



# Présentation GT-EPR : Orchestration et Optimisation de vols de drones UAV sur des réseaux 5G

Présenté par :

Samir SI-MOHAMMED  
ENS De Lyon - LIP - Stackeo  
[samir.si-mohammed@ens-lyon.fr](mailto:samir.si-mohammed@ens-lyon.fr)

Encadré par :

Pr Adlen KSENTINI (EURECOM)  
Pr Yacine CHALLAL (ESI)  
Pr Amar BALLA (ESI)



# Sommaire

---

- **Contexte**
- **Concepts fondamentaux**
- **Orchestration de vols**
- **Optimisation de vols**
- **Synthèse**

# Contexte



- **Objectifs**

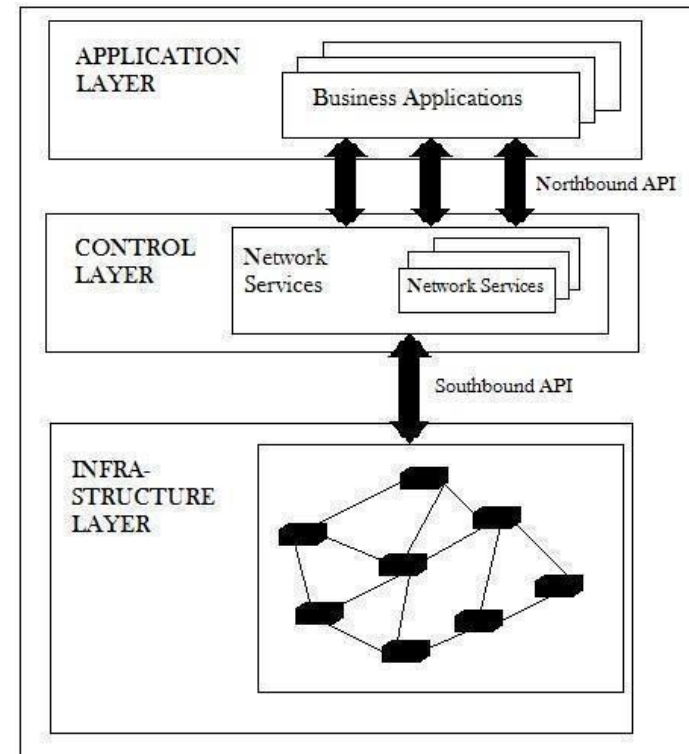
- 1. Développement d'un contrôleur pour l'orchestration de vols de drones sur des réseaux 5G.**
- 2. Proposition de solutions pour l'optimisation de vols de drones sur des réseaux 5G.**

---

# Concepts Fondamentaux

# Software Defined Networking

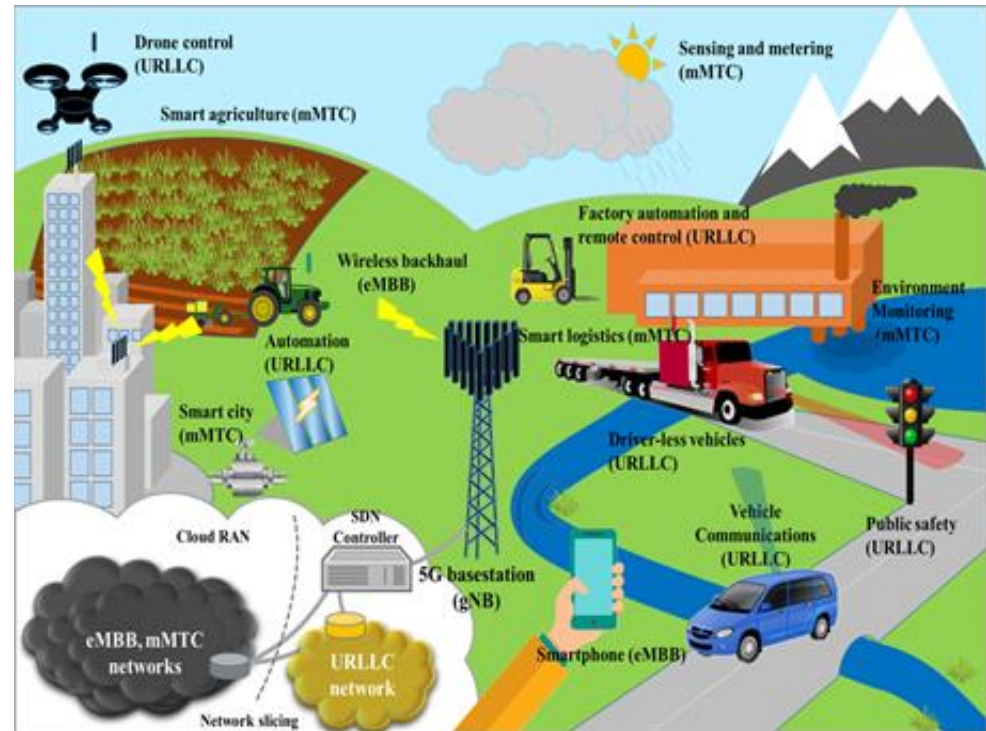
- Simplifie la gestion du réseau, introduit la programmabilité en centralisant logiquement l'intelligence du réseau.
- Sépare les éléments qui composent un réseau (tels que le flux de données et le flux de contrôle)
  - Couche contrôle : Slice Orchestrator



