Optimisation du confort pour les habitants des immeubles

Data ingestion pipeline

Komi Jean Paul ASSIMPAH
Louis MALMASSARY - François FLANDIN

Octobre 2025

Table des matières

Ubiquitous Language **1**

i Description Système **2**
i.1 Besoins capteurs . . . 2
i.1.a Température . . . 2
i.1.b Qualité de l'air . . . 2
i.1.c Ouverture fenêtre/porte d'entrée . . . 3
i.1.d Présence . . . 3
i.2 Données externes . . . 3
i.2.a Météo . . . 3
i.3 Environnement cible . . . 4
i.3.a Caractéristiques . . . 4
i.3.b Schema . . . 4

ii Sources d'incertitudes **5**
ii.0.a Incertitudes physiques (Capteurs) . . . 5
ii.0.b Incertitudes techniques (Infrastructure) . . . 6

iii Traitement et Fiabilisation des Données **6**
iii.1 Données manquantes. . . . 6
iii.2 Annotation des données . . . 6
iii.3 Fusion de capteurs . . . 6
iii.4 Filtrage des valeurs aberrantes . . . 7
iii.5 Maintenance et Calibration . . . 7
iii.6 Maintenance et Calibration . . . 7

iv Pipeline **8**
iv.1 Architecture réseau . . . 8
iv.2 Stockage de la data . . . 8
iv.2.a Traitement des flux (Stream vs. Batch) . . . 8
iv.3 Data ingestion pipeline . . . 9
iv.3.a Schema . . . 9

Ubiquitous Language

Client définit l'organisme ou la personne gérant l'immeuble sur lequel notre solution est installée.

Utilisateur définit les habitants d'un appartement.

Rappel objectif métier

L'objectif du projet est d'apporter pour chaque appartement d'un immeuble une solution qui permettrait de maintenir le confort des habitants, cela passe par une bonne regulation du chauffage, ainsi qu'une mesure de la qualité de l'air.
L'objectif est aussi de pouvoir apporter des métriques à l'organisme ou la personne qui gère l'immeuble pour orienter des travaux de rénovation ou d'isolation.

i Description Système

i.1 Besoins capteurs

i.1.a Température

Nous aurons besoin d'un capteur de température par pièce de vie, soit salon, cuisine, chambres et salle de bain. Ce qui nous permettra d'optimiser le chauffage pièce par pièce.
Les capteurs de température devront être capables de mesurer la température dans une plage étendue, idéalement entre -20 et 45,°C. 2 Cette plage large est nécessaire non seulement pour gérer le confort, mais aussi pour prévenir des situations "extrêmes" : une mesure à -20,°C peut détecter une panne de chauffage critique en hiver ou une fenêtre laissée ouverte, prévenant ainsi des risques de gel des canalisations, tandis que la limite haute détecte les surchauffes.
Étant donné l'inertie du changement de température, un haut taux de rafraichissement ne sera pas nécessaire. Une mesure toutes les 10 minutes sera suffisante, ce qui nous permettra d'observer les changements lents, tout en pouvant corriger "rapidement" si une mesure est manquante ou incohérente. Quant à la résolution de la mesure, une grande précision n'est pas nécessaire. L'objectif est d'assurer le confort thermique humain. Une petite différence de 0.2 à 0.5,°C n'est généralement pas perceptible par les utilisateurs. De plus, les actionneurs (les systèmes de chauffage) n'ont pas une telle finesse de régulation. Cette précision représente donc un compromis optimal entre le besoin utilisateur et le coût des capteurs.

i.1.b Qualité de l'air

Pour calculer la qualité de l'air, nous aurons besoin de capter et mesurer plusieurs gaz et particules. Plutôt qu'un seul point de mesure central, qui ne serait pas représentatif de tout le logement, il est nécessaire d'adopter un **placement stratégique des capteurs** en fonction de la source de pollution :

Nous aurons besoin de mesurer :

