



المعهد الوطني للبريد والمواصلات
ⵎⵓⵏⵉⵙⵜ ⵏ ⵓⵎⵎⵓⵏⵉⵙⵜ ⵏ ⵓⵎⵎⵓⵏⵉⵙⵜ
Institut National des Postes et Télécommunications

INSTITUT NATIONAL DES POSTES ET TÉLÉCOMMUNICATIONS

PLATEFORMES ET TECHNOLOGIES DE L'IOT

Conception et Implémentation d'une Plateforme IoT

Réalisé par :
Salmane BABA
Yassin NAJMI

Encadrante :
Pr AIT OUALHAJ OMAR

Année académique 2025–2026

Table des matières

1	Couche Embarquée : Edge Computing (Raspberry Pi)	3
1.1	Système d'Acquisition et d'Actionnement Physique	3
1.2	Détection de Présence Hybride et Vision Artificielle	3
1.2.1	Détection Matérielle Instantanée	3
1.2.2	Détection Logique et Qualification d'Intention (YOLOv11 ONNX)	4
1.3	Cinématique du Couvercle et Sécurité de Saturation	4
1.4	Architecture Logicielle du Programme Embarqué	5
1.4.1	Organisation des Threads Parallèles	5
1.5	Optimisation des Échanges : Edge Filtering et Formatage	5
1.5.1	Résilience et Persistance Tampon (Buffer Local)	6
1.6	Le Choix Architectural : MQTT sur WebSockets (Port 443)	6
2	Couche Cloud Azure : Ingestion et Traitement Analytique	7
2.1	Azure IoT Hub : Passerelle d'Ingestion Sécurisée	7
2.2	Azure Stream Analytics : Traitement Événementiel et Aiguillage	7
2.2.1	Logique de Requêtage Stream SQL et Séparation des Flux	8
2.2.2	Mécanismes Clés appliqués par le Job ASA	8
3	Modélisation des Données : Architecture Double Conteneur Cosmos DB	9
3.1	Conteneur 1 : Telemetry	9
3.2	Conteneur 2 : BinsInventory	10
3.3	Pourquoi ce Découplage ?	10
4	Couche Applicative : Plateforme Multi-Profiles	11
4.1	Architecture Technique de la Plateforme	11
4.2	Vue Manager : Tableau de Bord Décisionnel	11
4.3	Vue Driver : Navigation Opérationnelle	12
4.4	Vue AI Simulator : Simulation Prédicative	13
4.5	API REST et Services Métier	14
4.6	Bénéfices Fonctionnels	14