Architecture SDN (Tijare et Vasudevan, 2016)

# Network Slicing

- Assure la personnalisation des services et l'isolation dans une infrastructure physique, en permettant la séparation logique et physique des ressources.
- Un Slice réseau est un réseau virtuel avec un but bien précis.
  - Application : Vols de drones.



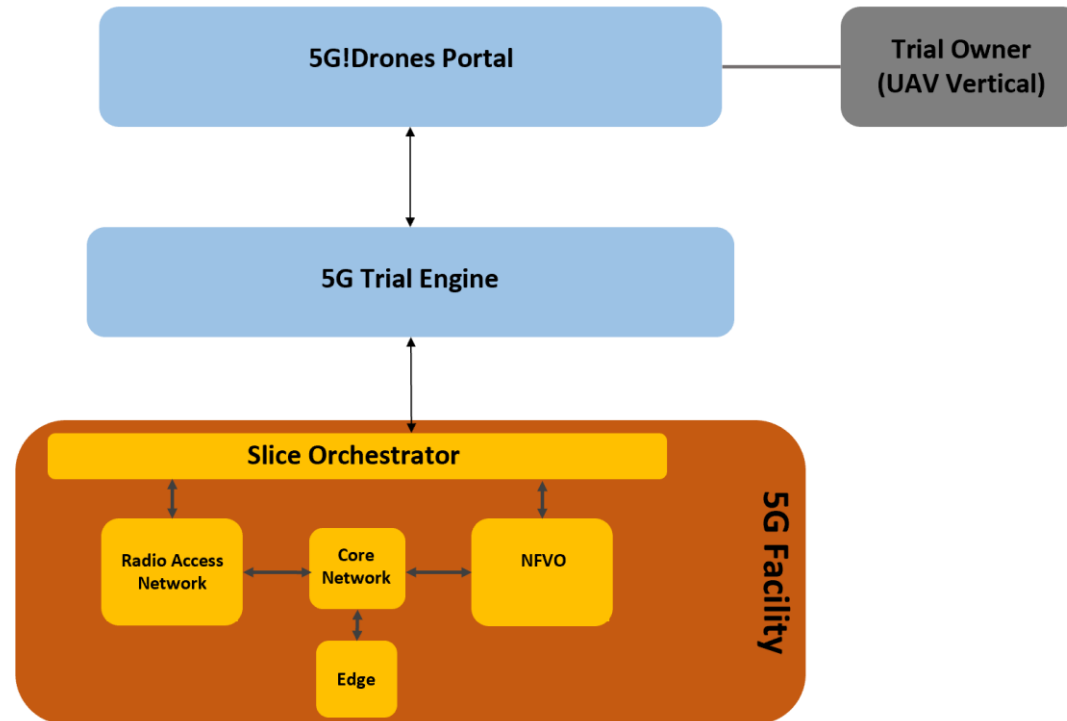
Catégories de Service 5G (Ji et al., 2018)

---

# Orchestration de vols

# Slice Orchestrator

- Slice Orchestrator servant de vis-à-vis entre 5GEVE et le système 5G!Drones.



Architecture du système UAV 5G!Drones



# 1. Slice Orchestrator

---

- **Besoins fonctionnels**

**Le Slice Orchestrator doit permettre de :**

- **Créer, modifier, supprimer et redéployer des Slices réseau.**
- **Lister les Slices réseau créés.**
- **Afficher les temps de création et de suppression de Slices réseau.**
- **Rediriger l'utilisateur vers une page de suivi de performances (KPI Monitoring) d'un Slice.**

# Slice Orchestrator

- Ensemble des méthodes développées pour l'API Nord de 5GEVE (Swagger).