- **dioxyde de carbone (CO2) :** Indicateur principal de ventilation. Si trop élevé (>1000 ppm), l'air devient vicié et fatigue les occupants. Ce capteur est particulièrement utile dans les chambres à coucher, où le CO2 s'accumule pendant la nuit et peut affecter la qualité du sommeil.
- **monoxyde de carbone (CO) :** Nécessaire impérativement s'il y a des appareils qui fonctionnent au gaz (plaques de cuisson, chaudière, etc.) et à placer à proximité de ces sources de combustion.
- **COV (Composés Organiques Volatils) :** Ce sont des particules qui sont libérées dans l'air par les meubles, la peinture, les produits ménagers, etc.
- **particules fines :** Important si l'appartement est situé au bord d'une route très fréquentée ou dans un endroit avec beaucoup de pollution.

Les capteurs de COV et de particules fines seront idéalement placés dans les pièces de vie principales (salon) et à proximité de sources potentielles comme la cuisine (cuisson, aérosols).

- **taux d'humidité :** L'humidité excessive favorise les moisissures et augmente la concentration de certains polluants. Ce capteur est pertinent dans la salle de bain et la cuisine.

Contrairement à la température, on aura besoin d'une bonne résolution pour mesurer la qualité de l'air, car ce sont des données importantes pour la santé, mais si cela revient trop cher on pourra réduire légèrement la résolution.

i.1.c Ouverture fenêtre/porte d'entrée

Nous aurons besoin d'un capteur par fenêtre et d'un capteur pour la porte d'entrée, pour tenir compte de l'air plus frais de l'extérieur, et prévenir l'utilisateur si il oublie de les refermer, il faudra alors définir une durée minimale à partir de laquelle prévenir l'utilisateur.
Pour cette donnée, la resolution est inutile, on envoie qu'un octet 1 ou 0, donc la priorité ici c'est d'avoir la donnée immédiatement.

i.1.d Présence

La présence de l'utilisateur à son domicile est une donnée importante, elle permet au système de savoir quand activer le chauffage. De plus, avec de l'apprentissage par intelligence artificielle, on pourra prédire quand l'utilisateur est sur le point de rentrer chez lui, permettant de régler le chauffage sur une température moindre avant qu'il n'arrive.

Pour la détection, la solution retenue est un **badge à accrocher aux clés** de l'utilisateur. Pour clarifier la technologie, nous proposons d'utiliser une solution basée sur le **Bluetooth Low Energy (BLE)**, par exemple via le protocole **iBeacon**. Le badge serait une simple balise émettrice, et notre IoT-Gateway dans l'appartement agirait en tant que récepteur.

Cette approche s'intègre parfaitement à notre architecture : la gateway détecte l'événement d'entrée (apparition du signal) ou de sortie (disparition du signal) et l'envoie immédiatement au pipeline.

Cette technologie répond donc au besoin sans nécessiter une précision de localisation exagérée dans l'appartement ; seule l'efficacité et la rapidité de détection lors de l'entrée ou de la sortie sont requises.

i.2 Données externes

i.2.a Météo

Pour les besoins de notre projet, les données météo proviendront de **deux sources complémentaires** :

- **Un service externe (API Météo) :** Il sera rafraîchi quotidiennement pour obtenir les prévisions de température. Ces prévisions sont principalement destinées au Modèle IA pour anticiper les besoins de chauffage.
- **Une station météo locale :** Une station météo est installée sur l'immeuble pour fournir les données en temps réel. Elle fournit la température hyper-locale, c'est-à-dire la mesure exacte au niveau du bâtiment, qui peut différer de plusieurs degrés des prévisions générales d'un service API.

Cette donnée locale (température) est critique pour la fiabilité du calcul de l'indice d'isolation thermique (IIT).

En plus de la température, la station météo fournit d'autres données essentielles pour améliorer les prédictions de chauffage :

- **La vitesse du vent :** C'est un point clé. Un vent fort accélère massivement la perte de chaleur d'un bâtiment (refroidissement éolien). Le Modèle IA peut utiliser cette donnée pour anticiper une demande de chauffage plus forte.
- **L'humidité extérieure :** L'air sec et l'air humide ne transfèrent pas la chaleur de la même manière. C'est également une donnée pertinente pour le modèle de prédiction.

