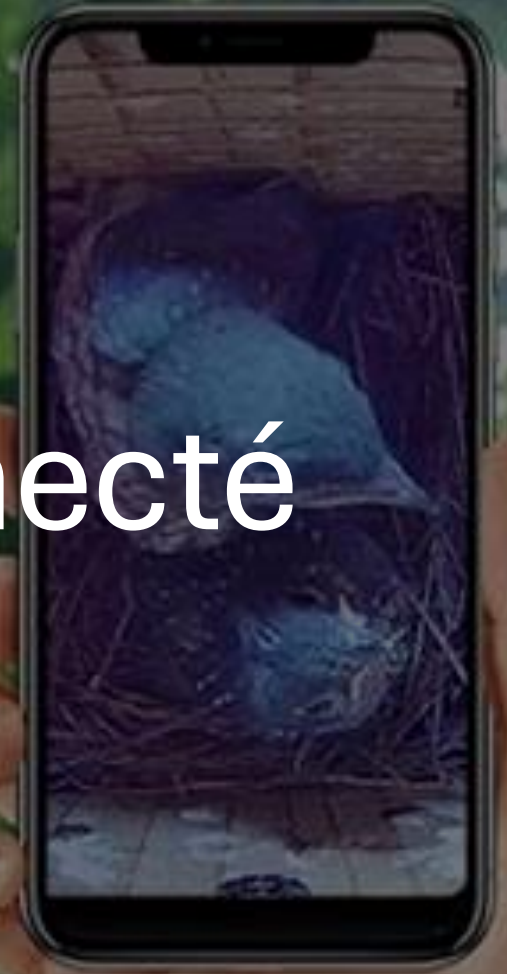


# Projet de nichoir connecté

SmartCities 2024-2025









## Analyse du marché

- Prix : 130 – 300 €
- Alimentation : Câblé, panneaux solaires
- Stockage : Carte SD, câblé, connexion WiFi cloud (privé), RSTP ...



# Nichoir Connecté - Proposition de Valeur

## **Objectif :**

Créer un nichoir intelligent et connecté qui permet de suivre la vie des oiseaux avec un coût accessible et une autonomie énergétique prolongée.

## **Caractéristiques principales :**

- **Prix :** Moins de 50 €
- **Autonomie énergétique :** 6 mois à 1 an sur batterie
- **Évolutif :** Possibilité d'ajouter un panneau solaire pour augmenter l'autonomie



Antenne

Capteur de présence PIR

ESPcam en  $\mu$ Python

Led IR

Nichoir



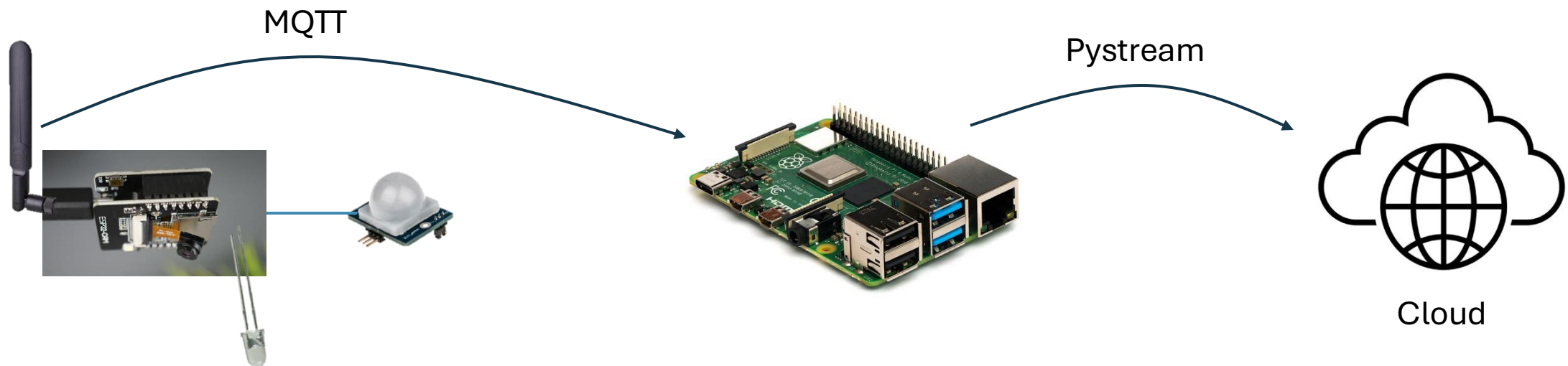
Raspberry Pi



Interface Web?  
Image dans le Cloud ?

