



Soutenance d'Ingénieur : Conception et Développement d'une API abstraite pour l'orchestration de vols de Drones sur un Réseau 5G



Présentée par : Devant le jury composé de : Encadré par :

Samir
SI-MOHAMMED

Mr Sofiane BATATA
Mr Abdelkrim CHEBIEB
Mme Soumia ZELLAGUI

Mr Adlen KSENTINI
Mr Yacine CHALLAL
Mr Amar BALLA

Sommaire

- **Contexte**
 - Objectifs
- **Concepts fondamentaux**
- **Conception**
 - Slice Orchestrator
 - Portail Web
 - Optimisation d'architecture
 - Optimisation de vols
- **Réalisation**
 - Slice Orchestrator
 - Portail Web
 - Optimisation de vols
 - Matériel utilisé
- **Conclusion**
- **Synthèse**
 - Limites et Perspectives
 - Contributions et Résultats
 - Publications issues du PFE

Contexte



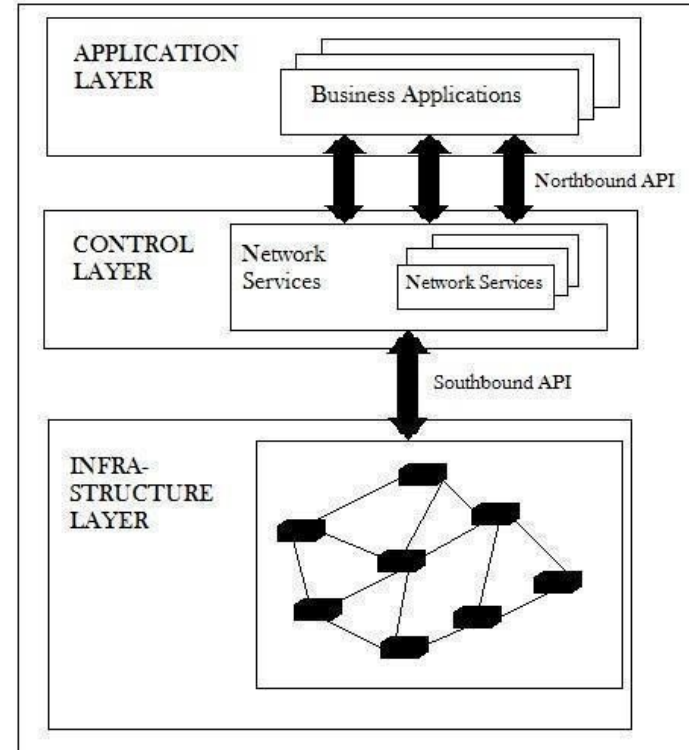
- **Objectifs**

- 1. Développement d'un Slice Orchestrator (SO) pour la gestion des Slices.**
- 2. Développement d'un Portail Web pour l'utilisation du SO.**
- 3. Proposition d'une architecture pour la planification de vols de drones.**
- 4. Proposition et implémentation d'un algorithme pour l'optimisation de vols de drones.**

Concepts Fondamentaux

Software Defined Networking

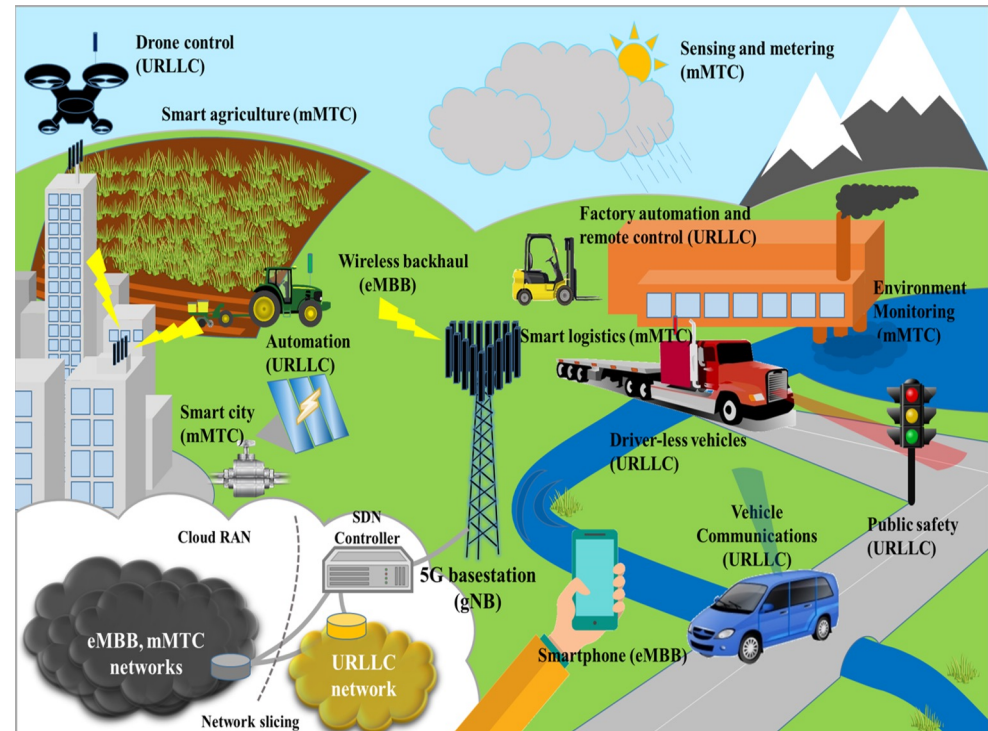
- Simplifie la gestion du réseau, introduit la programmabilité en centralisant logiquement l'intelligence du réseau.
- Sépare les éléments qui composent un réseau (tels que le flux de données et le flux de contrôle)
 - Couche contrôle : Slice Orchestrator



