Système de Surveillance Cardiaque Connecté

Une Architecture Hybride pour la Télémédecine Intelligence

Azzeddine BELILITA & Youssef DEROUICHE | M2 Infrastructures Cloud & Systèmes Distribués (SYRIUS)

Plan de la Présentation

1 **Le Constat**
Les limites de la surveillance traditionnelle

2 **Le Défi Technique**
Concilier données et autonomie

3 **Notre Vision**
L'intelligence au plus près du patient

4 **L'Architecture**
Une chaîne de traitement de bout en bout

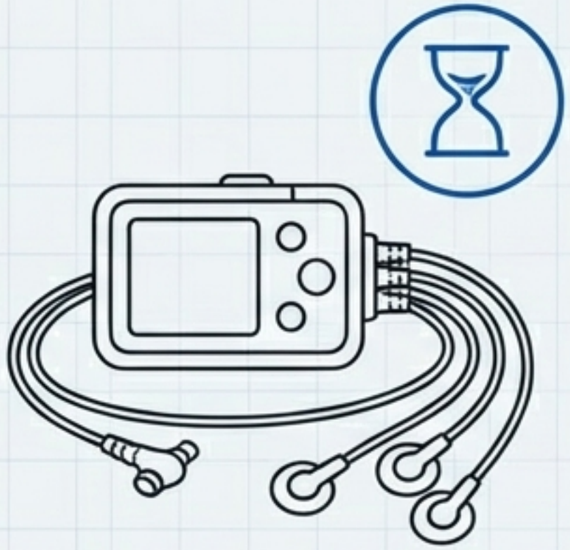
5 **Plongée Technique**
Au cœur du dispositif et du cloud

6 **La Démonstration**
De l'alerte à l'analyse clinique

7 **Conclusion & Perspectives**

L’Angle Mort du Holter Traditionnel

Le Holter traditionnel présente une limitation majeure : l'absence de temps réel. Les données sont stockées sur une carte mémoire et ne sont analysées par le médecin qu'une fois l'appareil rendu. Si une anomalie grave survient, le médecin n'en est pas informé immédiatement.

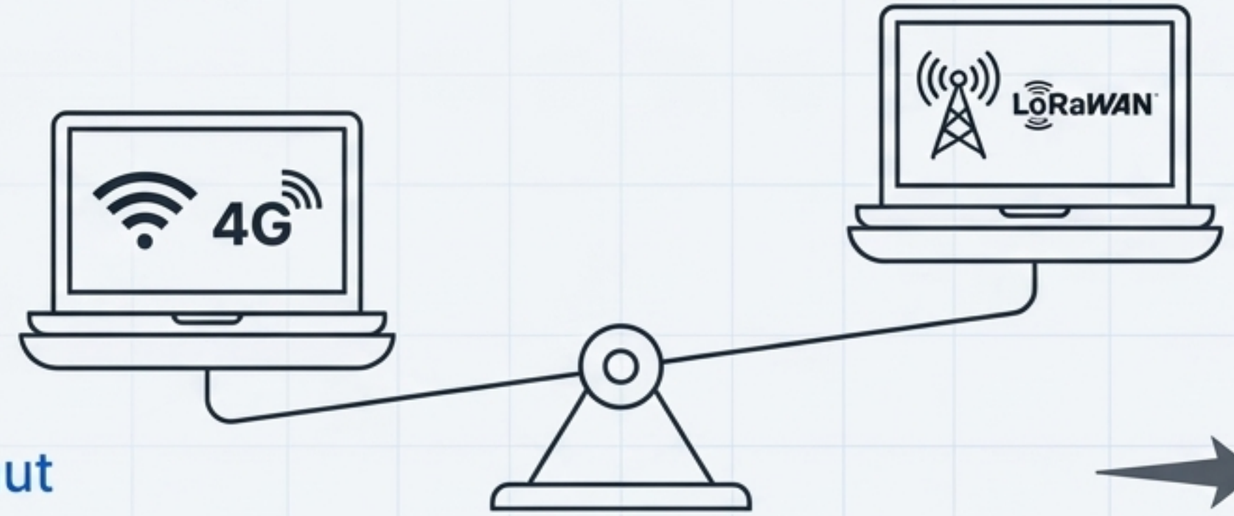


Diagnostic en différé

Besoin : Alerte immédiate

Le Dilemme Technique : Concilier Données et Autonomie

Transmettre l'intégralité d'un signal ECG haute fréquence nécessite une bande passante importante (Wi-Fi, 4G), ce qui est très énergivore. À l'inverse, les réseaux basse consommation comme LoRaWAN garantissent une grande autonomie, mais leur bande passante est trop faible.



