IOT

PROJET

Contrôle environnemental pour station spatiale (ECLSS)

Livrable 4

Remy MOREEL

Léo SCHAEFFER

Axel DELAR

Mohammed EL ORFALI

CESI
ÉCOLE D'INGÉNIEURS

Table des matières

1. Présentation de l'équipe

Voici notre équipe pour la réalisation de ce projet :

Mohammed EL ORFALI
Chef de projet

Léo SCHAEFFER

Axel DELAR

Remy MOREEL

2. Dossier de compréhension et d’analyse

Le projet vise à fiabiliser la remontée de données critiques (température, pression, qualité de l'air) dans un environnement sujet à des instabilités. L'analyse du Livrable 3 a permis de valider un Proof of Concept (POC) fonctionnel reposant sur des objets contraints (Arduino R4 WiFi) et une passerelle Edge (Raspberry Pi 4) communiquant via le protocole MQTT.

La problématique évolue désormais de la simple "preuve de fonctionnement" vers la durabilité et la sécurisation industrielle. Il ne s'agit plus seulement de transporter la donnée, mais de garantir qu'elle est authentique (émise par le bon capteur), intègre (non modifiée en route) et traçable (enregistrée de manière immuable) conformément aux normes de cybersécurité IoT (ETSI EN 303 645).

2.1 Identification des Enjeux

- Enjeux Techniques et de Fiabilité :

Le choix du protocole MQTT en mode "Publish/Subscribe" permet de découpler les capteurs des applications, mais introduit un point de centralisation critique (le Broker Mosquitto).

La gestion de la QoS 1 (Quality of Service) dans le POC assure la livraison des messages, mais nécessite une gestion des doublons au niveau applicatif pour ne pas fausser les analyses.

L'architecture doit supporter la perte de connexion réseau (mode dégradé) sans perte de données critiques.

- Enjeux de Confiance (Sécurité & Traçabilité) :

Authentification des objets : Éviter qu'un tiers n'injecte de fausses mesures de température (spoofing) qui pourraient déclencher des alarmes injustifiées ou masquer un incident.

Intégrité des données : Garantir que la mesure affichée sur le Dashboard Node-RED est strictement identique à la valeur physique captée par le BME280.

Preuve d'existence : L'usage d'une Blockchain (en cours d'implémentation) répond au besoin de certifier les données pour des audits futurs (non-répudiation).

- Enjeux Énergétiques et Durables :

Sobriété des échanges : L'absence de messages "Retain" et l'usage de "Clean Sessions" (validés dans le Livrable 3) favorisent une empreinte mémoire réduite.

Architecture Edge : Le prétraitement sur Raspberry Pi (Gateway) évite d'envoyer la totalité des données brutes vers le Cloud, réduisant la consommation de bande passante et l'énergie de transport.

3. Proposition d'architecture IoT sécurisée et durable

Cette proposition formalise l'industrialisation du POC décrit dans le Livrable 3.

3.1 Architecture Fonctionnelle Stratifiée

L'architecture se décompose en trois couches, respectant le modèle Edge Computing :

Couche Physique (Perception) :

Équipements : Cartes Arduino R4 WiFi équipées de capteurs (BME280, Qualité Air).

Rôle : Acquisition de données et signature cryptographique locale.

Communication : Wi-Fi local vers la passerelle. Publication sur les topics eclss/#.

Couche Edge (Passerelle & Sécurité) :

- Équipement : Raspberry Pi 4.
- Services Hébergés :
    - Broker MQTT (Mosquitto) : Centralise les flux.
    - Node-RED : Orchestrateur des flux. Il reçoit les messages, vérifie la sécurité (HMAC), filtre les données aberrantes et gère la logique locale (ex: déclenchement buzzer).
    - Module de Sécurité (Secure Storage) : Gestion des clés via un fichier secrets.h sécurisé ou un coffre-fort numérique local.

Couche Service & Traçabilité (Cloud/Blockchain) :

- Supervision: Dashboard Node-RED accessible via HTTPS (4G/5G).
- Registre Distribué : Blockchain de Consortium (type Hyperledger Fabric ou Quorum avec consensus PBFT) pour l'ancrage des hachages de données validées.

3.2 Intégration des Mécanismes de Sécurité

Conformément à la norme ETSI EN 303 645 et à l'analyse du Livrable 3, voici les mécanismes implémentés :

A. Sécurisation du Transport (Confidentialité)

- TLS 1.3 : Tous les échanges MQTT entre l'Arduino et le Broker Mosquitto sont encapsulés dans un tunnel TLS.
    - *Implémentation :* Le Broker Mosquitto est configuré avec un certificat serveur. L'Arduino R4 utilise le certificat racine (CA) pour valider l'identité du Broker avant d'envoyer la moindre donnée (protection Man-In-The-Middle).

B. Intégrité et Authenticité Applicative (HMAC)

Au-delà du transport, nous garantissons que la donnée n'a pas été altérée par une signature applicative.

- Algorithme : HMAC-SHA256.
- Processus sur l'Arduino (Émetteur) :
    1. Construction du payload JSON : { "id": "sensor_1", "val": 24.5, "ts": 1706025600 }.
    2. Calcul de la signature : Sign = HMAC_SHA256(SecretKey, Payload + Timestamp).
    3. Publication MQTT : { "payload": {...}, "signature": "a1b2..." }.
- Processus sur Node-RED (Récepteur) :
    1. Réception du message.
    2. Recalcul de la signature avec la même SecretKey (stockée sécurisée côté Edge).
    3. Comparaison : Si Sign_Calculée!= Sign_Reçue, le message est rejeté et une alerte de sécurité est levée.