# Objectifs

- **Surveiller l'activité dans un nichoir** via un système de détection de mouvement et de capture d'images.
- **Diminuer la consommation** pour une autonomie utile pour l'application.
- **Publier les données** avec une mise en page.



# Fonctionnement du système

## Détection de mouvement :

- Un capteur PIR détecte la présence d'oiseaux dans le nichoir.
- Lorsqu'un mouvement est détecté, l'ESP32-CAM capture une photo avec un éclairage IR.

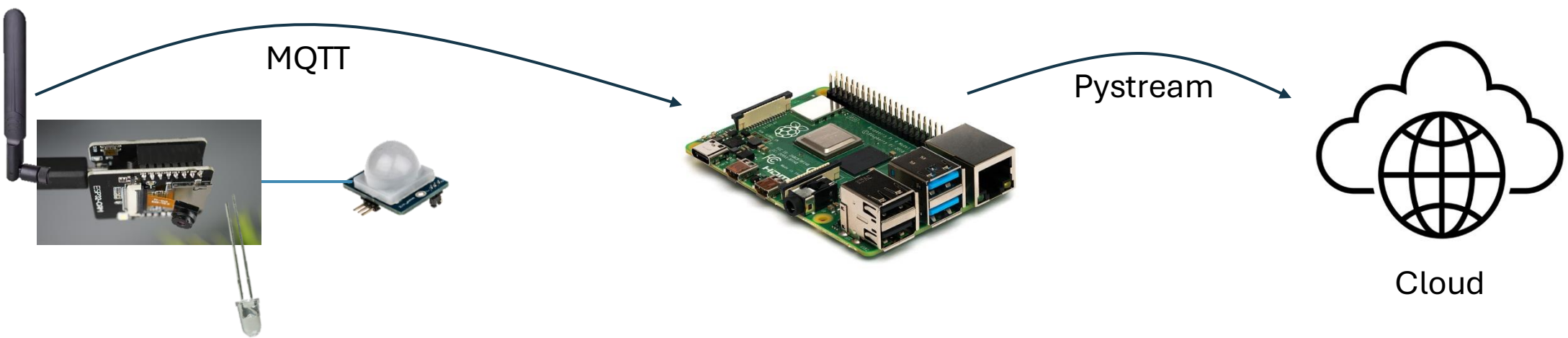
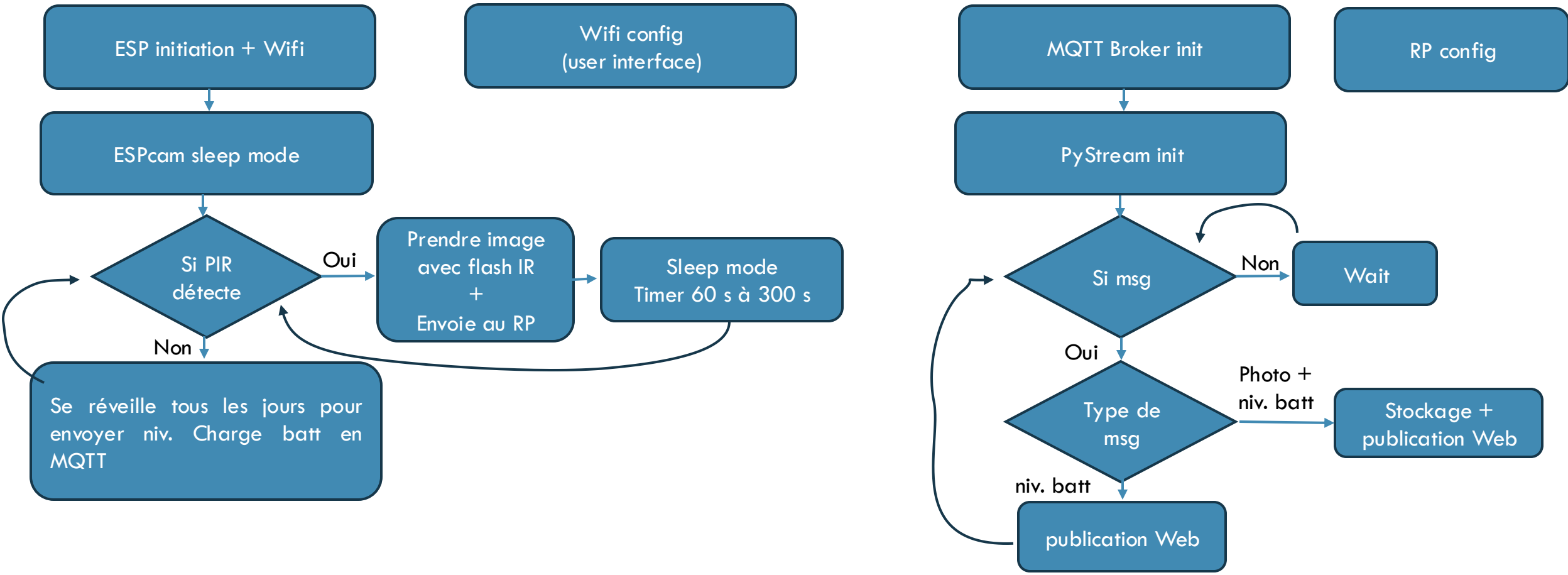
## Transmission des données :