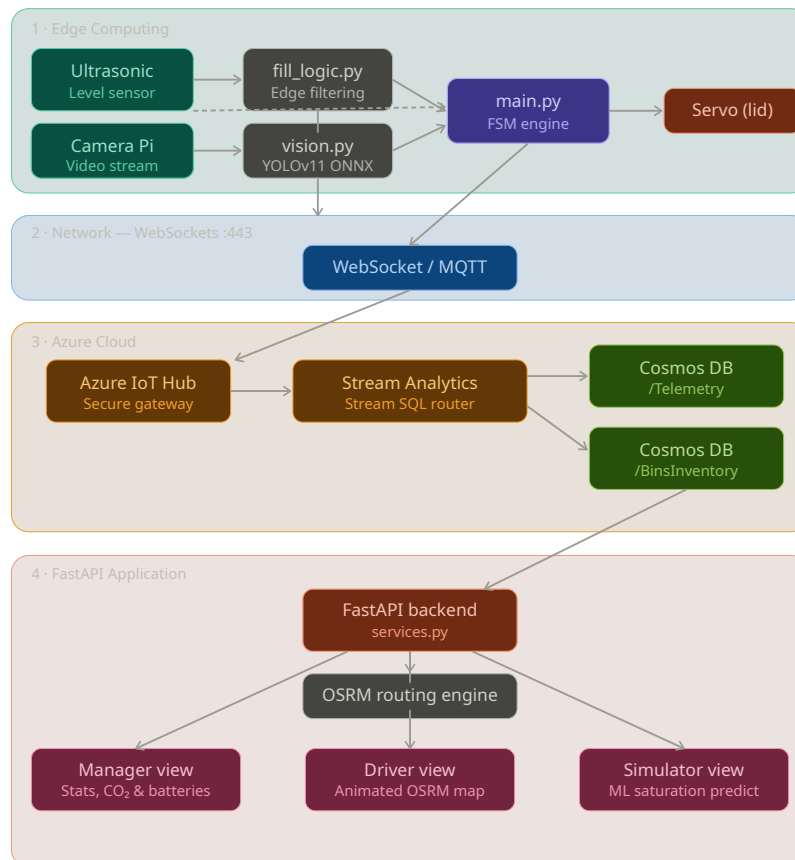
Introduction

Dans le contexte actuel de croissance urbaine accélérée et de transition vers les **Smart Cities**, la gestion intelligente des déchets constitue un enjeu majeur pour les collectivités territoriales. Les méthodes traditionnelles de collecte, souvent basées sur des tournées fixes et une absence de visibilité en temps réel, entraînent des inefficacités logistiques, des coûts élevés ainsi que des risques de saturation des infrastructures publiques.

Face à ces défis, le projet **GreenTech SmartBin** propose une solution innovante reposant sur la convergence des technologies **AIoT (Artificial Intelligence of Things)**, du **Cloud Computing** et de l'**Intelligence Artificielle**. L'objectif principal est de concevoir un système capable de surveiller automatiquement l'état de remplissage des bacs urbains, d'interagir intelligemment avec les usagers, puis de transmettre et d'exploiter ces données en temps réel afin d'optimiser les opérations de collecte.

L'architecture développée s'appuie sur une chaîne complète de traitement distribuée : une couche embarquée basée sur **Raspberry Pi** pour l'acquisition et l'analyse locale des données, une infrastructure Cloud **Microsoft Azure** pour l'ingestion, le traitement et le stockage massif des flux IoT, ainsi qu'une plateforme applicative multi-profil destinée à la supervision, à l'optimisation des tournées et à la simulation prédictive.

Ce rapport présente les différents composants techniques de cette architecture, depuis le traitement en périphérie (*Edge Computing*) jusqu'à la valorisation applicative des données, en mettant en évidence les choix technologiques, les mécanismes d'optimisation et les bénéfices apportés par cette approche dans le cadre d'une gestion urbaine durable et intelligente.



1. Couche Embarquée : Edge Computing (Raspberry Pi)

Cette section détaille la gestion en périphérie de réseau (*Edge computing*) de l'acquisition physique, de l'actionnement mécanique et de l'architecture logicielle asynchrone qui pilote le bac intelligent.

1.1. Système d'Acquisition et d'Actionnement Physique

Le cœur matériel de la solution repose sur une carte micro-calculatrice **Raspberry Pi** configurée pour orchestrer les interactions avec l'environnement urbain à travers trois canaux physiques distincts :

- **Mesure du Niveau de Remplissage (HC-SR04 Interne)** : Ce capteur à ultrasons est fixé sous le couvercle et orienté verticalement vers le fond de la cuve. Il émet une impulsion acoustique et calcule son temps de vol pour déterminer la distance brute avec la surface des déchets, convertie localement en pourcentage de remplissage.
- **Détection d'Approche (Capteur de Proximité Externe)** : Positionné sur la face avant du châssis, ce capteur mesure en continu la proximité des citoyens. Lorsqu'une présence est détectée à une distance inférieure à 10 cm, un signal d'interruption matériel est immédiatement transmis au processeur central.
- **Actionnement du Couvercle (Servomoteur)** : Un servomoteur industriel, piloté en modulation de largeur d'impulsion (*PWM*), est couplé au mécanisme d'ouverture du couvercle pour permettre un dépôt des déchets entièrement automatisé et sans contact (*contactless*).

1.2. Détection de Présence Hybride et Vision Artificielle

Pour offrir une réactivité maximale tout en qualifiant finement le contexte de chaque interaction, le système s'appuie sur deux méthodes complémentaires :

1.2.1 Détection Matérielle Instantanée

La détection par ultrason en façade offre une solution simple, robuste et extrêmement économe en ressources. Elle garantit un déclenchement instantané de l'ouverture dès qu'un usager franchit le seuil de proximité de la poubelle, éliminant ainsi toute latence de calcul perceptible par l'utilisateur.