## Slice Orchestrator API

This interface provides the offered methods of our Slice Orchestrator API, their parameters and their responses

### Slices

Show/Hide | List Operations | Expand Operations

GET	/createSlice/{sliceId}	Create a Slice that will be instantiated in our platform
POST	/deleteSlice	Delete a Slice
GET	/getSlice/{sliceId}	Retrieve a Slice from its ID
GET	/listSlices	List the available Slices
GET	/resumeSlice/{sliceId}	resume a Slice that will be instantiated in our platform
POST	/stopSlice	Stop a Slice
POST	/updateRunningSlice	Update a running Slice
POST	/updateStoppedSlice	Update a stopped Slice
POST	/validateResources	Validate the resources needed

### KPIs

Show/Hide | List Operations | Expand Operations

GET	/getDashboardKpi/{userId}	Get the Dashboard for KPI monitoring of a Slice
GET	/getDecommissioningTime/{sliceId}	Get the Decommissioning time of a Slice
GET	/getDeploymentTime/{sliceId}	Get the Deployment time of a Slice

### Trial

Show/Hide | List Operations | Expand Operations

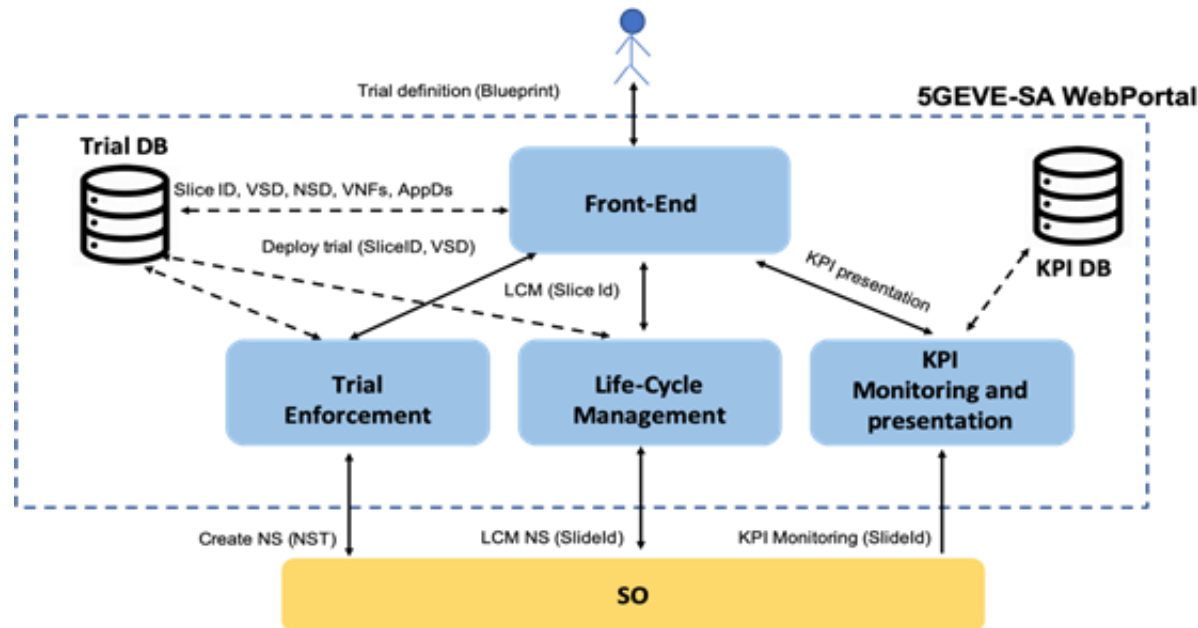
POST	/validateTrial	Validate the trial
------	----------------	--------------------

## API Nord du Slice Orchestrator

# Portail Web

- **Rôles du Portail Web :**

- Transmission des demandes de gestion de Slices au Slice Orchestrator.
- Collecte des mesures KPIs (Key Performance Indicators).



**Architecture du Portail Web**

# Portail Web

- Introduction d'informations à travers des formulaires sur le Portail Web (Flask).
- Informations réparties en Métadonnées, Radio et Cloud.

