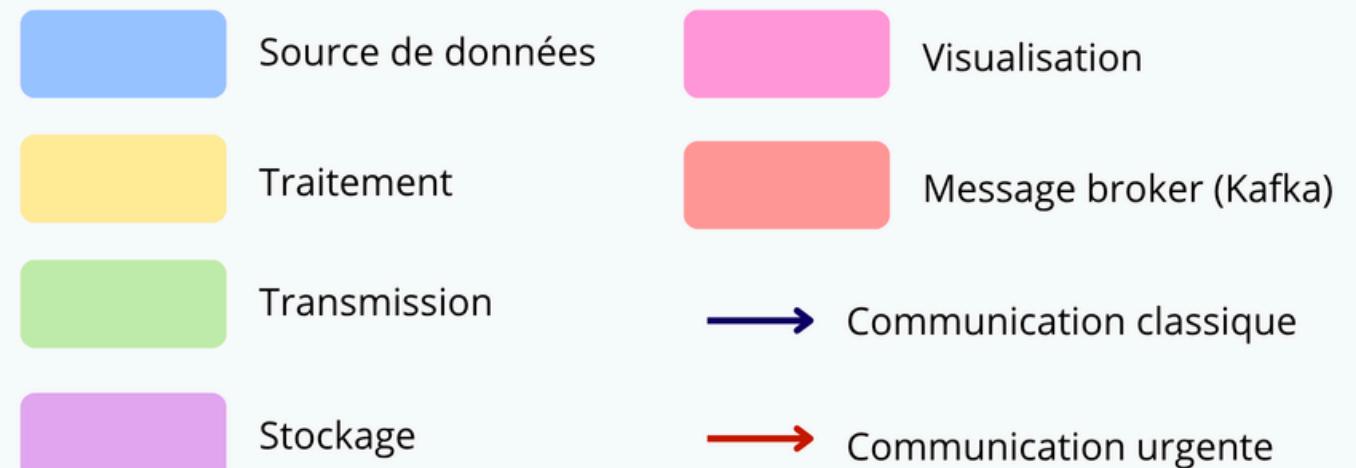
Légende des diagrammes de data pipelines



Flux de données

Chutes

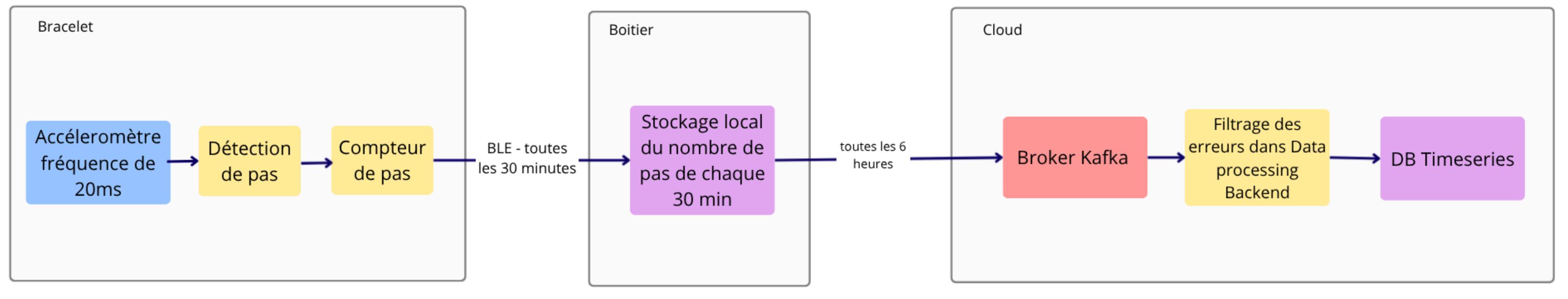
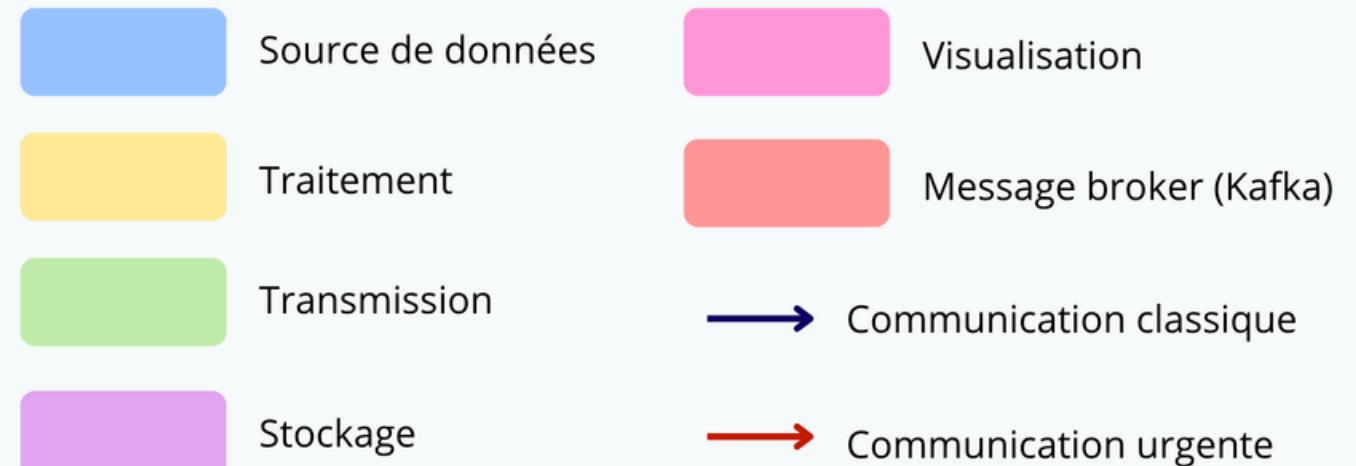
Légende des diagrammes de data pipelines

