

# Système de monitoring en temps réel de la pression artérielle

*Nouran GHALIOUNJI, Houda EL-BAIDAK, Reina BALTAGI*

# Contexte

La surveillance continue des paramètres vitaux constitue un enjeu majeur milieu médical.

Les systèmes temps réel permettent d'identifier rapidement les situations à risque et d'améliorer la prise en charge des patients.

Les architectures de streaming se révèlent particulièrement adaptées pour répondre à ce type de problématique.

# Objectifs du projet

- Simuler des mesures de pression artérielle en temps réel
- Transmettre ces données via une architecture distribuée
- Analyser et visualiser les données



# Logique et Flow du projet

Génération  
des données  
médicales

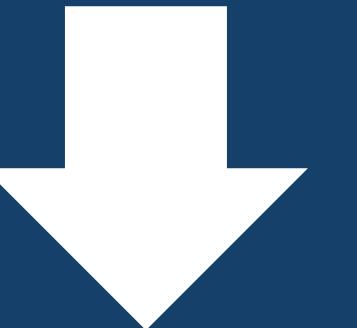
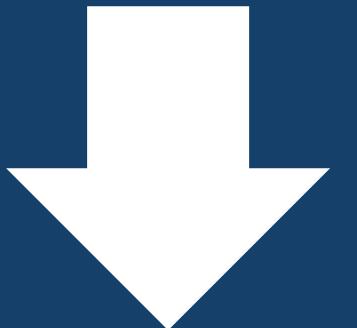
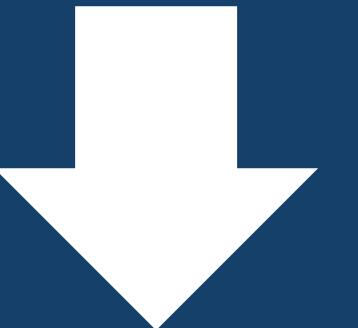
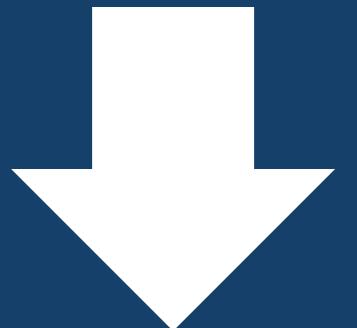
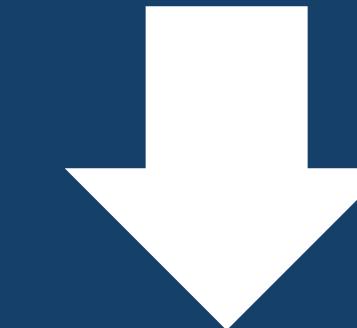
Streaming  
temps réel

Consommation  
et analyse

Détection des  
anomalies

Traitement  
des résultats

Visualisation



Observations  
de pression  
artérielle au  
format FHIR  
(JSON)

Envoi des  
observations  
via Kafka  
Producer vers  
un topic

Kafka Consumer  
récupère les  
messages  
et extrait des  
valeurs systolique  
et diastolique

Analyse basée  
sur des seuils  
médicaux  
&  
Classification :  
normal /  
anormal

Cas normaux :  
stockage local  
(JSON)  
Cas anormaux :  
indexation  
dans  
Elasticsearch

Analyse et suivi  
en temps réel  
via Kibana

# Outils et technologies

Génération et format des données



Python  
FHIR (HL7) - Standard Médical via  
fhir.resources sur Python

Streaming & traitement temps réel



Apache Kafka : Kafka Producer &  
Kafka Consumer via Docker et  
kafka-python

Stockage



Elasticsearch pour les anomalies &  
Stockage local en JSON pour les cas  
normaux

Visualisation & monitoring



Kibana

Orchestration



Docker : encapsule Kafka, Elasticsearch,  
Kibana & Docker Compose : orchestration  
des services pour faciliter le déploiement