Débit Très Haut

Débit Très Faible

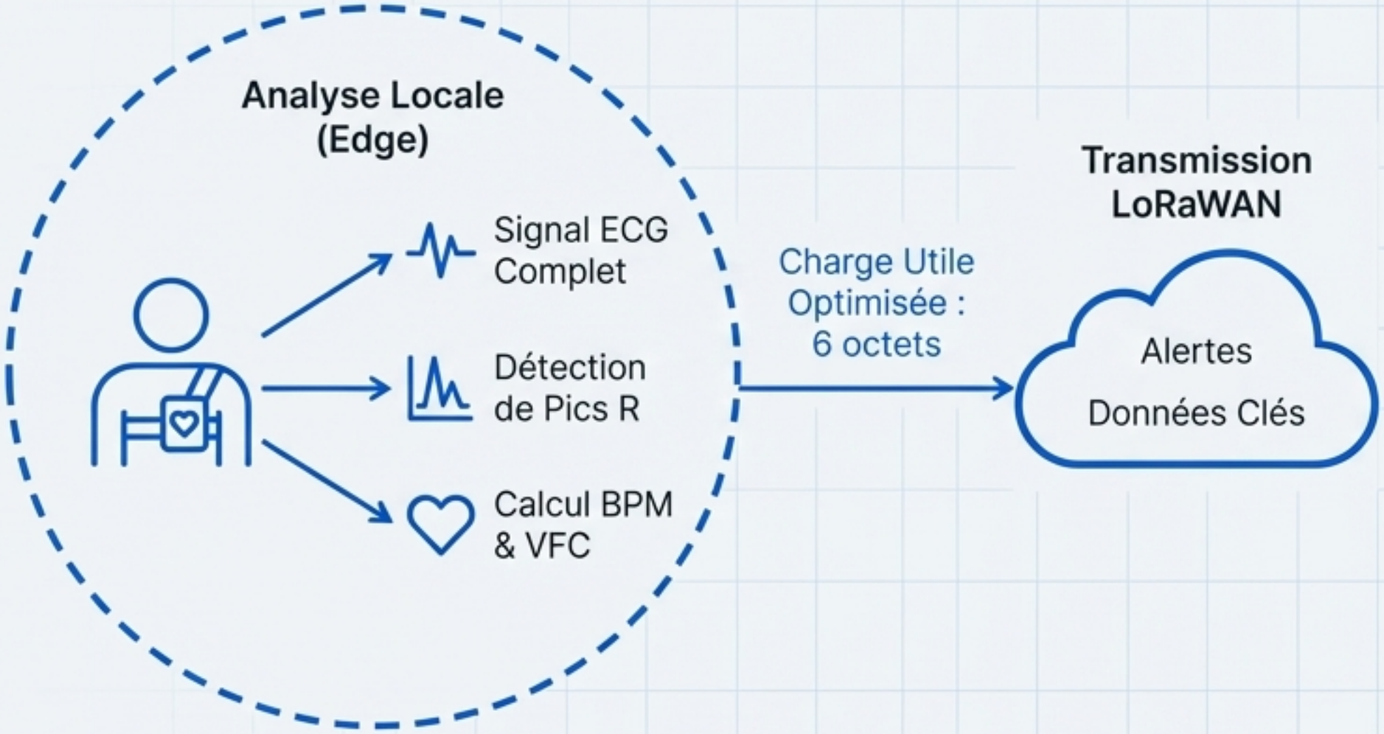
Autonomie Faible

Comment obtenir le meilleur des deux mondes ?