**Slice Info**

Name  
Test Slice 1

Provider  
EURECOM

Type  
☒ uRLLC  
☐ eMBB  
☐ Hybrid

Version  
1

Start Date Time  
06/17/2020

End Date Time  
06/17/2020

Start Hour  
09:59

End Hour  
09:59

KPI List  
☒ Service Deployment Duration  
☒ Reconfiguration Execution Duration

Next

**RAN Info**

Region  
☒ 5GEVE

Ue List  
☒ IMSI 208950000000007

Add UE Remove UE

Latency (ms)  
10.00

KPI List  
☒ Latency  
☐ Packet Delay Variation  
☐ Uplink Date Rate  
☐ Downlink Date Rate  
☒ Packet Loss Rate  
☒ Bandwidth  
☐ Interruption Time  
☐ IP Rate

Next

**Identifiant Equipement  
Utilisateur**

# Portail Web

**NSD Info**

Name

Version

Checksum

Application Name

Application Description

Virtual Memory Size

CPU Architecture

Number of Virtual CPU

Container Format

Min Disk number

Min RAM Size

Size

**Lien de téléchargement d'application**

Software Image URL

Operating System

Storage Size

**Nom DNS pour l'accès à l'application**

DNS Domain Name

IP Address Type

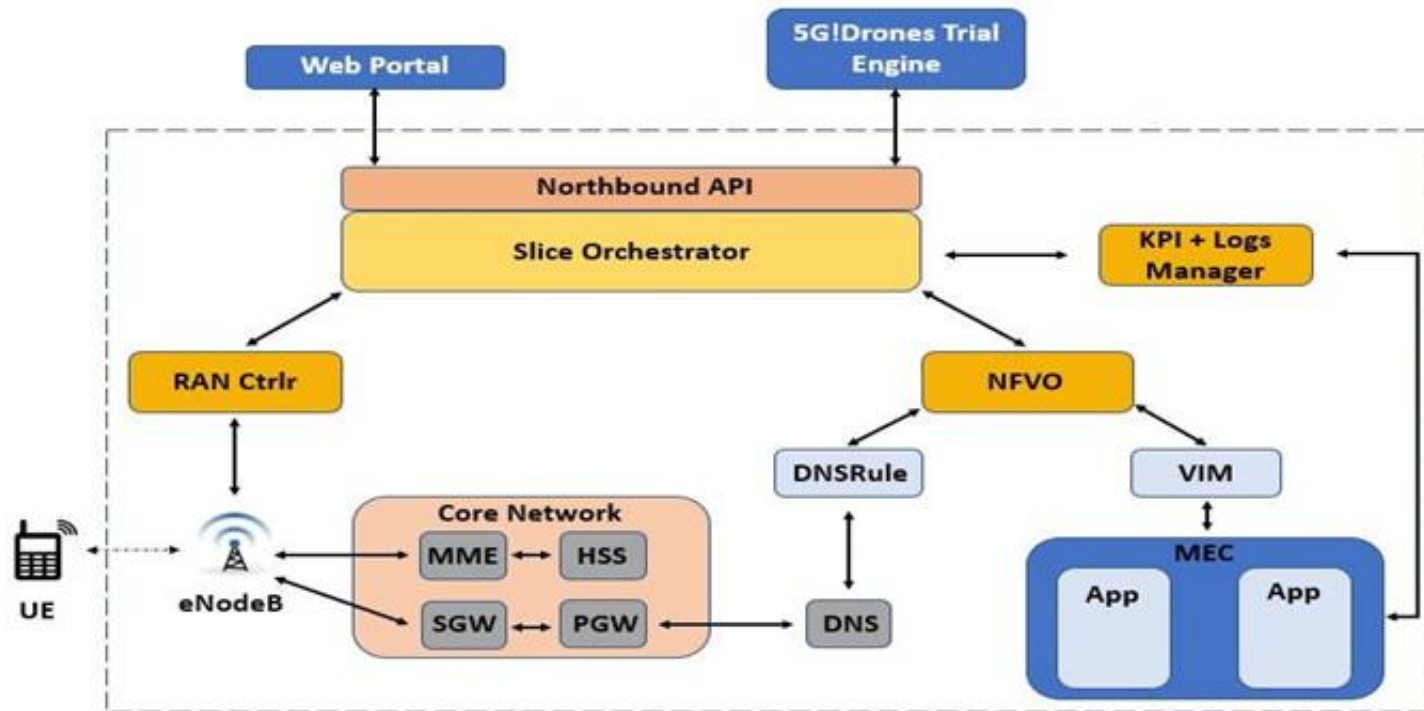
TTL

Min RAM Size

Time Unit

## Formulaire NSD

# Portail Web



Architecture 5GEVE-Sophia Antipolis

- Intégration du Slice Orchestrator et Portail Web dans 5GEVE.

