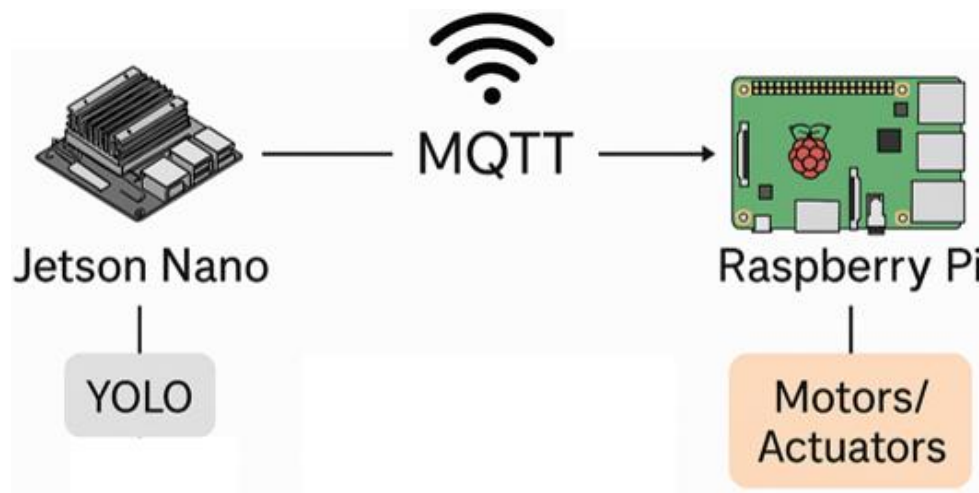




Communication Jetson ↔ Raspberry Pi via MQTT



1. Pourquoi MQTT ?

- **Problème** : Le Jetson (YOLO) et le Raspberry Pi (ROS + capteurs/actionneurs) doivent échanger des données **sans bloquer ROS**.
- **Solution** : MQTT (*Message Queuing Telemetry Transport*) est un protocole **léger** et **asynchrone**, idéal pour :
 - Envoyer les détections YOLO (Jetson → Pi). La jetson publie sur un topic.
 - Transmettre les commandes moteurs (Pi → Jetson). Le raspberry s'abonne sur ce topic.
 - Éviter les conflits avec les topics ROS.

```

Jetson Nano (YOLO + MQTT Publisher)
|
└─[WiFi/Ethernet]─> Raspberry Pi (MQTT Broker + Subscriber + GPIO Moteur)
                    |
                    └─ L298N → Moteur CC
  
```

2. Installation et Configuration

```

# Met à jour les paquets
sudo apt update

# Installe le serveur MQTT et ses clients
sudo apt install mosquitto mosquitto-clients -y

# Installe les bibliothèques Python nécessaires
pip install ultralytics opencv-python paho-mqtt

# Démarre le serveur MQTT
sudo systemctl start mosquitto

# Test de communication :
# Sur le Jetson (publier un test)
mosquitto_pub -h localhost -t "test" -m "Hello Raspberry"

# Sur le Raspberry Pi (s'abonner)
mosquitto_sub -h [IP_Jetson] -t "test"
  
```



3. Code python pour le suscriber et le publisher

La **Jetson** embarque actuellement le modèle **YOLO** pour détecter en temps réel les **chats** à partir d'un flux vidéo (par exemple, celui d'une caméra). Lorsqu'un chat est détecté, la Jetson envoie un **signal MQTT** sur un **topic spécifique**.

Le **Raspberry Pi**, qui est abonné à ce topic, reçoit ces messages. En fonction de l'information reçue (chat détecté ou non), il **active ou désactive le moteur d'un jouet**, déclenchant ainsi une réaction physique à la détection visuelle.

Ce système répartit donc les tâches entre deux plateformes :

- la **Jetson** pour le traitement d'image (détection)
- le **Raspberry Pi** pour le pilotage moteur (action)

Évolution prévue :

L'année prochaine, nous prévoyons de **centraliser tout le système sur la Jetson**. Elle assurera à la fois la détection des chats **et** le contrôle direct du moteur, ce qui simplifiera l'architecture et réduira la dépendance à une seconde carte (le Raspberry Pi).

Vous trouverez tous les codes sur le [github](#).