# Architecture globale du système

Générateur FHIR  
JSON  
 



Kafka Producer  




Topic Kafka  
  
Boîte aux lettres dans Kafka, où  
le producer dépose les  
messages



Kafka Consumer  
Lecture FHIR  
Analyse de la pression  
artérielle  
Gestion des résultats  




 kibana  
Kibana  
visualisation  
des données

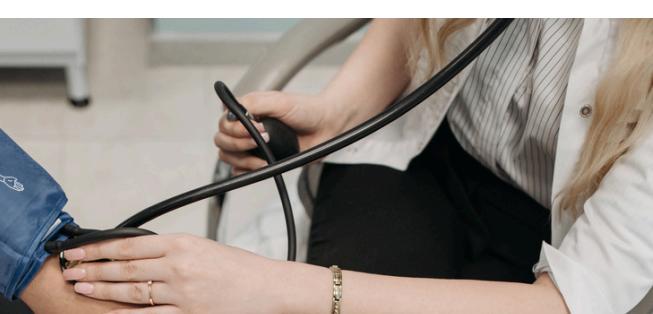


 elastic  
ElasticSearch  
(Stockage  
Anomalies)  
Stockage  
Local en  
Json  
(Cas  
Normaux)

# Ingestion des données

# Génération des données médicales (FHIR Generator)

- Génération d'observations de pression artérielle
- Format standard FHIR – Observation
- Données structurées en JSON
- Utilisation des codes médicaux LOINC
- Timestamp en temps réel



# Structure d'une Observation FHIR (pression artérielle)

Identifiant du patient

Pression diastolique

Unité : mmHg

Date et heure de la mesure

Pression systolique

# Envoi des données en temps réel avec Kafka Producer

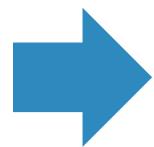
Kafka Producer  
écrit les  
données dans  
un topic Kafka

Envoi en  
temps réel

Un message =  
une observation  
FHIR

5 patients  
suivis  
simultanément

Simulation  
d'une évolution  
progressive



**Le Kafka Producer agit comme un capteur médical : il envoie régulièrement les nouvelles mesures dans Kafka**

# Analyse et Kafka Consumer

# Le Kafka Consumer

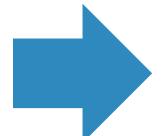
Réception des observations médicales depuis Kafka

Extraction des valeurs de pression artérielle (FHIR)

Analyse médicale (règles cliniques)

Décision : normal vs anomalie

Stockage adapté selon le niveau de risque



**Objectif : détecter en temps réel les patients à risque**

# Les règles d'analyse du diastolic et systolic

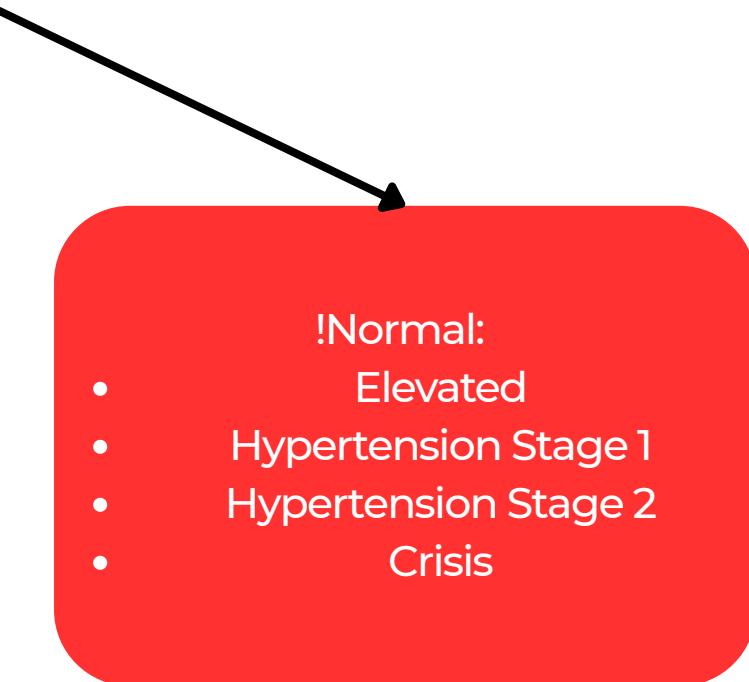
Catégorie	Condition
Normal	Systolic < 120 <b>AND</b> Diastolic < 80
Elevated	120 ≤ Systolic ≤ 129 <b>AND</b> Diastolic < 80
Hypertension – Stage 1	130 ≤ Systolic ≤ 139 <b>OR</b> 80 ≤ Diastolic ≤ 89
Hypertension – Stage 2	Systolic ≥ 140 <b>OR</b> Diastolic ≥ 90
Hypertensive crisis	Systolic > 180 <b>OR</b> Diastolic > 120