# Portail Web

- Ouverture d'un socket sur l'application Serveur sous le nom DNS.

```
HOST = 'gps-server.eurecom.mec' # The server's hostname or IP address
#HOST = '172.29.248.16'
#HOST = '192.168.12.167'
PORT = 65432 # The port used by the server
```

## Socket HOST

- Transition du trafic depuis le client vers le Serveur.

log	
> Connected by ('172.29.248.222', 51384)	
> Received b"{'altitude': 130.05, 'speed': 0.032, 'longitude': 7.07, 'climb': 0.02, 'latitude': 43.61}"	Q Q
> Received b"{'altitude': 130.12, 'speed': 0.036000000000000004, 'longitude': 7.19, 'climb': 0.03, 'latitude': 43.93}"	
> Received b"{'altitude': 130.19, 'speed': 0.040000000000000001, 'longitude': 7.3100000000000005, 'climb': 0.04, 'latitude': 44.25}"	
> Received b"{'altitude': 130.26, 'speed': 0.044000000000000001, 'longitude': 7.4300000000000001, 'climb': 0.05, 'latitude': 44.57}"	
> Received b"{'altitude': 130.32999999999998, 'speed': 0.0480000000000000015, 'longitude': 7.5500000000000001, 'climb': 0.060000000000000005, 'latitude': 44.89}"	