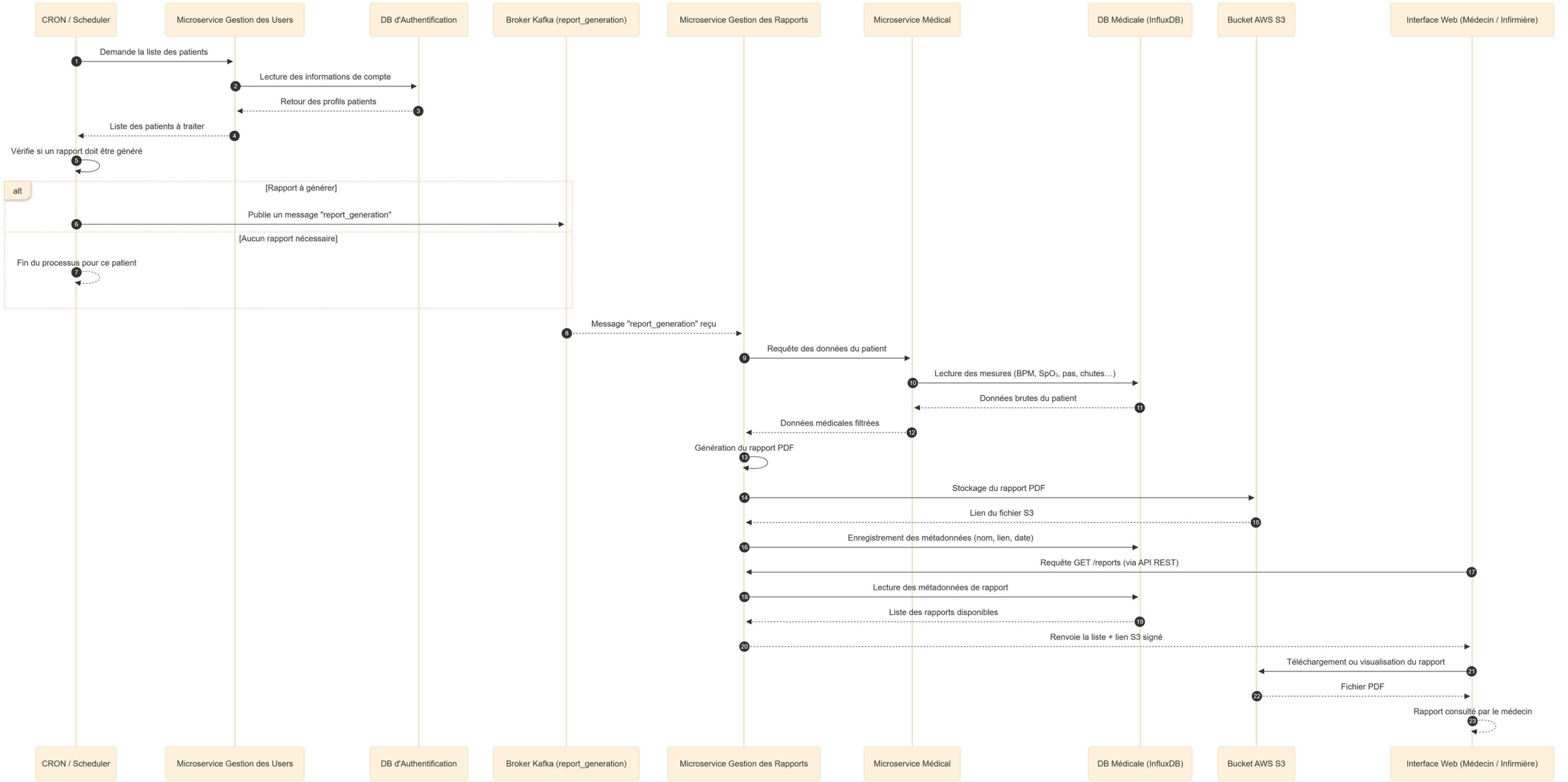


Flux de données

Nombre de pas

Légende des diagrammes de data pipelines





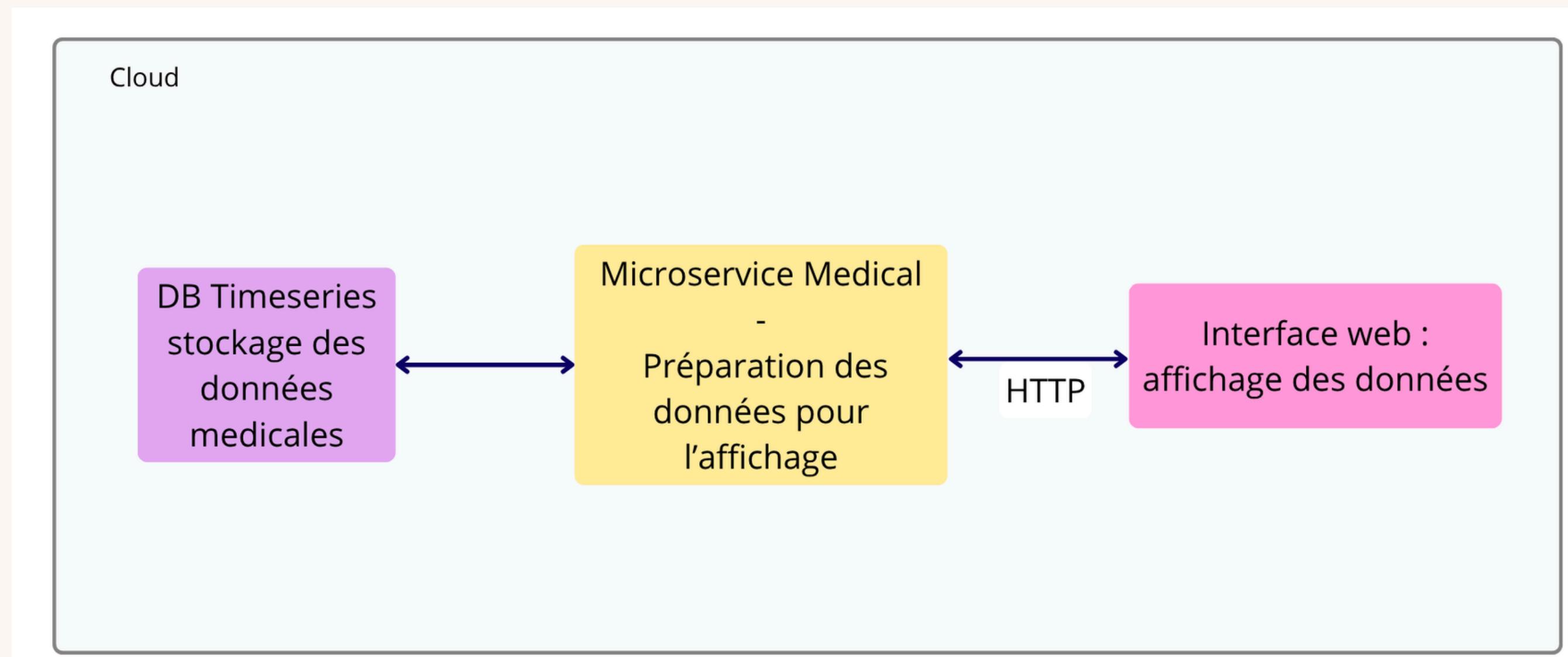
Flux de données

Exploitation des données

Légende des diagrammes de data pipelines

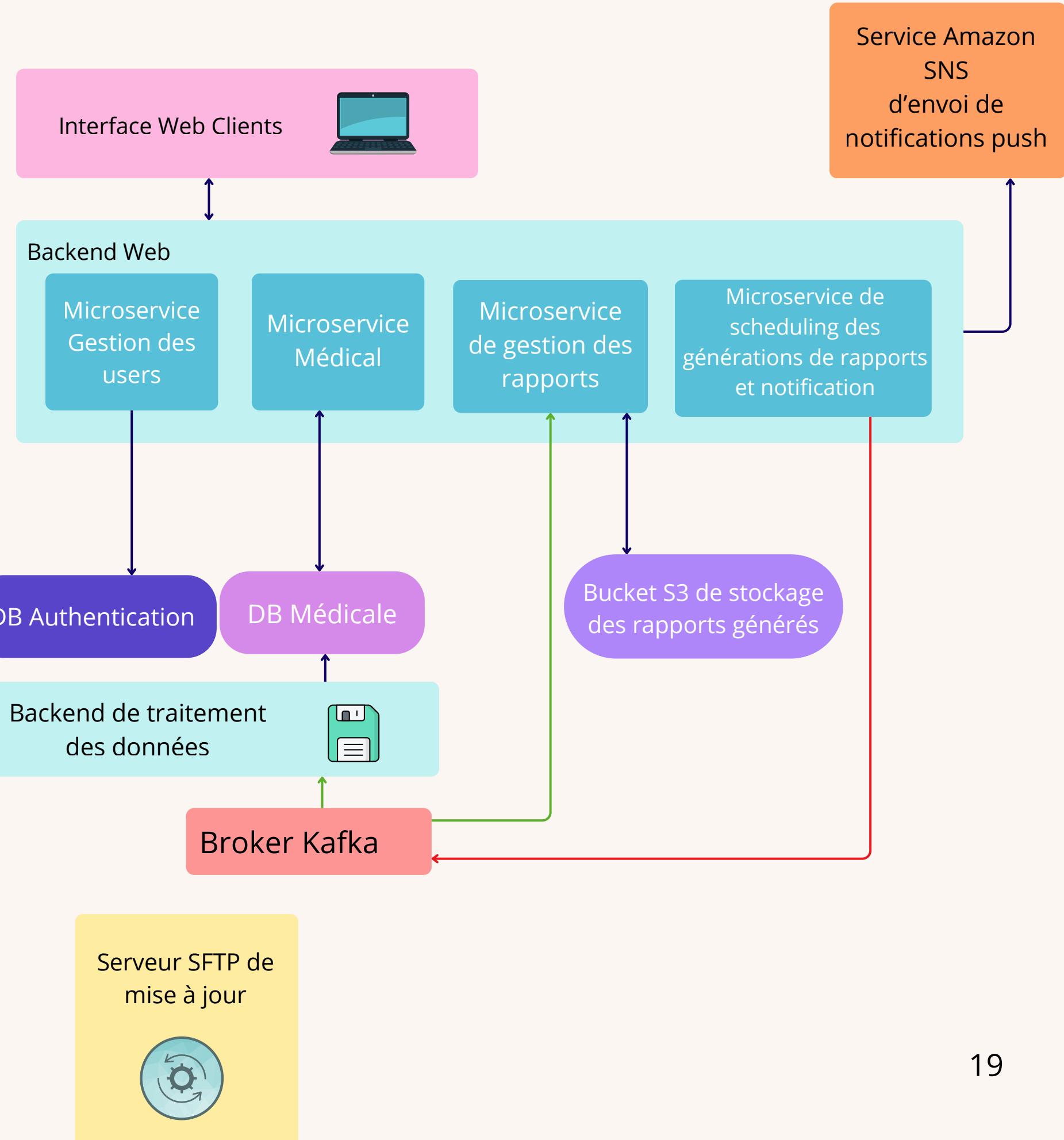
Source de données	Visualisation
Traitements	Message broker (Kafka)
Transmission	Communication classique
Stockage	Communication urgente

Une fois les données traitées et stockées en DB, pour les visualiser dans le frontend :



Service Cloud choisi :