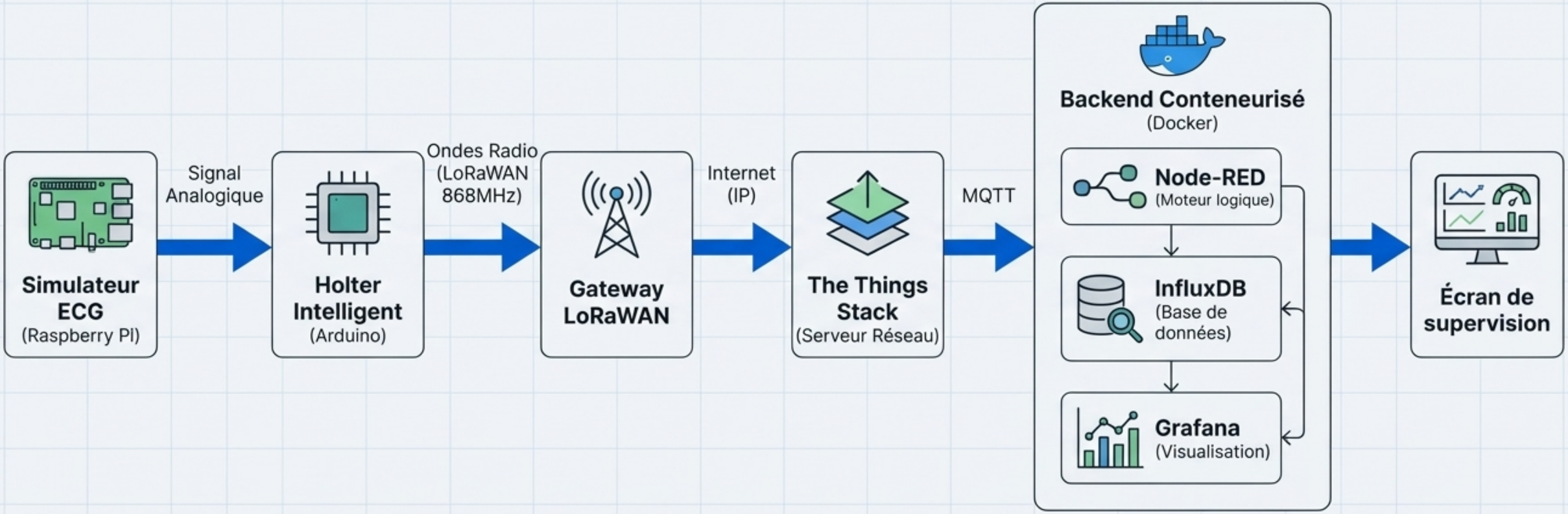
Autonomie Excellente

Notre Vision : L'Intelligence en Périphérie (Edge Computing)

Plutôt que de transmettre le signal brut, notre dispositif analyse les données à la source. Il n'envoie que les indicateurs clés et les alertes critiques, préservant ainsi l'autonomie tout en assurant une surveillance efficace.