## Logs Application GPS

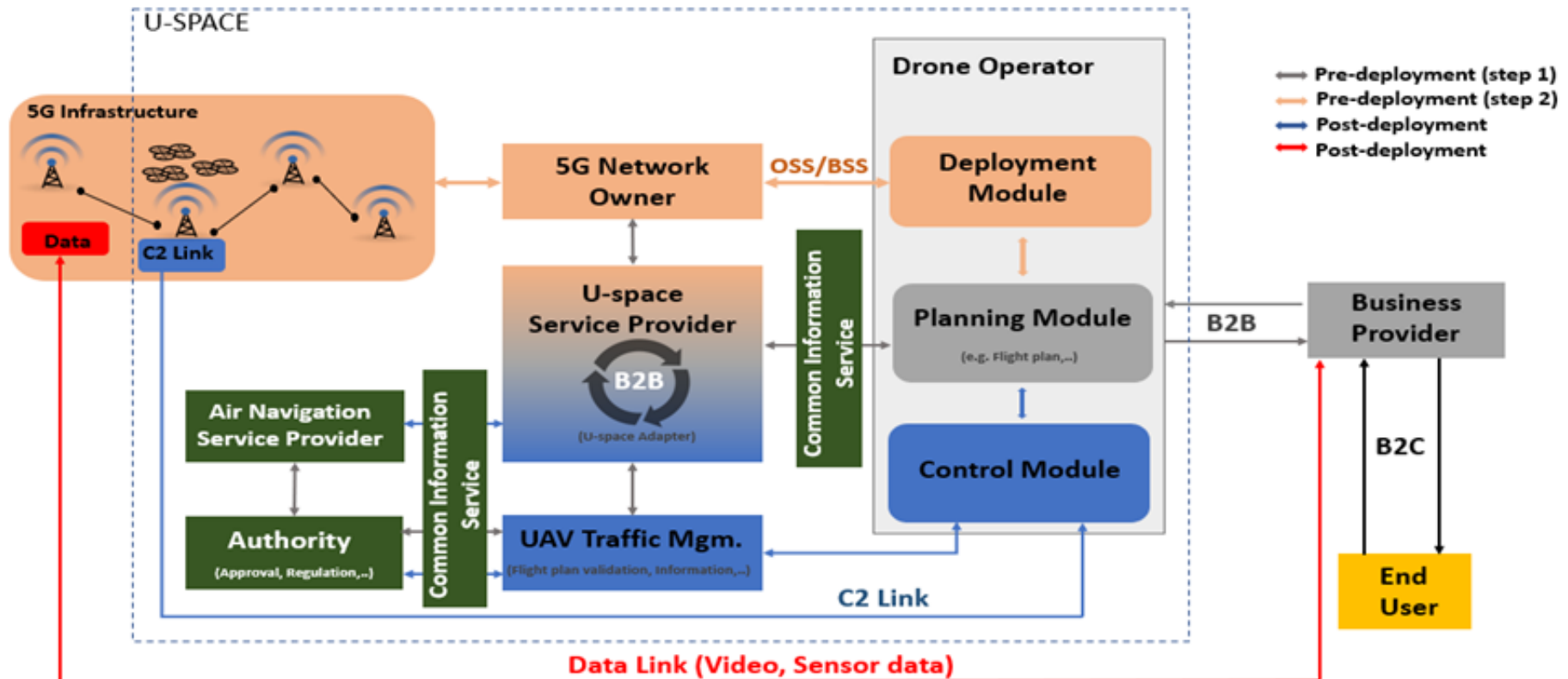
---

# Optimisation de vols



# Optimisation d'architecture

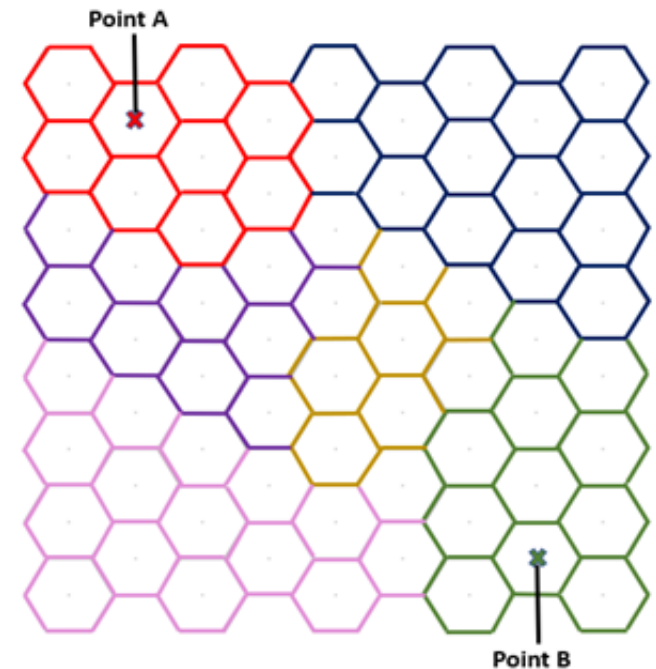
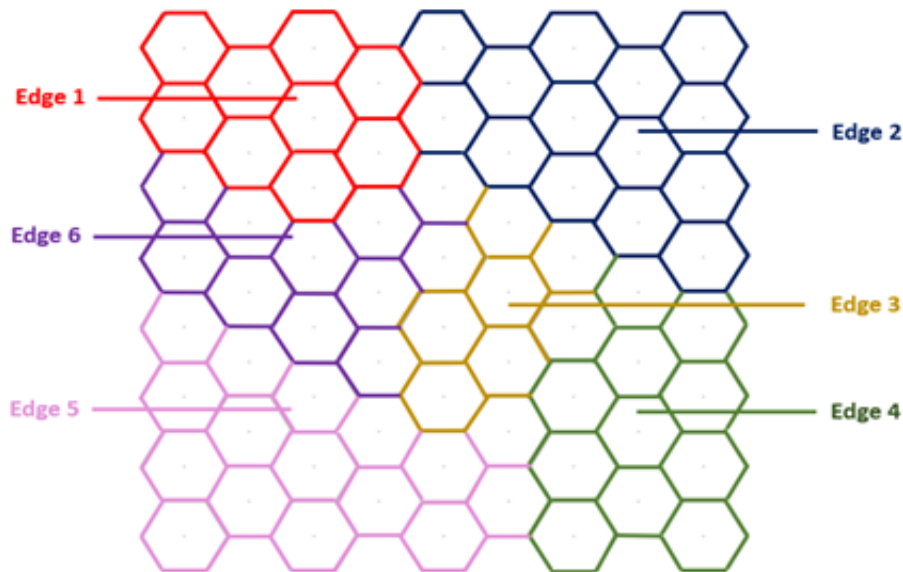
- Architecture permettant la planification et gestion de vols de drones sur des réseaux 5G et faisant intervenir l'ensemble des parties prenantes.