Diastolic: Pression du sang quand le cœur pompe le sang dans le corps

Systolic: Pression du sang quand le cœur se repose entre deux battements

# Stockage

Décision basée sur le niveau de risque  
Toutes les mesures ne sont pas traitées pareil :



obs_20260130_132238_522566	30/01/2026 14:22
obs_20260130_132604_589341	30/01/2026 14:26
obs_20260130_132604_606890	30/01/2026 14:26
obs_20260130_132609_590240	30/01/2026 14:26
obs_20260130_132609_729994	30/01/2026 14:26
obs_20260130_132609_731654	30/01/2026 14:26
obs_20260130_132614_617794	30/01/2026 14:26
obs_20260130_132614_779579	30/01/2026 14:26
obs_20260130_132619_645972	30/01/2026 14:26
obs_20260130_132619_894386	30/01/2026 14:26
obs_20260130_132629_713765	30/01/2026 14:26
obs_20260130_132645_049551	30/01/2026 14:26
obs_20260130_132649_940623	30/01/2026 14:26

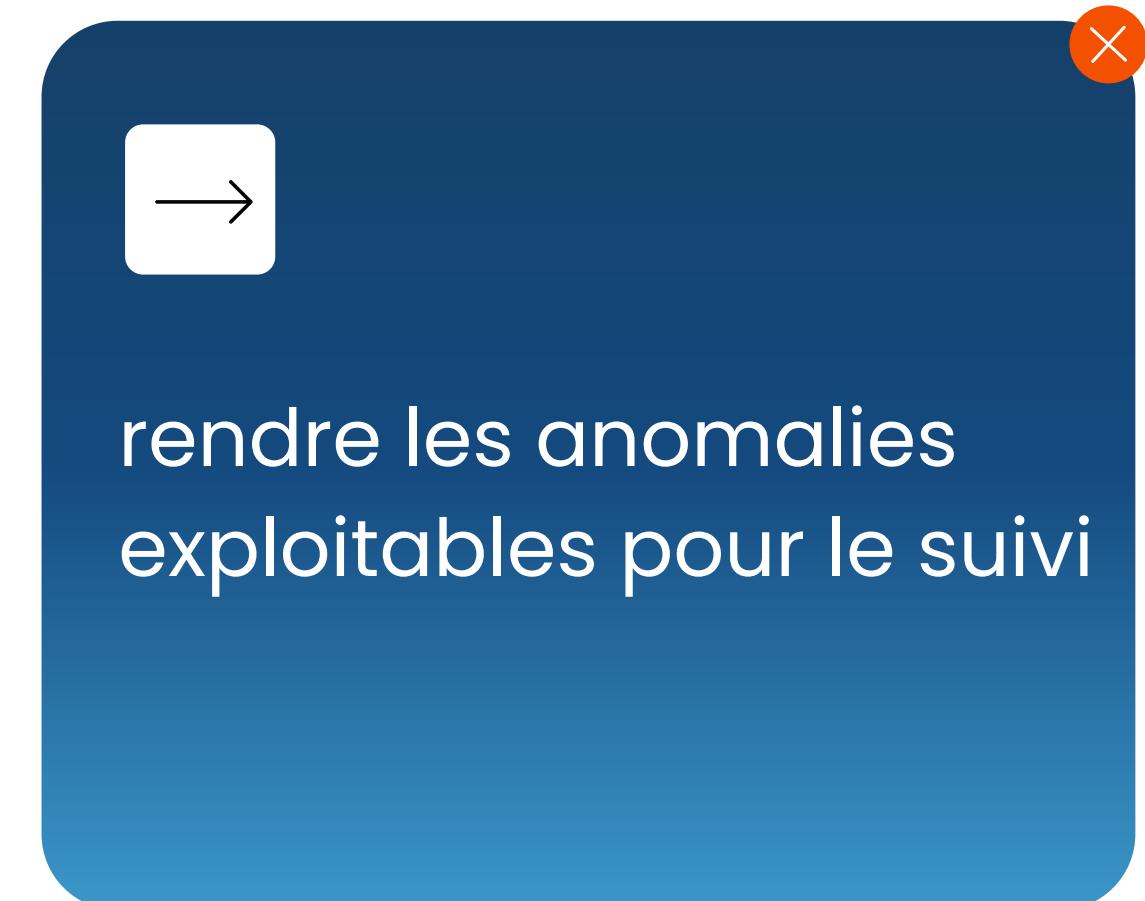
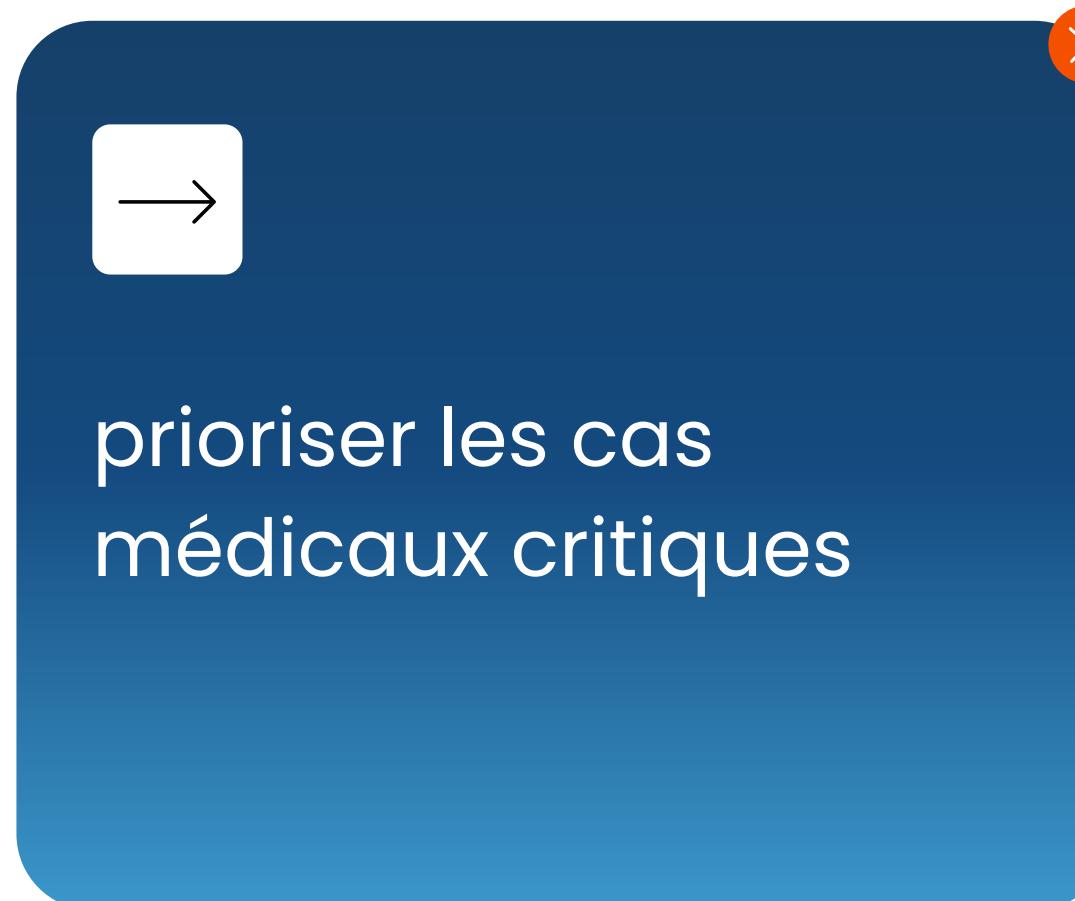
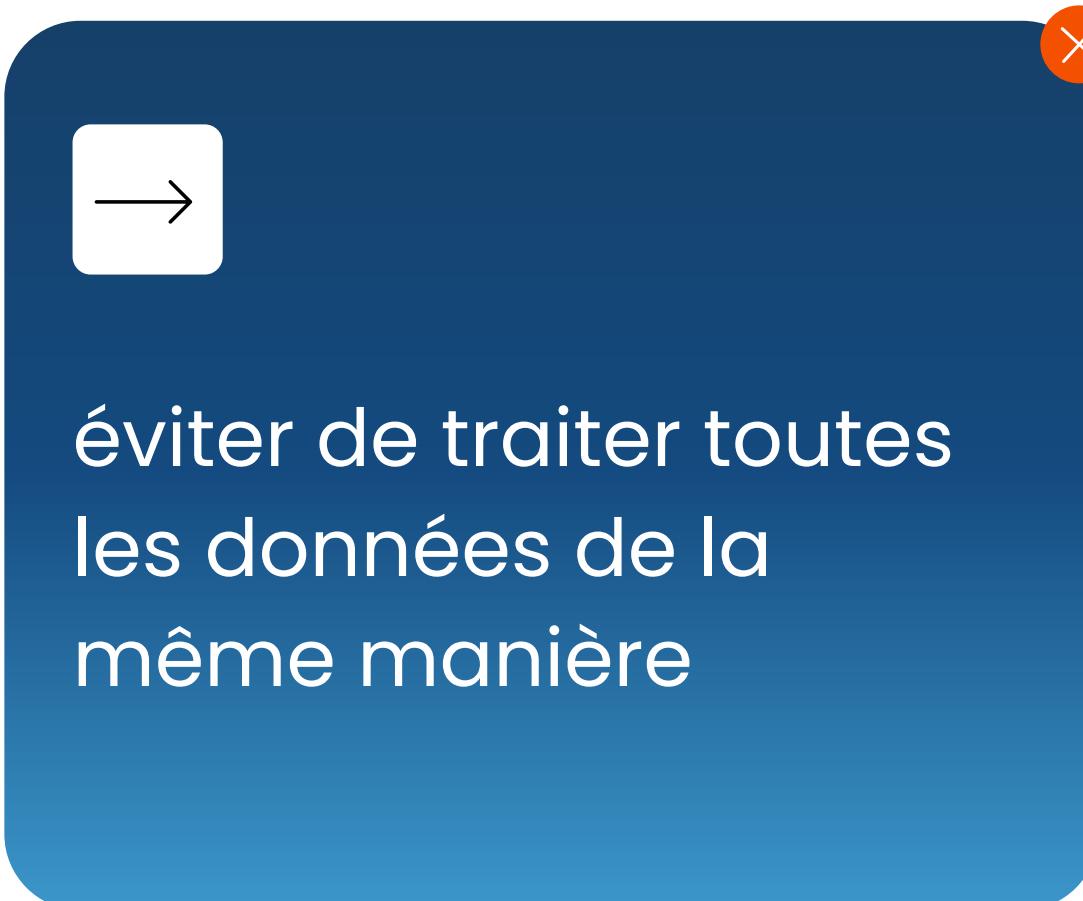
```
(venv) PS C:\Users\noura\blood-pressure-project> python -m src.kafka_consumer
⚠ ANOMALIE ['hypertension_stage_1'] | patient=Patient/PAT-002 | sys=81.0 dia=80.0 | t=2026-01-30T13:33:27.052001Z
📦 Donnée indexée dans Elasticsearch
⚠ ANOMALIE ['hypertension_stage_2'] | patient=Patient/PAT-003 | sys=86.0 dia=119.0 | t=2026-01-30T13:33:27.059116Z
📦 Donnée indexée dans Elasticsearch
⚠ ANOMALIE ['hypertension_stage_2'] | patient=Patient/PAT-004 | sys=107.0 dia=119.0 | t=2026-01-30T13:33:27.065057Z
📦 Donnée indexée dans Elasticsearch
⚠ ANOMALIE ['hypertension_stage_2'] | patient=Patient/PAT-005 | sys=154.0 dia=91.0 | t=2026-01-30T13:33:27.070791Z
📦 Donnée indexée dans Elasticsearch
✅ NORMAL | patient=Patient/PAT-001 | sys=88.0 dia=56.0 | t=2026-01-30T13:33:32.083723Z
💾 Sauvegardé en local : data\normal\obs_20260130_150434_344307.json
⚠ ANOMALIE ['hypertension_stage_2'] | patient=Patient/PAT-002 | sys=177.0 dia=58.0 | t=2026-01-30T13:33:32.093359Z
```