Pour l'intégration de ce capteur, la station météo sera un capteur partagé pour tout l'immeuble. Le système central (Fog dans l'immeuble) gérera ce capteur et publiera ses données sur le Kafka Cluster. Les différentes Gateway ModBus (de chaque appartement) s'abonneront simplement à ce flux de données pour récupérer les valeurs en temps réel nécessaires à leurs propres calculs d'IIT.

i.3 Environnement cible

i.3.a Caractéristiques

Pour les caractéristiques de l'environnement, on est dans un appartement avec des conditions plutôt moyennes, il faut que les capteurs résistent à ces paramètres :

température	de -20 à 45 °C
humidité	de 0 à 100 h.r

Table 1: Plage d'utilisation cible des capteurs

i.3.b Schema

FIGURE 1: Schéma d'un appartement type avec le placement des capteurs

ii Sources d'incertitudes

L'incorrection des données peut provenir de multiples facteurs, qu'il est crucial d'identifier pour assurer la fiabilité du système. Nous les classons en deux catégories : physiques et techniques.

ii.0.a Incertitudes physiques (Capteurs)

Chaque type de capteur possède ses propres sources d'erreur spécifiques :

- **Capteurs de température :** L'incertitude peut venir d'un mauvais placement (trop près d'un radiateur, en plein soleil) ou d'une dérive de calibration dans le temps, faussant la mesure de confort.
- **Capteurs de qualité de l'air (Gaz, COV, Particules) :** Ils sont très sensibles à l'encrassement. La poussière ou les aérosols (produits ménagers) peuvent fausser les lectures. Spécifiquement dans la cuisine, les dépôts de graisse peuvent obstruer les capteurs de COV et de particules fines.
- **Capteurs de contact (Portes/Fenêtres) :** L'incertitude est souvent mécanique, comme une usure ou un désalignement de l'aimant, menant à de faux négatifs (fenêtre signalée fermée alors qu'elle est ouverte).
- **Badge de présence (BLE) :** Des interférences avec d'autres appareils 2.4GHz (Wi-Fi, micro-ondes) ou une batterie faible sur le badge peuvent causer des détections erratiques (fausses absences).
- **Défaillance générale :** Un capteur peut devenir défectueux suite à un choc quelconque ou simplement à son usure naturelle.

ii.0.b Incertitudes techniques (Infrastructure)

- **Connectivité (Modbus filaire) :** Bien que filaire et robuste, notre architecture Modbus n'est pas exempte d'incertitudes. Nous pouvons citer les perturbations électromagnétiques si les câbles de données longent des câbles d'alimentation à forte puissance, ou un dommage physique aux câbles (lors de travaux, rongeurs) pouvant causer une perte de communication.
- **IoT-Gateway :** La gateway est un point central de défaillance. Une panne logicielle ou matérielle de celle-ci entraîne la perte de collecte des données de capteurs dans l'appartement.
- **Annotation des données :** Une mauvaise définition des unités de mesure ou une mauvaise association pièce/capteur au niveau de la gateway lors de l'installation peut corrompre les données dès leur entrée dans le pipeline.

iii Traitement et Fiabilisation des Données

Suite aux incertitudes identifiées à la section précédente, cette partie détaille les méthodes de traitement appliquées pour nettoyer, enrichir et fiabiliser les données brutes avant leur utilisation.

iii.1 Données manquantes.