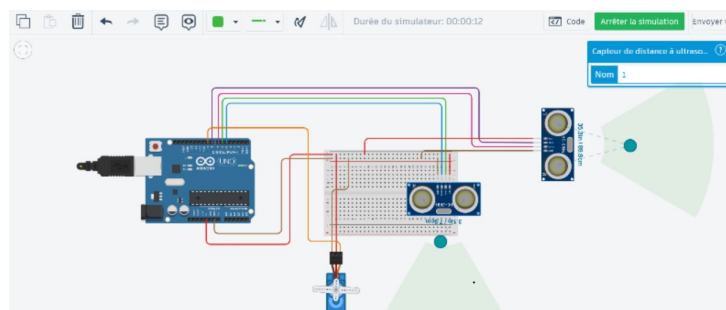


FIGURE 1 – Détection de présence matérielle par capteur à ultrasons externe.

1.2.2 Détection Logique et Qualification d'Intention (YOLOv11 ONNX)

En parallèle, une caméra embarquée capture le flux vidéo en temps réel. Afin d'enrichir le contexte métier de la télémétrie, un modèle de réseaux de neurones convolutifs **YOLOv11**, optimisé et compilé au format **ONNX**, est exécuté localement via le runtime CPU de la Raspberry Pi.

Cette approche Edge AI qualifie l'intention de l'utilisateur en isolant spécifiquement les classes liées aux personnes et aux dépôts de déchets (*classes person, backpack, handbag*). Si le modèle valide la présence d'une personne munie d'un sac, l'événement est corrélé et une trame de télémétrie filtrée de type `presence_event` est préparée pour l'envoi vers le Cloud.



FIGURE 2 – Qualification de l'intention de dépôt par le modèle YOLOv11 ONNX.

1.3. Cinématique du Couvercle et Sécurité de Saturation

L'état mécanique du servomoteur est strictement dicté par les règles d'hygiène et de sécurité du système :

- **Ouverture Automatique** : Dès qu'une présence valide est signalée (par interruption ultrason ou par inférence visuelle), le servomoteur bascule à un angle de 90°.
- **Fermeture Temporisée** : Afin de confiner les odeurs et de protéger le conteneur des intempéries, une minuterie non bloquante est initialisée. Si aucun signal de présence n'est détecté pendant une durée consécutive de 15 secondes, le signal PWM est réinitialisé pour ramener le servomoteur à sa position initiale (0°).
- **Verrouillage de Sécurité pour Saturation** : Lorsque le capteur interne enregistre une distance inférieure ou égale à 5 cm (indiquant un bac plein à 100%), la Raspberry Pi désactive immédiatement l'ouverture automatique et maintient le servomoteur verrouillé à 0°. Cette mesure de sécurité empêche tout débordement physique et protège le mécanisme d'un blocage de tringlerie.

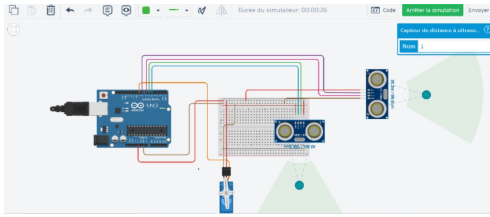


FIGURE 3 – Fermeture automatique (0°).

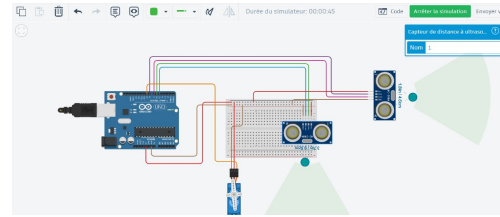


FIGURE 4 – Poubelle saturée à 4.5 cm.

1.4. Architecture Logicielle du Programme Embarqué

Le logiciel principal s'exécutant en arrière-plan (implémenté dans `main.py`) est architecturé selon un paradigme multi-threadé asynchrone afin de découpler les entrées/sorties physiques lourdes du traitement de données.

1.4.1 Organisation des Threads Parallèles

L'application divise son cycle d'exécution en trois processus légers s'exécutant en concurrence :

1. **Thread de Télémétrie (Fréquence : 2s)** : Il interroge les capteurs de distance, filtre le bruit de signal et applique la logique de paliers.
2. **Thread de Vision (Fréquence : Continu)** : Il effectue le décodage d'images de la caméra et pilote les prédictions YOLOv11.
3. **Thread de Communication Cloud (MQTT Asynchrone)** : Il gère la pile de messages sortants, écoute les directives descendantes du Cloud et maintient la liaison avec Azure IoT Hub.

