Livrable 4 — Campus IoT (Groupe 3 FISA INFO 2024/2027)

Projet : Campus IoT

Auteurs : BANIZETTE Matthieu, GACHENOT Antoine, PELLIZZARI Théo

Date : 06/02/2026

1) Dossier de compréhension et d'analyse

1.1 Reformulation du contexte et de ce qui pose question

Le projet Campus IoT vise à instrumenter les salles du bâtiment Orion afin de collecter des mesures environnementales (température, humidité, CO2, présence) et piloter des actionneurs (moteur, speaker) en temps réel. L'architecture actuelle s'appuie sur des capteurs/actuateurs connectés à une gateway locale, un broker MQTT (Mosquitto) et un backend FastAPI exposant des APIs et des WebSockets pour le front-end. Le protocole MQTT a été spécifié (arborescence de topics, QoS, retain, session persistante) et des tests de fiabilité/latence ont été menés.

Le cas d'usage principal est la supervision environnementale (qualité de l'air, confort thermique) et la réaction automatique (alertes sonores, ouverture/fermeture motorisée). Les utilisateurs visés sont les équipes techniques, les responsables de site et les occupants. Le système doit donc concilier temps réel, disponibilité et fiabilité tout en restant sobre en ressources.

Les questions clés portent sur la fiabilité des données, la sécurité des échanges et la sobriété énergétique dans un contexte Edge/IoT. Plus précisément :

- Comment garantir l'intégrité et l'authenticité des mesures et commandes ?
- Comment limiter les pertes de messages lors des micro-coupures réseau ?
- Comment réduire le trafic tout en conservant la réactivité en cas d'anomalie ?
- Comment tracer les actions et les événements critiques de manière robuste ?

Ces interrogations exigent une architecture durable et sécurisée, cohérente avec les contraintes IoT et les recommandations ETSI EN 303 645. Elles impliquent également de préciser les responsabilités entre objet, gateway et backend, et de définir un périmètre clair de données critiques (alertes CO2, commandes d'actuateurs, accès administratifs).

1.2 Enjeux techniques, énergétiques et de confiance

Enjeux techniques

- Fiabilité de transport : garantir la livraison des messages sensibles (QoS 1 + sessions persistantes).
- Interopérabilité : topics normalisés, payloads cohérents, compatibilité capteurs/actuateurs.
- Scalabilité : capacité à ajouter des salles ou des typologies de capteurs sans refonte.
- Observabilité : logs et métriques pour diagnostiquer latence, pertes, charge.
- Résilience : tolérance aux coupures courtes, reprise automatique et synchronisation via retain.
- Qualité de données : gestion des valeurs aberrantes, horodatage fiable, filtrage côté edge.

Enjeux énergétiques

- Sobriété réseau : limiter la fréquence d'émission, utiliser des modes adaptatifs (ex. BME280 en « fast » seulement en anomalie).
- Sobriété côté capteurs : minimiser les transmissions inutiles, privilégier les deltas et les heartbeats.
- Empreinte infrastructure : un broker local limite les allers-retours cloud et réduit la latence.
- Compromis précision/sobriété : fréquence adaptée par typologie de capteurs et niveau d'alerte.

Enjeux de confiance

- Authentification forte des utilisateurs (JWT, rôles, mots de passe chiffrés avec bcrypt).
- Intégrité des données (HMAC-SHA256, horodatage, traçabilité).
- Conformité aux bonnes pratiques (ETSI EN 303 645 : mots de passe, rôles, mises à jour, durcissement).
- Traçabilité des décisions automatiques pour expliquer les actions (audit, responsabilité).

1.3 Vision globale des éléments à fiabiliser, sécuriser ou améliorer

1. Fiabilisation des échanges MQTT
    - Maintenir QoS 1, retain pour les mesures critiques, sessions persistantes pour la continuité.
    - Adapter les rythmes d'émission selon le type de capteur et le niveau d'alerte.
2. Sécurisation des communications
    - Contrôler l'accès applicatif via authentification JWT côté backend.
    - Appliquer les règles d'accès par rôles pour les fonctions sensibles.
3. Authentification et autorisation
    - Gestion des utilisateurs par rôles (admin, technician, manager, user).

- Vérification des permissions sur les endpoints sensibles.

