

## TP2 partie 2 : Flux NiFi

### Énoncé

Ce TP est à réaliser avec le même groupe que celui du TP 2.1.

Dans l'architecture Docker réalisée lors de la partie 1, écrire le fichier de flux NiFi permettant de traiter les trames télématiques dans le but de les envoyer au serveur REST. Ce flux doit lire les données depuis le broker EMQX mais aussi simuler les trames des deux protocoles permettant de faciliter le développement du flux de traitement.

Il a été relevé que certains véhicules envoyaient des trames corrompues avec des VIN incorrects. Pour notre système de télématique toulousains, un VIN est considéré incorrect dans les conditions suivantes :

- Il ne commence pas par « VF7 »
- Il ne fait pas 17 caractères

Il est imposé de séparer le flux en deux groupes de processeurs (au minimum) :

- Un groupe de processeurs d'extraction de données pour
  - Lire les messages MQTT depuis le docker EMQX
  - Simuler une trame du protocole 1 (fréquence 1 seconde)
  - Simuler une trame du protocole 2 (fréquence 1 seconde)
- Un groupe de processeurs de transformation de données pour
  - Extraire les données nécessaires des trames des deux protocoles de communication présentées dans le CDC
  - Filtrer les trames corrompues et publier les dans un message de log niveau WARN.
  - Mettre en place un système de cache pour ne transmettre les données par VIN que toutes les 10 secondes (optionnel).
  - Produire le json qui sera le corps de la requête REST POST présentée dans le CDC
  - Envoyer le json au serveur REST (serveur qui sera réalisé dans le TP suivant)
  - Publier ce json dans un message de log niveau INFO.

Le fichier README.md doit contenir en plus :

- Un paragraphe de conseil sur la sécurité du système et les bonnes pratiques.
- Un paragraphe sur les moyens employés pour valider le bon filtrage des trames corrompues.

### Données d'entrées :

- Le simulateur de trames du TP1
- L'architecture Docker du TP2.1
- Le fichier flow.xml.gz fourni sur moodle
- Cahier des charges disponible sur moodle (Cloud\_2024\_2\_CDC\_Exercice.pdf).

### Objectif

- Écrire le fichier de flux NiFi