1.5. Optimisation des Échanges : Edge Filtering et Formatage

Pour minimiser les coûts d'ingestion Cloud et optimiser la bande passante, le module `fill_logic.py` applique un **Edge Filtering** par paliers. Les données volumétriques ne sont envoyées que si la valeur franchit un seuil de transition significatif ([20%, 50%, 70%, 80%, 90%, 100%]).

Pour maintenir un schéma d'indexation fixe et performant dans la base NoSQL Azure Cosmos DB, l'application implémente un **schéma aplati**. Toutes les clés structurales sont systématiquement transmises dans le corps du payload JSON, forçant les variables non concernées par l'événement courant à la valeur `null`.

Listing 1 – Exemple de payload JSON pour l'événement `fill_level_update`

```
{
  "deviceId": "Bin_1",
  "type": "fill_level_update",
  "fill_level": 50,
  "alert": false,
  "detection_type": null
}
```

Listing 2 – Exemple de payload JSON pour l'événement `presence_event`

```
{
  "deviceId": "Bin_1",
  "type": "presence_event",
  "fill_level": null,
  "alert": null,
  "detection_type": "person_arrived"
}
```

1.5.1 Résilience et Persistance Tampon (Buffer Local)

En cas de perte de connectivité réseau (perte de signal 4G/GPRS dans l'environnement urbain), le client MQTT ne bloque pas l'exécution locale du bac. Les trames JSON générées sont immédiatement stockées dans une file d'attente FIFO (*First In, First Out*) managée en mémoire vive. Dès la restauration de la liaison avec Azure IoT Hub, le thread réseau vide le buffer local en transmettant les données en rafale (**Burst transmission**), garantissant l'intégrité temporelle absolue de l'historique de télémétrie.

1.6. Le Choix Architectural : MQTT sur WebSockets (Port 443)

Pour assurer la communication entre la couche embarquée et le Cloud Azure, nous avons choisi d'associer deux protocoles complémentaires : **MQTT** et **WebSockets**, routés sur le port **443**. Cette synergie s'inspire des meilleures pratiques industrielles actuelles (notamment dans le suivi logistique en temps réel et la gestion des flottes de véhicules connectés) et se justifie par trois critères majeurs :

- Efficacité Applicative de MQTT** : Le protocole MQTT est conçu pour l'IoT grâce à son modèle *Publish/Subscribe* ultra-léger. Contrairement au protocole HTTP qui nécessite l'ouverture d'une lourde session à chaque requête (mécanisme de *Request/Response*), MQTT maintient une connexion persistante avec une entête de message minimale (seulement 2 octets). Cela réduit drastiquement la consommation CPU de la Raspberry Pi et le volume global des données transmises.
- Standardisation et Compatibilité Web (Le rôle des WebSockets)** : Bien que MQTT soit optimal pour les machines, il n'est pas nativement supporté par les navigateurs web ou certaines passerelles applicatives standard. En encapsulant les paquets MQTT dans des connexions WebSockets, le flux de télémétrie s'aligne sur les standards du Web moderne. Cela permet un streaming bidirectionnel fluide en temps réel et facilite l'interopérabilité directe entre les données brutes de l'appareil et notre plateforme FastAPI.

2. Couche Cloud Azure : Ingestion et Traitement Analytique

La couche Cloud assure la réception, la validation et l'aiguillage des données en continu.

2.1. Azure IoT Hub : Passerelle d'Ingestion Sécurisée

Le flux réseau est réceptionné en première ligne par **Azure IoT Hub**. IoT Hub ne se comporte pas comme un simple broker MQTT standard (tel que Mosquitto) ; il agit comme une passerelle d'entreprise managée. Il gère un registre d'identités strict où chaque bac possède ses propres clés de chiffrement, et supporte le concept de Jumeau Numérique pour configurer les appareils à distance.