- La photo est envoyée au **Raspberry Pi** via Wi-Fi pour traitement et stockage en MQTT.
- Le Raspberry Pi transmet ensuite les photos vers un serveur **cloud** pour l'accès à distance OU joue le rôle de serveur.

## Niveau de la batterie :

- **Mode standby** : En absence de mouvement dans le nichoir, le système se réveille une fois par jour et envoie le niveau de la batterie en MQTT.
- **Mode Présence** : Lorsque le capteur PIR détecte un mouvement, le niveau de la batterie est associé à chaque photo capturée.







## Projet 1 : Découverte

Projet pour découvrir le IoT, MQTT, RB Pi.

- Librairie  $\mu$ Python existante
- Connexions existantes
- Consommation trop importante pour l'application.



**ESP32-CAM**

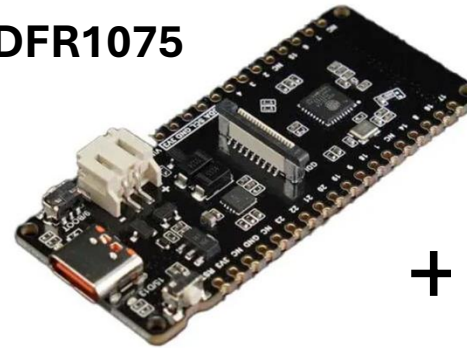


## Projet 2 : Optimisation

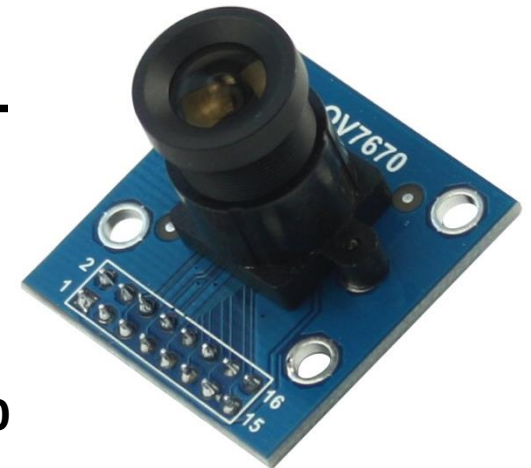
Projet pour optimiser la consommation d'un IoT  
+ Développer un module  $\mu$ Python.

- Librairie  $\mu$ Python à réaliser
- Connexions à réaliser
- Optimisation de la consommation pour l'application réelle (Deep sleep, gestion de l'énergie ...).

**DFR1075**



+



**OV7670**