Si on observe qu'il manque des données, par exemple suite à un **problème de connectivité** temporaire, il faudra rejouer les données précédemment reçues. Étant donné que l'on observe et mesure des données qui ont une inertie importante (température, humidité, qualité de l'air, ...) manquer une mesure n'impactera pas notre système. La logique de **rejeu de données** ou d'interpolation simple sera gérée au niveau de l'IoT-Gateway.

iii.2 Annotation des données

On aura besoin pour notre projet d'annoter les données pour mieux les répertorier plus tard dans la base de données ainsi que d'éviter les erreurs liées aux **mauvaises définitions des unités de mesure**. On aura donc :

- Une nité de mesure des capteurs : qui nous permettra d'être sûr qu'aucune erreur n'est faite sur la nature des données.
- Une pièce : surtout utile pour les capteurs de fenêtres et la température, nous permettra de déterminer dans quelle pièce la fenêtre est ouverte, et de réguler la température dans une pièce précise.
- Un appartement, un immeuble, une ville, permettra d'affiner les analyses pour le client, en fonction de la position de l'appartement, son orientation, etc.

iii.3 Fusion de capteurs

Il sera necessaire de faire de la fusion de capteurs sur 2 points :

- Pour calculer l'**IAQ (Indoor Air Quality)**. Ce score n'est pas une norme officielle mais un indicateur composite calculé par notre système. Sa pondération s'inspirera des seuils et recommandations des normes reconnues (par exemple, les lignes directrices de l'OMS ou les réglementations nationales sur les polluants de l'habitat). C'est en ayant ce score qu'on pourra prévenir l'utilisateur qu'il doit aérer chez lui ou non.

- Pour calculer l'**IIT (Indice d'Isolation Thermique)**. Il s'agit d'un indice propriétaire développé pour le client (le gérant de l'immeuble). Il ne remplace pas un diagnostic de performance énergétique (DPE) officiel, mais s'en inspire en corrélant la température intérieure, la température extérieure et la consommation énergétique du chauffage et la surface de l'appartement pour donner une évaluation dynamique de la performance de l'enveloppe de l'appartement

Toutes les deux heures, une moyenne des scores de qualité de l'air (IAQ-2H) sera calculée pour le client. Cet indicateur est particulièrement utile pour évaluer la performance des systèmes dont le client a la charge, comme la ventilation (VMC). Par exemple, si le score IAQ-2H est constamment mauvais dans plusieurs appartements à cause d'un taux de CO2 élevé (une composante clé de l'IAQ), c'est un indicateur direct d'un système de VMC défaillant ou sous-dimensionné, orientant ainsi des travaux de maintenance ciblés.

iii.4 Filtrage des valeurs aberrantes

Pour garantir la fiabilité des données, l'IoT-Gateway effectue un filtrage intelligent à la source. Ce filtrage gère deux types de situations :

- **Données Impossibles (Aberrantes) :** Si une donnée est physiquement impossible (par exemple, une température de $150\,°C$ ou un taux de CO2 négatif), la gateway la filtre et l'ignore. Cela est essentiel pour ne pas polluer le pipeline avec des données absurdes qui fausseraient les calculs.
- **Données Suspectes (Extrêmes) :** Si une donnée n'est pas impossible mais qu'elle est extrême et soudaine (par exemple, une très forte hausse de COV), la gateway l'analyse. Le système doit déterminer s'il s'agit d'une alerte réelle (comme un début d'incendie) ou d'un capteur défectueux (une erreur qui se répète). Ces données sont traitées avec une logique spécifique avant d'être validées.

iii.5 Maintenance et Calibration

Pour répondre aux incertitudes physiques comme la **poussière** ou la **graisse** sur les capteurs, le système enverra des notifications de **maintenance préventive** à l'utilisateur (via l'application) pour lui demander de nettoyer ses capteurs à intervalle régulier, garantissant ainsi la fiabilité des mesures de qualité de l'air.

iii.6 Maintenance et Calibration

Pour répondre aux incertitudes physiques comme la poussière ou la graisse sur les capteurs, le système enverra des notifications de maintenance préventive à l'utilisateur (via l'application) pour lui demander de nettoyer ses capteurs.