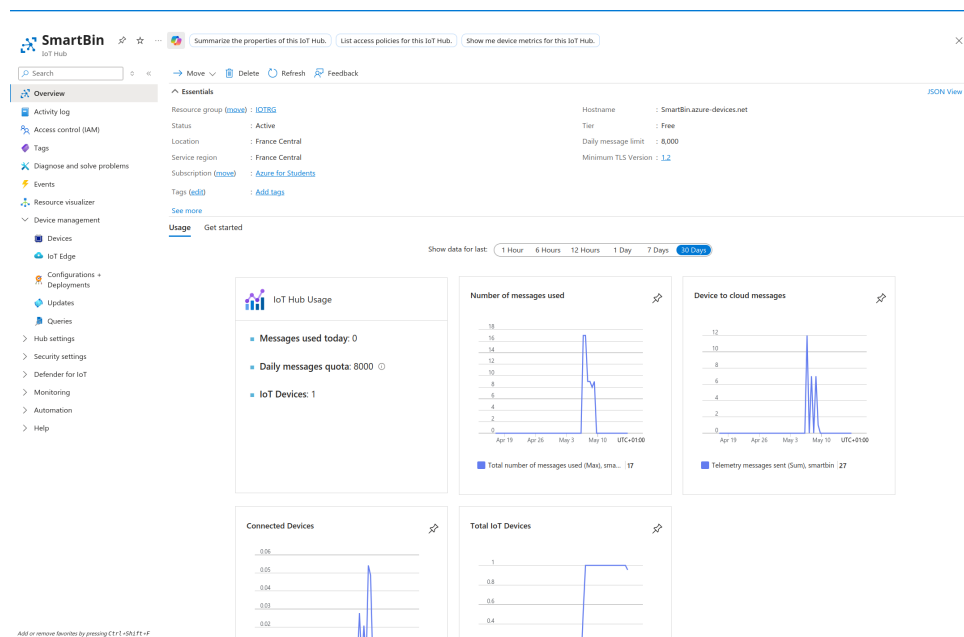


FIGURE 5 – Azure IoT Hub dashboard

2.2. Azure Stream Analytics : Traitement Événementiel et Aiguillage

Directement connecté à la sortie d'**Azure IoT Hub**, le job d'analyse managé **Azure Stream Analytics (ASA)** traite les flux de données continus en temps réel avec une latence de l'ordre de la milliseconde.

Son rôle va bien au-delà d'un simple relais : il agit comme un routeur logique, un validateur de schéma et un moteur de transformation de données avant leur persistance.

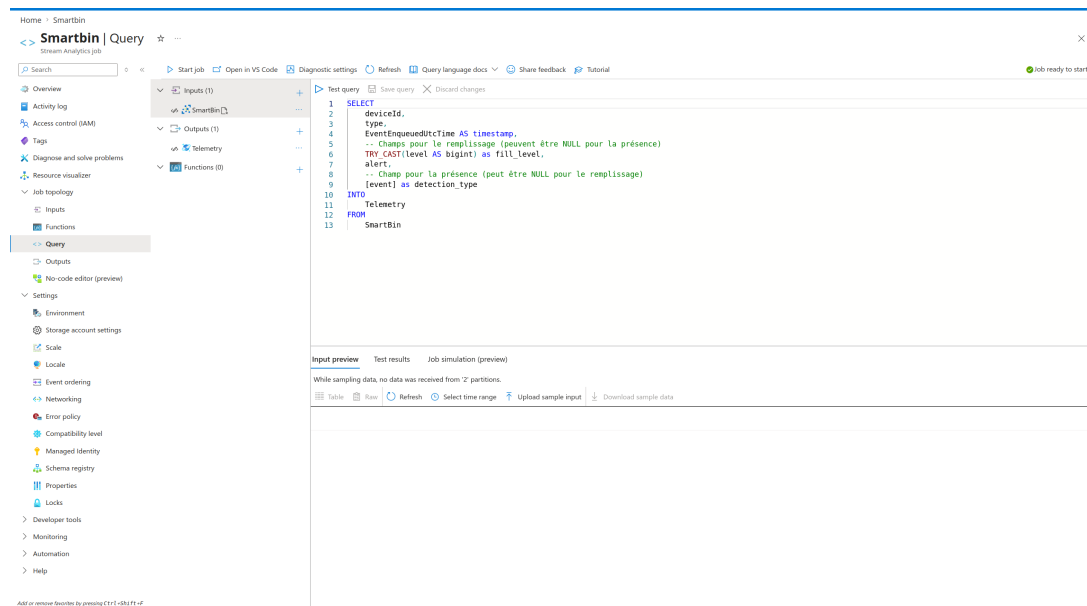


FIGURE 6 – Topologie du Job Azure Stream Analytics : Configuration des entrées, sorties et de la requête SQL d'aiguillage.