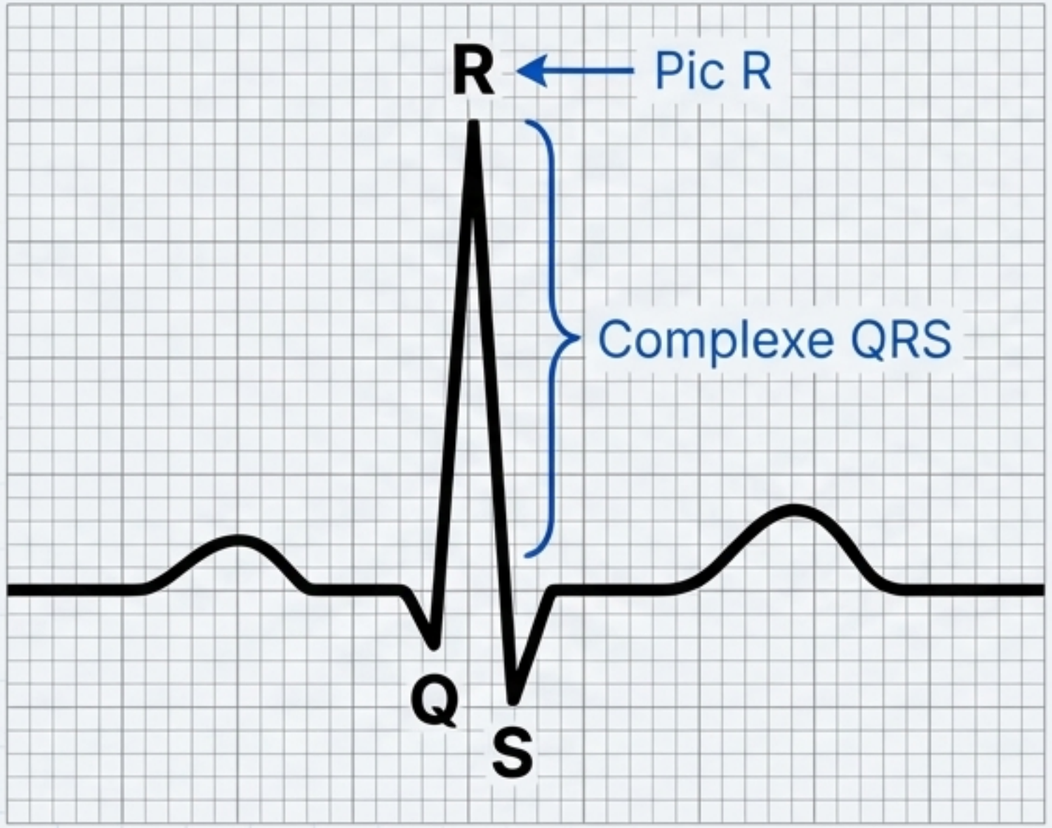


Architecture Complète du Système : du Capteur au Clinicien



Au Cœur du Dispositif : Le Traitement du Signal Embarqué

Le firmware C++ implémenté sur l'Arduino agit comme un filtre intelligent. Il ne se contente pas de relayer l'information, il l'analyse en **temps réel** pour transformer une onde complexe en **métriques vitales**.



1. **Acquisition** : Lecture du signal analogique brut (0-1023).
2. **Détection de Pic** : Algorithme de seuillage et d'hystérésis pour identifier chaque battement (Pic R).
3. **Calcul BPM** : Comptage des pics sur une fenêtre de 10s et extrapolation en Battements Par Minute.
4. **Calcul VFC (HRV)** : Mesure de l'intervalle R-R en microsecondes pour évaluer la variabilité.

Traitement du signal embarqué sur l'Arduino

Logique de détection

1. Acquisition:
value = analogRead(A0)

→

2. Filtre (Seuil):
si value > 240
=> pic potentiel

↓

3. Verrou (Flag):
évite le double comptage d'un même pic.

Calcul du Rythme (BPM)

Extrapolation sur une fenêtre courte de 10s pour une meilleure réactivité :

$$\mathrm{HR} = \mathrm{Count10s} \times 6$$

où Count10s est le nombre de pics R détectés sur 10s.

Calcul de la Variabilité (HRV)

Comparaison théorique vs réelle :

$$\mathrm{HRV} = \frac{\mathrm{HR}}{60} \text{ [Fréquence théorique (Hz)]} - \frac{\mathrm{Intervalle_}\mu\mathrm{s}}{10^6} \text{ [Durée réelle entre deux pics (s)]}$$

Une Communication Optimisée : La Charge Utile LoRaWAN

Pour maximiser l'autonomie et respecter les contraintes du réseau, nous n'envoyons pas de texte, mais une **charge utile binaire** de 6 octets. Un script de décodage côté serveur la transforme en données JSON exploitables.

Payload Binaire

```
Payload: 00 66 EE 48 01 9B
```

Payload Formatter

Script de Décodage (Serveur)

```
1   function decodeUplink(input) {
2     if (input.bytes.length !== 6) {
3       return { errors: ["Payload length must be 6 bytes"] };
4     }
5
6     // Décodage des 2 octets pour Heart Rate
7     const heartRate = (input.bytes[0] << 8) | input.bytes[1];
8     // Décodage des 2 octets pour HRV raw
9     const hrv_raw = (input.bytes[2] << 8) | input.bytes[3];
11    // Décodage des 2 octets pour ECG value
12    const ecg = (input.bytes[4] << 8) | input.bytes[5];
13
14    // Correction et Conversion
15    const hrv_ms = hrv_raw / 1000; // Convertir µs → ms
16
17    return {
18      data: {
19        heart_rate: heartRate,
20        hrv_ms: hrv_ms,
21        ecg_value: ecg
22      }
23    };
24  }
```