Le déclenchement de cette notification repose sur deux méthodes complémentaires :

- **Analyse intelligente (Méthode principale) :** Le module **Real Time Edge Computing** analyse en continu les flux de données. S'il détecte une dérive anormale (par exemple, un taux de COV qui ne redescend jamais à son niveau de base habituel), il en conclut que le capteur est probablement encrassé et déclenche une alerte.
- **Basée sur le temps (Méthode de sécurité) :** Indépendamment de l'analyse, une notification de rappel sera envoyée à intervalle régulier (par exemple, tous les 6 mois) pour garantir une vérification périodique.

Cette double approche garantit la fiabilité des mesures de qualité de l'air sur le long terme.

iv Pipeline

iv.1 Architecture réseau

Notre architecture est une architecture hybride de type **Edge/Fog computing**. Il est important de distinguer les deux environnements physiques :

- **L'Edge (par appartement) :** Un boîtier IoT-Gateway physique est installé dans chaque appartement. Il héberge tous les services locaux : `Gateway ModBus`, `Service Association`, `Real Time Edge Computing`, `Modèle IA` et la `BDD Locale (MongoDB)`.
- **Le Fog (central) :** Un serveur central est installé dans le local technique de l'immeuble. Il héberge tous les services partagés et centraux : le `Kafka Cluster`, le service `Météo`, le `Kafka Connect` et la `BDD Cloud (TimescaleDB)`.

La communication entre l'Edge (chaque appartement) et le Fog (le serveur central) se fait via le réseau Ethernet local de l'immeuble.

Pour la communication, nous opterons pour une technologie filaire, étant donné que l'espace d'un appartement est relativement restreint. Le câblage supplémentaire restera peu coûteux et nous pourrons tirer parti des installations électriques existantes.
Concernant le choix de la technologie de communication, il est nécessaire de privilégier une solution simple à déployer et à maintenir. De plus, le volume de données à transmettre reste limité, puisqu'aucun flux vidéo n'est prévu. Nous avons donc retenu `Modbus` pour assurer la communication entre les capteurs et la passerelle. L'architecture sera de type bus, avec une logique master/slave, où le master est le gateway modbus et les slave sont les capteurs, ModBus supporte plus de 240 slaves, donc largement suffisant pour le projet.

iv.2 Stockage de la data

Pour le stockage des données, différentes technologies seront utilisées en fonction du contexte. En local, il est essentiel que les données soient rapidement accessibles et que l'accès à la dernière information disponible soit garanti. En revanche, dans le cloud, accessible uniquement par le client, l'intégrité des données doit être assurée afin de garantir que le client dispose d'informations fiables et précises, ce qui permet d'éviter toute erreur dans la réalisation de travaux de rénovation.
Ainsi, pour le stockage local, une base de données légère telle que `MongoDB` pourra être utilisée, tandis que pour le cloud, une solution comme `TimescaleDB`, une base SQL optimisée pour l'analyse de données temporelles, s'avère particulièrement adaptée à notre objectif métier.

iv.2.a Traitement des flux (Stream vs. Batch)

L'architecture du pipeline est conçue pour gérer deux types de flux de données, chacun ayant un objectif distinct :

- **Flux Continu (Stream) :** Ce flux est dédié au traitement en temps réel (faible latence) et au confort immédiat de l'habitant. Il concerne toutes les données événementielles et instantanées :

- Les **événements de présence** et l'**ouverture/fermeture des fenêtres** (capteurs de contact).
- Les **mesures brutes** de température et de qualité de l'air.

Ce flux alimente le Real Time Edge Computing et le Modèle IA pour la prise de décision rapide (ex : ajustement du chauffage) et le filtrage des valeurs aberrantes.

- **Flux par Lot (Batch) :** Ce flux est dédié au traitement différé (latence de quelques heures) pour les analyses du gérant de l'immeuble. Il concerne :
  - Les données externes, comme les **prévisions météo** (rafraîchies quotidiennement).
  - Le calcul des **métriques** (scores IIT et IAQ-2H), qui sont des moyennes agrégées sur une période de 2 heures avant d'être envoyées au cloud.