2.2.1 Logique de Requête Stream SQL et Séparation des Flux

Comme illustré dans l'interface d'architecture d'Azure, le traitement s'appuie sur une requête structurée de type **Stream SQL**. Cette requête écoute l'unique source d'entrée (IoTHubInput) et sépare les flux dynamiquement selon la nature de l'événement pour les distribuer vers leurs tables de destination respectives :

Listing 3 – Requete Stream SQL de routage des donnees dans Azure Stream Analytics

```

SELECT
  deviceId,
  type,
  EventEnqueuedUtcTime AS timestamp,
  -- Champs pour le remplissage (peuvent tre NULL pour la pr sence)
  TRY_CAST(level AS bigint) as fill_level,
  alert,
  -- Champ pour la pr sence (peut tre NULL pour le remplissage)
  [event] as detection_type
FROM
  [IoTHubInput]
WHERE
  type = 'fill_level_update' OR type = 'presence_event'

```

2.2.2 Mécanismes Clés appliqués par le Job ASA

Lors du transit des messages, Azure Stream Analytics exécute trois opérations fondamentales :

1. **Aplatissement du Schéma (*Schema Flattening*)** : Les trames JSON envoyées par la Raspberry Pi peuvent parfois arriver sous des formes imbriquées ou altérées par les métadonnées système d'IoT Hub. Le moteur ASA extrait les variables utiles

4. Couche Applicative : Plateforme Multi-Profiles

Cette couche constitue l'interface directe entre les données intelligentes produites par l'infrastructure Cloud et les différents acteurs opérationnels du système SmartBin. Elle expose, à travers une plateforme web moderne, les fonctionnalités de supervision, d'aide à la décision et de simulation prédictive adaptées à plusieurs profils métiers.

L'application a été développée selon une architecture **Frontend/Backend découplée**, reposant sur un backend **FastAPI** en Python et une interface utilisateur **ReactJS** dynamique.

4.1. Architecture Technique de la Plateforme

L'architecture applicative repose sur deux composants principaux :

- **Backend FastAPI** : expose une API REST sécurisée permettant l'accès aux données de télémétrie, aux algorithmes d'optimisation de tournées et aux modules d'intelligence artificielle.
- **Frontend React** : fournit une interface web interactive multi-vues adaptée aux différents rôles utilisateurs, avec mise à jour temps réel et visualisations graphiques avancées.

Le backend agit comme une couche d'orchestration centrale entre Azure Cosmos DB, le moteur de routage OSRM et les composants de visualisation.

4.2. Vue Manager : Tableau de Bord Décisionnel

Le profil **Manager** est destiné aux responsables municipaux ou opérateurs de supervision.

Cette interface centralise les indicateurs critiques du réseau de poubelles intelligentes :

- **Statistiques globales** : nombre total de bacs actifs, taux moyen de remplissage, nombre de bacs critiques.
- **Historique analytique** : évolution des niveaux de remplissage sur plusieurs périodes (7, 30 ou 90 jours), permettant l'identification de tendances saisonnières.
- **Visualisation cartographique** : carte interactive générée dynamiquement affichant l'ensemble des bacs avec leur état courant.

Cette vue transforme les données brutes de télémétrie en indicateurs stratégiques exploitables pour la planification des opérations.

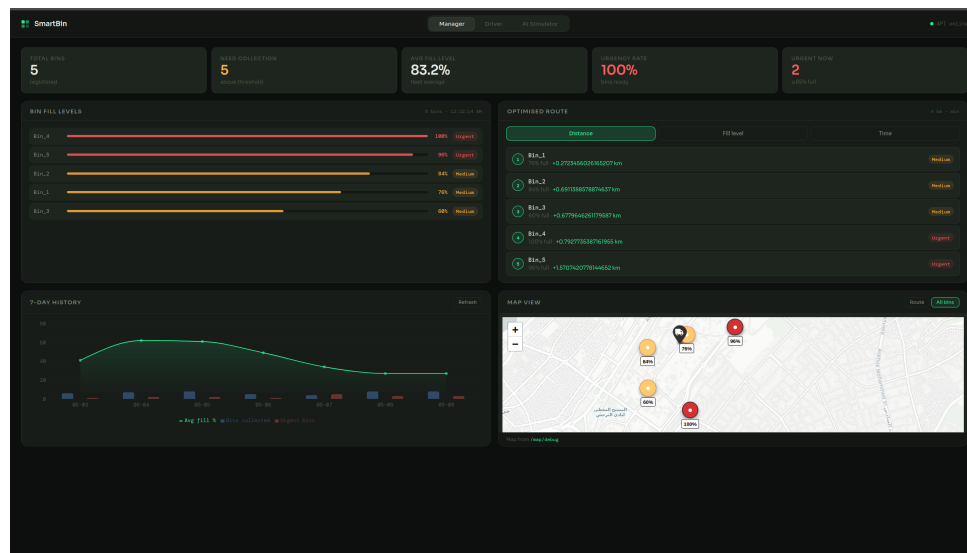


FIGURE 9 – Interface Manager : supervision globale et statistiques.