Architecture de vol

# Optimisation de plan de vols

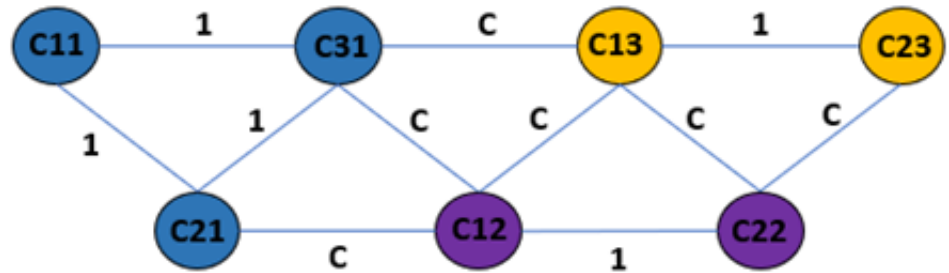
- Modélisation du réseau en forme hexagonale, où chaque groupe de régions est couvert par un Serveur d'Application.



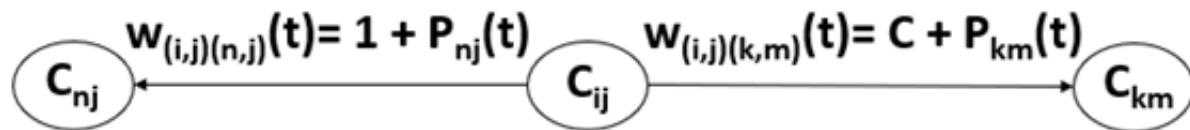
Topologie du réseau

# Optimisation de plan de vols

- Modélisation sous forme de graphe où les nœuds sont les régions, et les arêtes la distance entre elles.
- Distance entre nœuds égale à 1 s'ils sont sous la couverture du même Serveur, C sinon.
- Probabilité de surcharge de la région destination ajoutée aux poids.
- Paramètres dotés de coefficients afin de pouvoir diriger l'algorithme.



Structure du graphe



$$W_{(i,j)(k,m)} = \begin{cases} 1 + (1 - \alpha) P_{km}(t) & \text{si } j = m \\ C + \alpha P_{km}(t), & \text{sinon} \end{cases}$$

Poids des arêtes

- $0 < \alpha < 1$

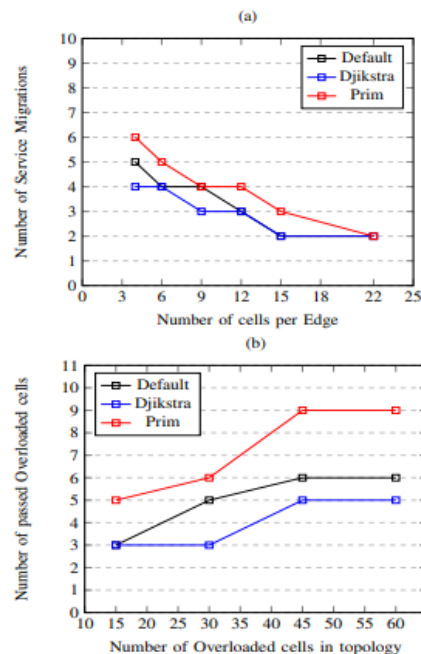


Problème du Plus Court Chemin

# Optimisation de vols

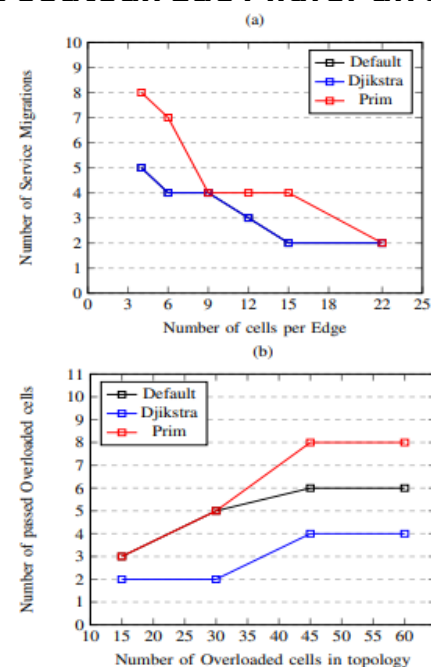
$$W_{(i,j)(k,m)} = \begin{cases} 1 + (1 - \alpha) P_{km}(t) & \text{si } j = m \\ C + \alpha P_{km}(t), & \text{sinon} \end{cases}$$

- **Scenario 1 : Plus de poids est donné à la Migration de Service, i.e. Migrer un Service est plus coûteux que traverser une cellule chargée.**