Résultat JSON

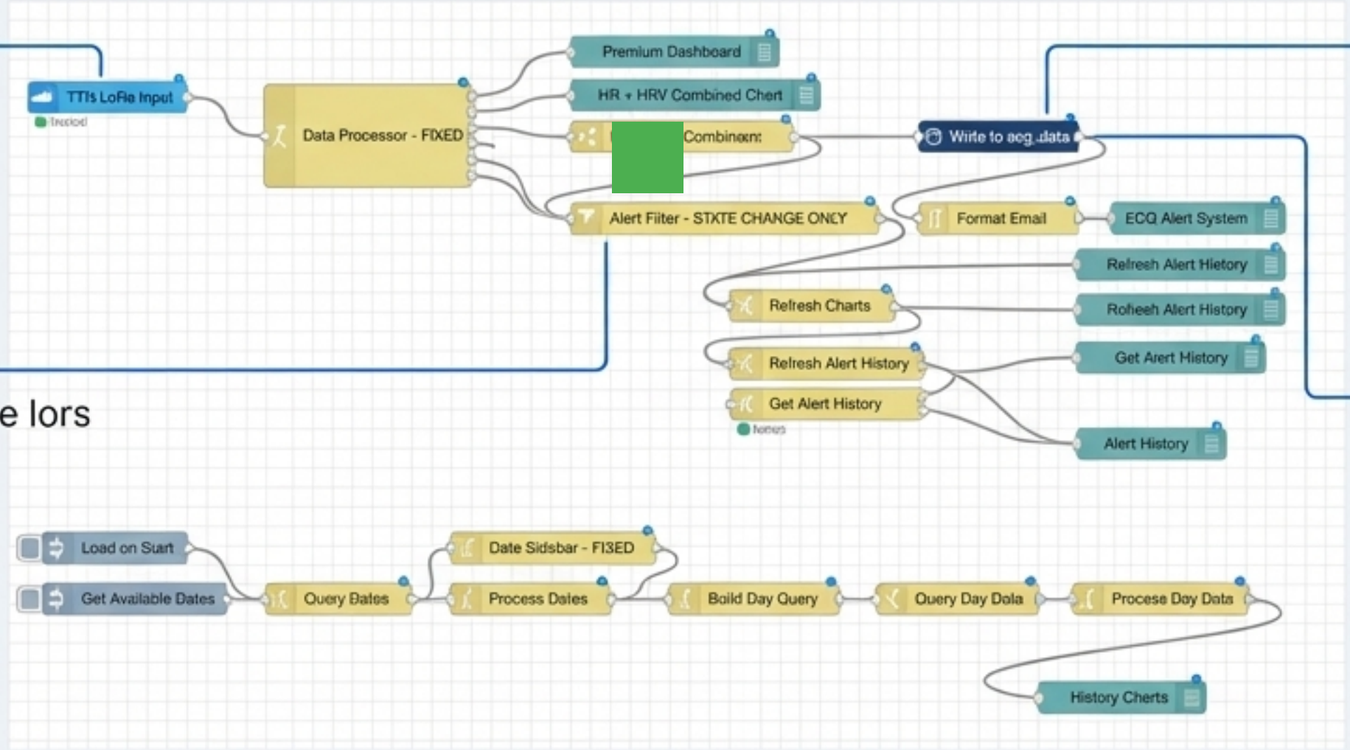
```
{
  "heart_rate": 102,
  "hrv_ms": 61,
  "ecg_value": 411
}
```

Le Moteur Logique : Orchestration du Flux de Données avec Node-RED

Le flux Node-RED est le cerveau analytique du système. Il reçoit les données via MQTT, exécute la logique de classification **clinique**, **filtre** les alertes pour éviter le spam, et persiste chaque mesure dans InfluxDB.