4.3. Vue Driver : Navigation Opérationnelle

Le profil **Driver** est conçu pour les chauffeurs ou agents de collecte sur le terrain.

Cette interface fournit une assistance directe aux tournées :

- **Affichage de la tournée optimisée** : liste ordonnée des bacs à collecter selon la stratégie choisie.
- **Navigation cartographique** : affichage de l'itinéraire calculé par le moteur OSRM basé sur les données routières OpenStreetMap.
- **Informations par arrêt** : niveau de remplissage du bac, distance depuis l'arrêt précédent, priorité de collecte.
- **Réduction du temps de trajet** : optimisation automatique visant à minimiser la distance totale parcourue.

Cette vue transforme l'algorithme d'optimisation en outil directement exploitable sur le terrain.

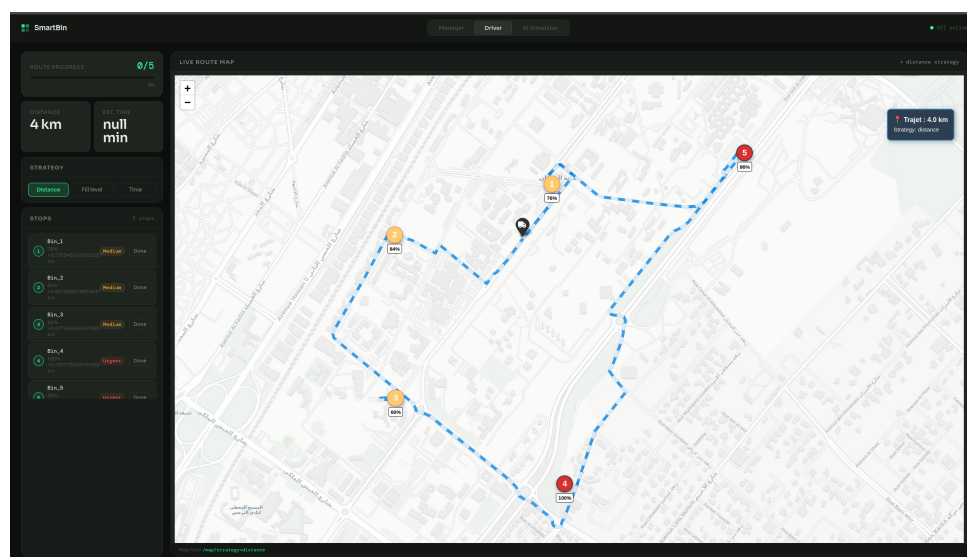


FIGURE 10 – Interface Driver : affichage de la tournée optimisée.

4.4. Vue AI Simulator : Simulation Prédicative

Le troisième profil est dédié à l'analyse prédictive et à la simulation.

Le module **AI Simulator** exploite un modèle de machine learning basé sur un **Random Forest Regressor** entraîné à partir de l'historique de télémétrie.

Ses capacités incluent :

- **Prédiction future des niveaux de remplissage** : estimation de l'état des bacs à une date cible donnée.
- **Simulation de scénarios "What-if"** : comparaison des tournées futures selon différents horaires.
- **Optimisation anticipative** : préparation proactive des collectes avant saturation réelle.
- **Visualisation cartographique simulée** : génération d'une carte représentant l'état futur prédit du réseau.

Le modèle apprend les rythmes de remplissage à partir de plusieurs variables temporelles :

- heure de la journée ;
- jour de la semaine ;
- vitesse historique de remplissage ;
- comportement spécifique de chaque bac.

Cette intelligence prédictive constitue la transition vers une gestion réellement autonome et anticipative des déchets urbains.