  Ce flux garantit la fiabilité et l'intégrité des données à long terme pour le module **Analytics** du client.

iv.3 Data ingestion pipeline

Le modèle final de notre data ingestion pipeline, visible sur notre schéma d'architecture, sépare clairement les rôles de collecte, d'enrichissement et de traitement :

- **1. Collecte et Commande (Gateway ModBus) :** La **Gateway ModBus** agit en tant que maître sur le bus filaire. Elle a un double rôle :
  - **Collecter** les données brutes des différents capteurs (esclaves) et les transmettre au Service Association.
  - **Recevoir** les "Commandes actionneurs" depuis le Kafka Cluster pour piloter les actionneurs (chauffage et le climatiseur)
- **2. Enrichissement (Service Association) :** Les données brutes sont immédiatement transmises au **Service Association**. C'est ce module qui gère l'annotation des données : il enrichit les mesures brutes en y ajoutant le contexte indispensable (type de donnée, unité de mesure, pièce associée, identifiant de l'appartement et timestamp).
- **3. Diffusion (Kafka Cluster) :** Une fois annotées, les données sont publiées sur le **Kafka Cluster**, qui sert de bus de données central.
- **4. Traitement (Edge & Fog) :**
  - **En local (Edge) :** Le **Real Time Edge Computing** et le **Modèle IA** consomment les données en temps réel depuis Kafka pour prendre les décisions (chauffage), envoyer des notifications et effectuer le filtrage des valeurs aberrantes. Les données sont aussi stockées dans la **base de données locale (MongoDB)** pour l'application utilisateur.
  - **Au niveau de l'immeuble (Fog) :** Le service **Kafka Connect** transfère les données pertinentes (les scores IIT et IAQ-2H) vers le **Fog dans l'immeuble** pour un stockage à long terme dans la **base de données Cloud (TimescaleDB)**, à destination du client pour ses analyses.

iv.3.a Schema

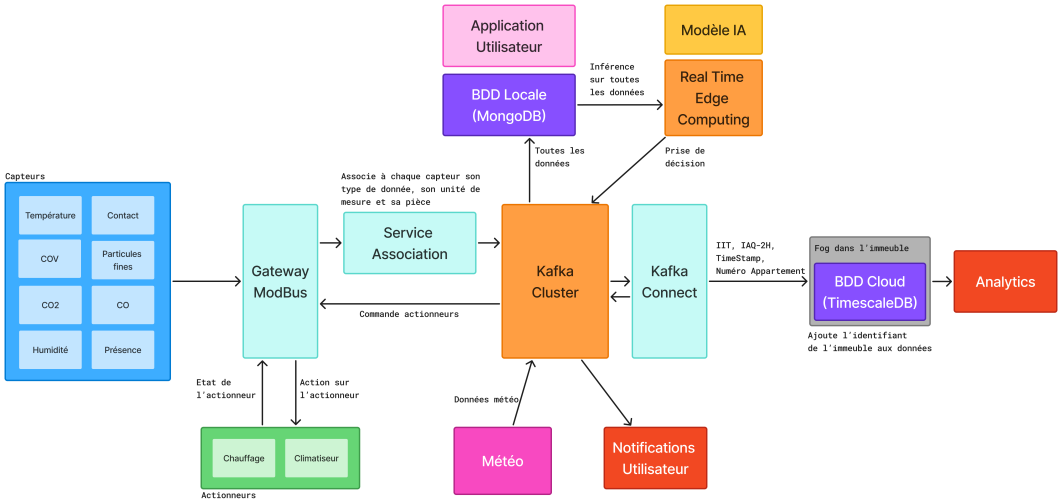

FIGURE 2: Data ingestion pipeline

Légende de l'architecture :

- **Composants "Edge" (dans l'appartement) :** Capteurs, Gateway ModBus, Service Association, BDD Locale, Modèle IA, Real Time Edge Computing, Application et Notifications Utilisateur.
- **Composants "Fog" (serveur central) :** Kafka Cluster, Météo, Kafka Connect, BDD Cloud, et Analytics.