C. Protection contre le Rejeu (Anti-Replay)

- Utilisation du TimeStamp inclus dans le calcul HMAC. Node-RED vérifie que Heure_Actuelle - TimeStamp_Message < Delta_Max (ex: 5 secondes). Un message intercepté et renvoyé plus tard sera automatiquement rejeté.

3.3 Mécanisme de Traçabilité (Blockchain PBFT)

Pour répondre à l'exigence d'intégrité logicielle et de traçabilité des données mentionnée dans le Livrable 3 :

- Technologie : Blockchain de Consortium utilisant le consensus PBFT (Practical Byzantine Fault Tolerance).
- Pourquoi PBFT ? Contrairement au "Minage" (Proof of Work), PBFT est économe en énergie (durable) et offre une validation immédiate des transactions, ce qui est adapté aux flux IoT industriels.
- Fonctionnement :
    1. Node-RED valide les données entrantes (via HMAC).
    2. Périodiquement (ou sur alerte), Node-RED calcule un Hash (SHA-256) du lot de données.
    3. Ce Hash est envoyé vers le réseau Blockchain pour ancrage.
    4. Bénéfice : Impossible de modifier les logs de température a posteriori dans la base de données, car le hash ne correspondrait plus à celui gravé dans la Blockchain (Preuve d'intégrité).

4. Restitution synthétique et argumentée

4.1 Justification des Choix Réalisés

- MQTT sur TLS vs Modbus : Le choix de MQTT (Livrable 3) est validé pour sa légèreté et sa capacité à gérer des réseaux instables (reconnexion automatique). L'ajout de TLS comble la lacune de sécurité native de Modbus.
- Gateway Edge (Raspberry Pi) : Ce choix architectural permet de sécuriser les objets contraints (Arduino) derrière un point unique capable de porter des calculs cryptographiques lourds (vérification Blockchain) et de bufferiser les données en cas de coupure 4G.
- QoS 1 : Ce niveau de service garantit qu'aucune alerte critique (ex: Qualité de l'air dangereuse) n'est perdue, quitte à gérer quelques doublons, ce qui est préférable à une perte d'information dans un contexte de sécurité industrielle.

4.2 Bénéfices Attendus

- Sécurité "Baseline" ETSI respectée : Le prototype dépasse la simple connectivité pour offrir une résistance aux attaques élémentaires (Rejeu, Écoute, Usurpation).
- Confiance Numérique : Grâce à la signature HMAC et à l'ancrage Blockchain, les données visualisées sur le Dashboard peuvent servir de preuve légale en cas d'incident.
- Durabilité : L'architecture est modulaire. On peut remplacer un capteur Arduino sans toucher au cœur du système (Broker/Node-RED), et l'usage du consensus PBFT minimise l'impact écologique du module Blockchain.

4.3 Limites et Risques

- Gestion des Clés (Secrets.h) : Actuellement, les clés sont stockées sur les cartes SD des Raspberry Pi et dans le code Arduino. Risque : Si une carte est volée, la clé est compromise. Amélioration future : Utiliser des modules matériels de sécurité (TPM ou Secure Element ATECC608) sur les Arduinos.
- Point de Défaillance Unique (SPOF) : La Raspberry Pi est le point central. Si elle tombe en panne, toute la remontée de données s'arrête. Une redondance (Cluster de 2 RPi) serait nécessaire pour une mise en production réelle.

4.4 Cohérence Globale

Nous partons d'un besoin de fiabilité (Livrable 1), passons par une preuve de concept fonctionnelle (Livrable 3 avec MQTT/Arduino), et aboutissons ici à une architecture mature (Livrable 4) qui ajoute les couches indispensables de cybersécurité (HMAC/TLS) et de traçabilité (Blockchain) pour transformer un prototype en solution industrielle viable.

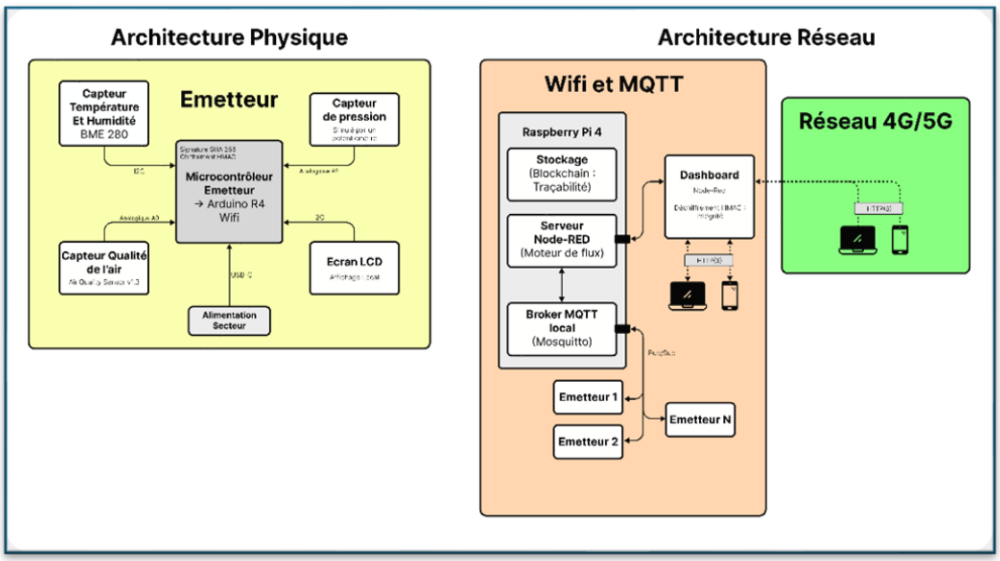

Figure 1 : Architecture Réseau