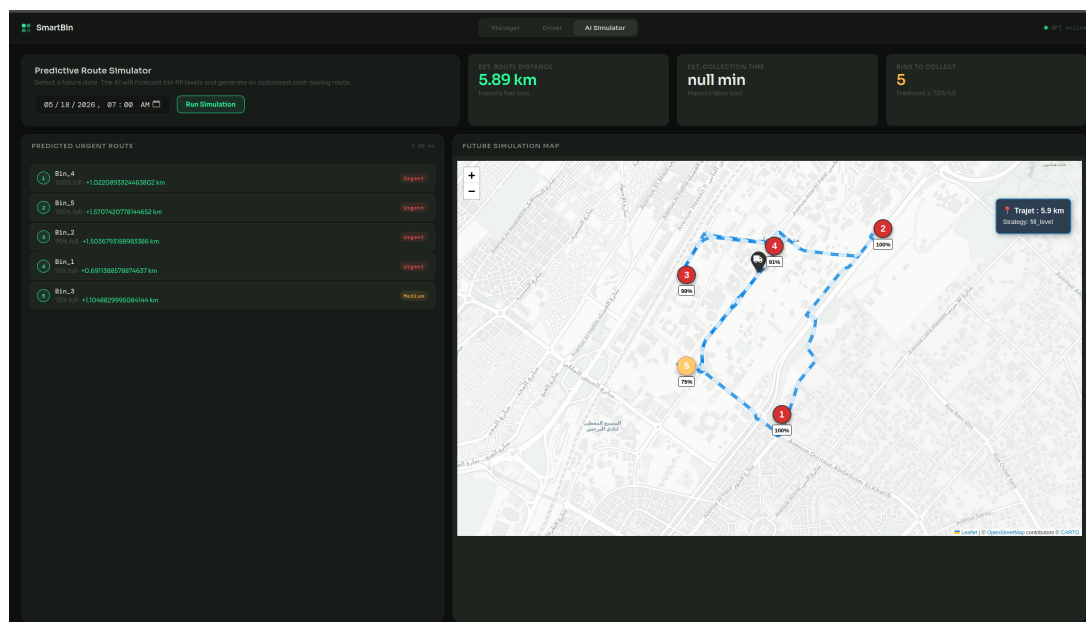


FIGURE 11 – AI Simulator : prédiction future et simulation de tournée.

4.5. API REST et Services Métier

Toutes les interactions de la plateforme transitent par une API REST structurée autour de plusieurs services :

1. **Bin Service** : récupération des bacs et calcul des statistiques.
2. **Route Optimizer** : génération des tournées selon trois stratégies :
 - optimisation par distance (*Nearest Neighbor*),
 - optimisation par priorité de remplissage,
 - optimisation temporelle prédictive.
3. **History Service** : analyse des historiques et tendances.
4. **AI Service** : entraînement du modèle et simulation future.
5. **Map Generator** : production des cartes interactives HTML via **Folium**.

Cette organisation modulaire garantit une maintenance facilitée, une forte extensibilité et une séparation claire des responsabilités logicielles.

4.6. Bénéfices Fonctionnels

La plateforme applicative SmartBin apporte plusieurs bénéfices majeurs :

- centralisation des opérations sur une interface unique ;
- réduction des coûts de collecte ;
- diminution des risques de débordement ;
- amélioration de la réactivité opérationnelle ;
- prise de décision assistée par intelligence artificielle ;
- préparation à une gestion urbaine autonome (*Smart City*).

Conclusion

Le projet **GreenTech SmartBin** démontre comment l'intégration cohérente des technologies **AIoT**, du **Cloud Computing** et de l'**Intelligence Artificielle** permet de concevoir une solution complète de gestion intelligente des déchets urbains.

À travers une architecture distribuée allant du **Edge Computing** embarqué sur Raspberry Pi jusqu'à une plateforme applicative web multi-profil, le système assure en temps réel la collecte, le traitement, l'analyse et la valorisation des données terrain. L'utilisation d'**Azure IoT Hub**, d'**Azure Stream Analytics** et d'**Azure Cosmos DB** garantit une infrastructure robuste, scalable et adaptée aux contraintes industrielles des environnements connectés.

Au-delà de la simple supervision, l'intégration de mécanismes d'optimisation de tournées et de modèles prédictifs basés sur le **Machine Learning** permet une transition vers une gestion proactive et anticipative, réduisant à la fois les coûts opérationnels, les risques de saturation et l'impact environnemental.

Ce projet illustre concrètement la convergence entre systèmes embarqués, services Cloud et intelligence décisionnelle, tout en s'inscrivant pleinement dans la vision des **Smart Cities** de demain, où les infrastructures urbaines deviennent autonomes, connectées et capables d'optimiser durablement les ressources publiques.