Résultats Scénario 1

- **Scenario 2 : Plus de poids est donné à la Surcharge des Cellules, i.e. Traverser une cellule chargée est plus coûteux que Migrer un Service.**



Résultats du Scénario 2

# Matériel utilisé



**Drone utilisé**



**Télécommande de drone**

# Matériel utilisé



Pixhawk 4 Autopilot



ARDU Pilot Mission Planner



# Matériel utilisé

```
1 from dronekit import *
2 import time
3 import json
4 from datetime import datetime, date
5
6 init = {}
7 with open('telemetry_info.json', 'w') as json_file:
8     json.dump(init, json_file)
9
10 telemetry_infos = []
11
12 def fill_info(info):
13     with open('telemetry_info.json', 'r+') as file:
14         data = json.load(file)
15         data.update(info)
16         file.seek(0)
17         json.dump(data, file, sort_keys=True, indent=2)
18
19
20 #vehicle = connect('127.0.0.1:14550', wait_ready=True)
21 vehicle = connect('127.0.0.1:14550', wait_ready=False)
22
23 #vehicle.mode = VehicleMode("MISSION")
24 while (True):
25     # vehicle is an instance of the Vehicle class
26     print ("Autopilot Firmware version: ", vehicle.version)
27     #print ("Autopilot capabilities (supports ftp): ", vehicle.capabilities.ftp)
28     print ("Global Location: ", vehicle.location.global_frame)
29     print ("Global Location (relative altitude): ", vehicle.location.global_relative_frame)
30     print ("Local Location: ", vehicle.location.local_frame) #NED
31     print ("Attitude: ", vehicle.attitude)
32     print ("Velocity: ", vehicle.velocity)
33     print ("GPS: ", vehicle.gps_0)
34     print ("Groundspeed: ", vehicle.groundspeed)
35     print ("Airspeed: ", vehicle.airspeed)
36     print ("Gimbal status: ", vehicle.gimbal)
37     print ("Battery: ", vehicle.battery)
38     print ("EKF OK?: ", vehicle.ekf_ok)
39     print ("Last Heartbeat: ", vehicle.last_heartbeat)
40     print ("Rangefinder: ", vehicle.rangefinder)
41     print ("Rangefinder distance: ", vehicle.rangefinder.distance)
42     print ("Rangefinder voltage: ", vehicle.rangefinder.voltage)
43     print ("Heading: ", vehicle.heading)
44     print ("Is Armable?: ", vehicle.is_armable)
45     print ("System status: ", vehicle.system_status.state)
46     print ("Mode: ", vehicle.mode.name) # settable
47     print ("Armed: ", vehicle.armed) # settable
```

## Application de télémétrie Dronekit

# Matériel utilisé



Mini-PC



Configuration Mini-PC



---

# Synthèse

# Perspectives

---

- **Opérations de Data Mining sur les KPI en plus de leur collecte.**
- **Gestion de l'aspect sécurité et isolation des Slices lors de la création, et au cours de l'exécution.**
- **Lien entre le Portail Web et l'UTM pour effectuer une validation synchronisée des plans de vol.**

# Conclusion

---

- **Résultats ayant rendu la plateforme 5GEVE utilisable par les partenaires du projet 5G!Drones.**
- **Architecture adoptée par les partenaires du projet : Gain en temps de négociations, grâce à la concordance des vues des différents métiers (Réseau et Aviation).**
- **Optimisation des vols de drones en termes de Relocations de Service, et accroissement du contrôle des pilotes durant les missions critiques.**

# Publications issues du PFE

---

- **Publication acceptée à une Conférence Internationale IEEE GLOBECOM 2020 :** ‘Samir Si-Mohammed, Adlen Ksentini, Maha Bouaziz, Yacine Challal et Amar Balla. « *UAV mission optimization in 5G: On reducing MEC service relocation* », IEEE Global Communications Conference, Taipei, Taiwa, December 2020’.
- **Publication soumise à un Journal International IEEE Vehicular Magazine :** ‘Samir Si-Mohammed, Maha Bouaziz, Hamed Hellaoui, Oussama Bekkouche, Adlen Ksentini, Tarik Taleb, Lechoslaw Tomaszewski, Thomas Lutz, Gokul Srinivasan, Tanel Jarvet et Pawel Montowtt. « *Supporting UAV Services in 5G Networks: New architecture integrating 5G with U-space* », IEEE Vehicular Magazine’.

---

# Merci pour votre attention !