4. Intégrité et validation
    - HMAC-SHA256 + timestamp côté capteurs/gateway, vérification côté backend.
    - Détection d'anomalies basiques sur les valeurs et cohérence des unités.
5. Traçabilité
    - Journalisation des événements clés (alertes, commandes d'actuateurs, connexions).
    - Conservation en base pour l'audit et l'analyse.
6. Sobriété énergétique
    - Rythmes d'émission adaptatifs, delta-based, heartbeat minimal.
    - Optimisation du trafic et limitation des duplications.

2) Proposition d'architecture IoT sécurisée et durable

2.1 Architecture IoT fonctionnelle (objets, communications, services)

Objets IoT

- Capteurs : BME280 (temp/hum), capteur CO2, capteur de présence (HC-SR04), potentiomètre.
- Actuateurs : moteur (ouverture/fermeture), speaker (alertes).
- Capteurs configurés avec des seuils de confort et un mode alerte déclenchant une fréquence plus élevée.

Gateway locale

- Agrégation des capteurs, pré-traitement (filtrage/deltas), signature HMAC.
- Routage vers le broker MQTT.

Broker MQTT (Mosquitto)

- Topics structurés : campus/orion/{ROOM}/...
- QoS 1, retain sur capteurs, sessions persistantes (clean session = false).

- Publication et souscription via Mosquitto en mode standard (allow_anonymous = true dans la configuration actuelle).

Backend (FastAPI)

- Validation HMAC + timestamp.
- Gestion des alertes, anomalies, règles métier.
- APIs REST + WebSocket pour le front.
- Historisation des mesures côté base de données.

Frontend

- Supervision (tableaux de bord, alertes, historique).

Base de données

- Stockage des mesures, alertes, journaux d'audit.
- Indexation par salle/horodatage pour faciliter les analyses et audits.

2.2 Intégration des mécanismes de sécurité IoT

Authentification

- Utilisateurs : JWT + rôles (admin, technician, manager, user).
- Mots de passe stockés en bcrypt côté backend.

Chiffrement

- Communications MQTT sur le broker local en mode non chiffré dans l'état actuel.
- Accès au backend via HTTP local et WebSocket selon l'architecture du projet.

Intégrité

- HMAC-SHA256 sur les messages capteurs et commandes.
- Validation des timestamps et des signatures côté backend.

Autorisation

- Backend : RBAC selon rôles.
- Traçabilité des actions administratives (qui, quoi, quand) via l'audit.

Durcissement

- Rotation des secrets applicatifs et suppression des credentials par défaut.
- Mise à jour régulière des firmwares.

2.3 Traçabilité / registre distribué (quand pertinent)

Pour les événements critiques (alertes CO2, commandes d'ouverture/fermeture, actions administratives), la traçabilité s'appuie sur les journaux d'audit du backend et l'historisation en base de données. L'objectif est de fournir une preuve d'intégrité fonctionnelle et un historique exploitable pour l'analyse, sans ajouter un registre distribué qui n'est pas présent dans le projet.

3) Restitution synthétique et argumentée

3.1 Choix réalisés

- MQTT QoS 1 + retain + sessions persistantes : compromis fiabilité/latence adapté à l'Edge.
- Rythmes d'émission adaptatifs : sobriété énergétique en régime normal et réactivité en anomalie.
- HMAC + timestamps : intégrité et lutte contre la falsification/replay.
- Journalisation côté backend : traçabilité des actions critiques.

3.2 Bénéfices attendus

- Fiabilité accrue des mesures et commandes, même lors de micro-coupures.
- Réduction du trafic et meilleure autonomie des capteurs.
- Confiance renforcée grâce à l'authentification et l'intégrité des données.
- Audit facilité pour les événements sensibles.
- Meilleure qualité de service avec une latence stable et une reprise rapide.

3.3 Limites et risques

- Complexité accrue liée à la maintenance des règles d'accès et des mécanismes de sécurité applicatifs.
- Coût opérationnel pour la maintenance de la sécurité et des firmwares.
- Risque de dérive si les règles de qualité de données ne sont pas suivies.

3.4 Cohérence globale

La proposition s'appuie sur les choix validés au Livrable 3 (MQTT, QoS 1, retain, sessions persistantes) et les complète par une couche sécurité applicative, tout en conservant la sobriété énergétique. L'architecture est évolutive, conforme aux bonnes pratiques ETSI, et cohérente avec les contraintes d'un campus IoT réel. Les mécanismes proposés restent pragmatiques et compatibles avec une montée en charge progressive (ajout de salles, nouveaux capteurs, nouveaux usages).