DB Authentification (RDS t4g.medium)	59.21 \$
Interface Web (S3 Static Hosting)	4.25 \$
Backend Web (EC2 t4g.xlarge)	73.59 \$
DB Médicale (Timestream)	4505.59 \$
Data Processing (ECS Fargate)	66.33 \$
Broker Kafka (MSK t3.small ×3)	148.02 \$
Bucket S3 (Gestion des rapports)	34.59 \$
Serveur SFTP (Amazon Transfer Family)	257.20 \$
Service de Notifications (SNS)	81.21 \$
Coût total :	5230 \$/mois



Scénario du PoC

Le bracelet :

- se connecte en WebSocket au boitier
- génère un BPM toutes les secondes, envoie une moyenne toutes les 10 secondes
- Toutes les 40 secondes, génère des BPM de tachycardie au lieu d'un BPM stable
- A l'envoie de la moyenne, si moyenne très élevée, envoie un signal de Tachycardie

Le Boitier :

- Envoie un message au Broker Kafka pour chaque moyenne reçue
- Signal de crise → Requête Synchrone directement au Cloud

Le Backend “Data Processing” :

- Traite chaque message Kafka pour ajouter un point dans *InfluxDB*
- Ajoute un point d'alerte pour chaque requête HTTP d'alerte reçue

Web Interface Backend :

- Récupère les données de la DB pour l'affichage

Démonstration du POC

Merci
pour
votre
écoute !