1. Réception (TTN LoRa Input): S'abonne au topic MQTT et ingère les données.

3. Filtrage Intelligent (Alert Filter): N'envoie une alerte que lors d'un changement d'état pathologique.



2. Traitement (Data Processor): Classifie l'état clinique (Normal, Tachycardie, etc.) et lisse les données.

4. Stockage (Write to ecg_data): Écrit les données formatées dans la base InfluxDB.

La Boucle est Bouclée : De l'Alerte à l'Analyse Post-Crise

Ce système offre une surveillance complète, du temps réel à l'analyse a posteriori.

1. Détection en Temps Réel

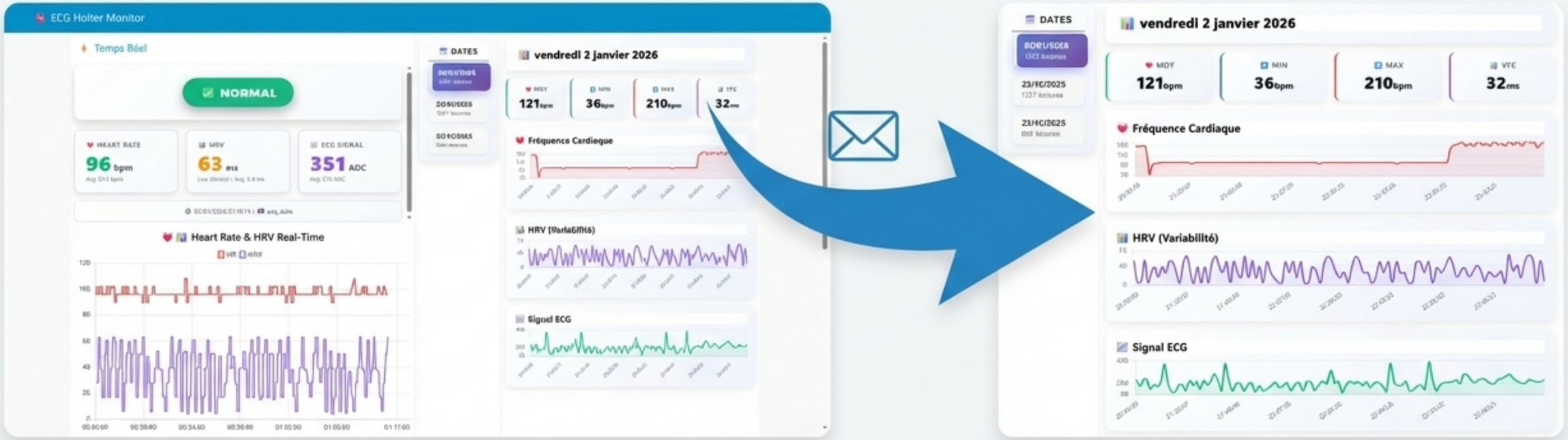
Le tableau de bord principal affiche l'état actuel du patient. Le statut '**NORMAL**' indique que tout va bien.

2. Alerte Instantanée

Si une anomalie survient, le système d'alerte notifie immédiatement le personnel médical par e-mail.

3. Analyse Historique (Replay)

Le médecin peut alors se connecter, sélectionner le jour de l'incident et rejouer la séquence complète pour analyser le signal ECG, la fréquence cardiaque et la variabilité qui ont précédé et suivi l'événement critique.



Au-delà du Temps Réel : Le Mode « Replay » pour l'Analyse Historique

L'intégralité des données est stockée dans InfluxDB. L'interface permet de sélectionner n'importe quelle journée d'enregistrement passée pour rejouer la séquence, analyser les tendances et investiguer les anomalies en détail.

La barre latérale « DATES » : Montrant la facilité de navigation entre les jours d'enregistrement.

Les indicateurs agrégés (MOY, MIN, MAX) : Statistiques clés pour la journée sélectionnée.

Les graphiques (Fréquence Cardiaque, HRV, Signal ECG) : Visualisation complète des données sur 24h.

L'Infrastructure Cloud : Un Backend Micro-Services Conteneurisé

L'ensemble de l'infrastructure de supervision est défini "as code" et orchestré par Docker Compose. Cette approche garantit une **isolation des services**, une reproductibilité parfaite et une **portabilité multi-plateforme** (serveur local, cloud public, etc.).

InfluxDB (Le Stockage Temporel)

Base de données optimisée pour l'écriture massive de données médicales.