# Gestion et valorisation des données après analyse

# Objectif :

Après la détection des anomalies par le consumer Kafka, mon module décide comment et où stocker les données.



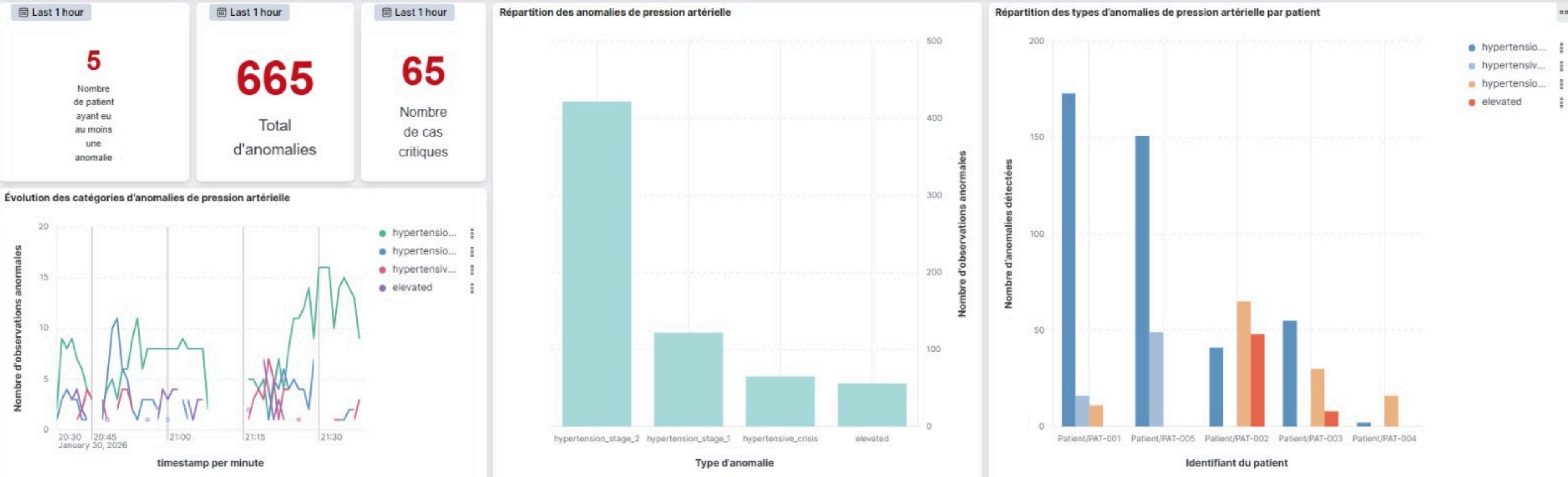
# Logique de stockage intelligente :

Le système applique une logique simple mais essentielle :

Type de mesure	Stockage	Finalité
Normale	Fichier JSON local	Historique
Anormale	Elasticsearch	Surveillance & analyse

# Tableaux de bord sous Kibana :

## Système de surveillance des anomalies de pression artérielle



# Les limites du projet

## Génération de données simplifiée

Les données sont générées de manière aléatoire. Une approche plus réaliste aurait consisté à utiliser des distributions statistiques issues de données médicales réelles.

## Analyse basée uniquement sur des règles

L'analyse repose exclusivement sur des seuils médicaux fixes. Bien que pertinents, ces seuils ne prennent pas en compte la variabilité individuelle des patients ni l'évolution temporelle des mesures.

## Absence de modèles de Machine Learning

Le projet ne comprend pas de modèles de Machine Learning pour prédire ou anticiper le risque patient. Des modèles supervisés auraient pu être ajoutés afin de prédire la catégorie de pression artérielle ou d'estimer un score de risque.

# Merci !