Node-RED (Le Moteur Logique)

Orchestrateur de flux qui traite, classifie et gère les alertes en temps réel.

Grafana (L'Interface de Supervision)

Front-end de visualisation temps réel et d'analyse historique.

Le Tableau de Bord Clinique : Une Vue à 360° en Temps Réel

L'interface Grafana offre une lecture immédiate et synthétique des signes vitaux. Les données sont agrégées et visualisées en moins d'une seconde, permettant au personnel médical de prendre des décisions éclairées rapidement.

Jauges BPM & HRV :
Indicateurs critiques avec code couleur dynamique.

Graphique Trend History :
Suivi de l'évolution du rythme cardiaque sur les dernières minutes.

Graphique Realtime ECG :
Validation visuelle du signal brut.

Tableau Alert History :
Journal des anomalies détectées.

L'Alerte Intelligente : Une Notification Ciblée au Moment Critique

En cas d'anomalie, le système génère et envoie un courriel d'alerte riche au personnel soignant. La notification contient toutes les informations nécessaires pour une première évaluation de l'urgence.

Titre (« Tachycardia ») :
Classification claire et immédiate de l'anomalie.

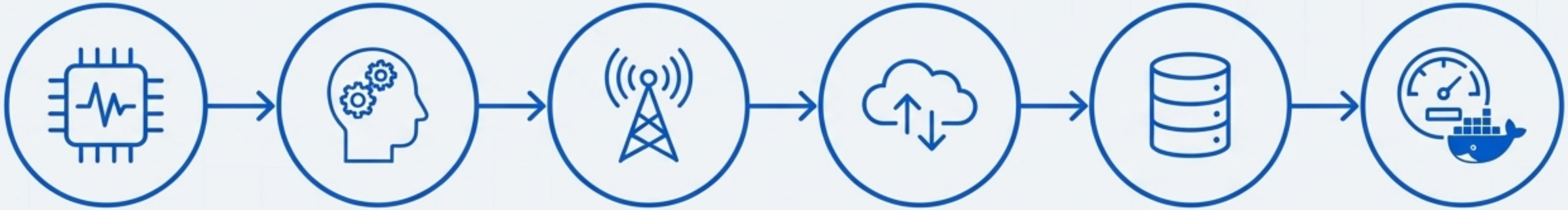
Code Couleur (Jaune/Orange) :
Indication visuelle du niveau de sévérité.

Section « Vital Signs » :
Snapshot des signes vitaux au moment précis de l'événement.

Section « Alert Details » :
Horodatage et identifiants pour une traçabilité complète.

Synthèse des Réalisations Techniques

Le projet a couvert l'ensemble de la chaîne de valeur, de l'acquisition du signal biomédical à la visualisation des données sur une interface professionnelle.



1. Simulation & Acquisition

Acquisition d'un signal ECG brut stable

2. Edge Computing

Transformation de données brutes en métadonnées légères

3. Connectivité LoRaWAN

Transmission longue portée basse consommation

4. Backend & Logique

Réception temps réel et tri intelligent des alertes

5. Stockage

Historisation persistante pour la relecture

6. Visualisation & DevOps

Interface professionnelle déployable en une commande

Conclusion et Perspectives

Ce projet démontre la faisabilité d'un Holter ECG connecté, intelligent et économe en énergie en utilisant des technologies ouvertes et standardisées.

Réalisations Clés :

- **Architecture Hybride Validée** : L'approche combinant Edge Computing et LoRaWAN résout efficacement le dilemme entre volume de données et autonomie.
- **Chaîne de Traitement Complète** : Le prototype fonctionnel couvre l'ensemble du flux, du capteur à l'interface clinique.
- **Infrastructure Déployable** : L'utilisation de Docker garantit la reproductibilité et la robustesse du système backend.

Prochaines Étapes Possibles :

- → Miniaturisation du matériel pour un port plus confortable.
- → Intégration d'algorithmes de Machine Learning pour une détection d'anomalies plus fine.
- → Essais cliniques pour validation en conditions réelles.

Auteurs du projet :
Azeddine BELILITA & Youssef DEROUICHE
M2 Infrastructures Cloud & Systèmes Distribués