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)

Solutions Open Source d'orchestration

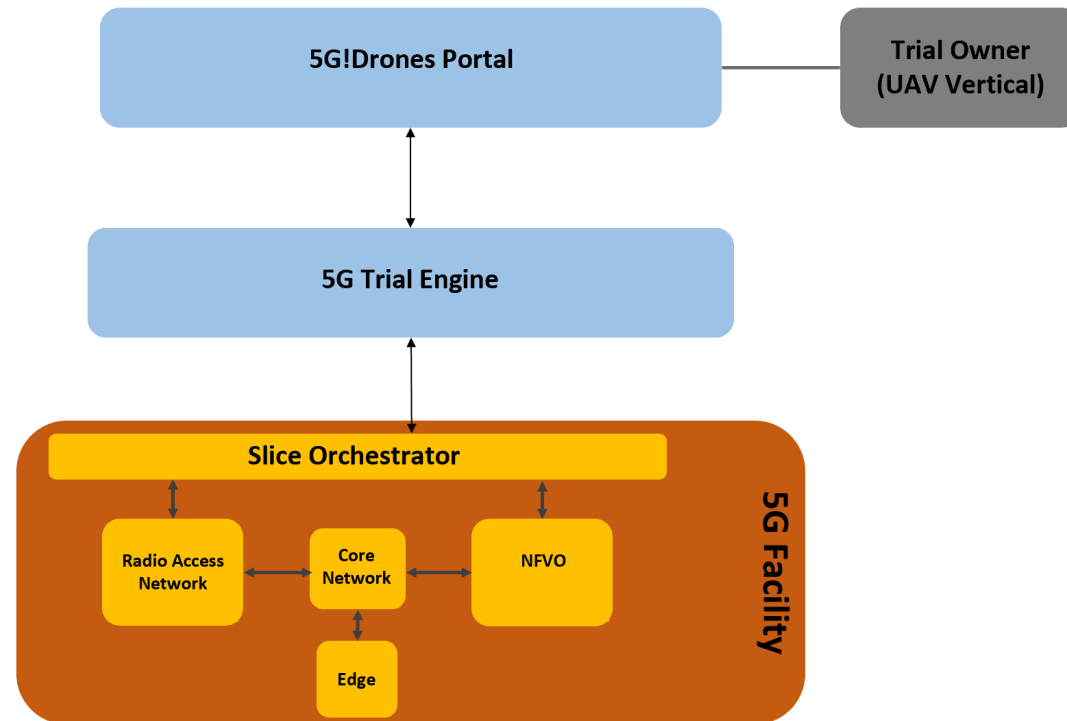
| Solution | Leader | Domaine | | | | Interface | | |
|-----------------------------------|---------------------|---------|-----|-----|--------|-----------|-----|-----|
| | | Cloud | SDN | NFV | Legacy | CLI | API | GUI |
| ONAP (ONAP, 2017) | Linux Foundation | X | | X | | X | X | X |
| Gohan (NTT, 2015) | NTT Data | X | X | X | X | X | X | X |
| OSM (ETSI) | ETSI | X | X | | X | X | X | X |
| Cloudify (GigaSpaces, 2015) | GigaSpace | X | | X | | X | X | X |

Solutions Open Source d'orchestration réseau (De Sousa et al., 2019)

Conception

1. 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**

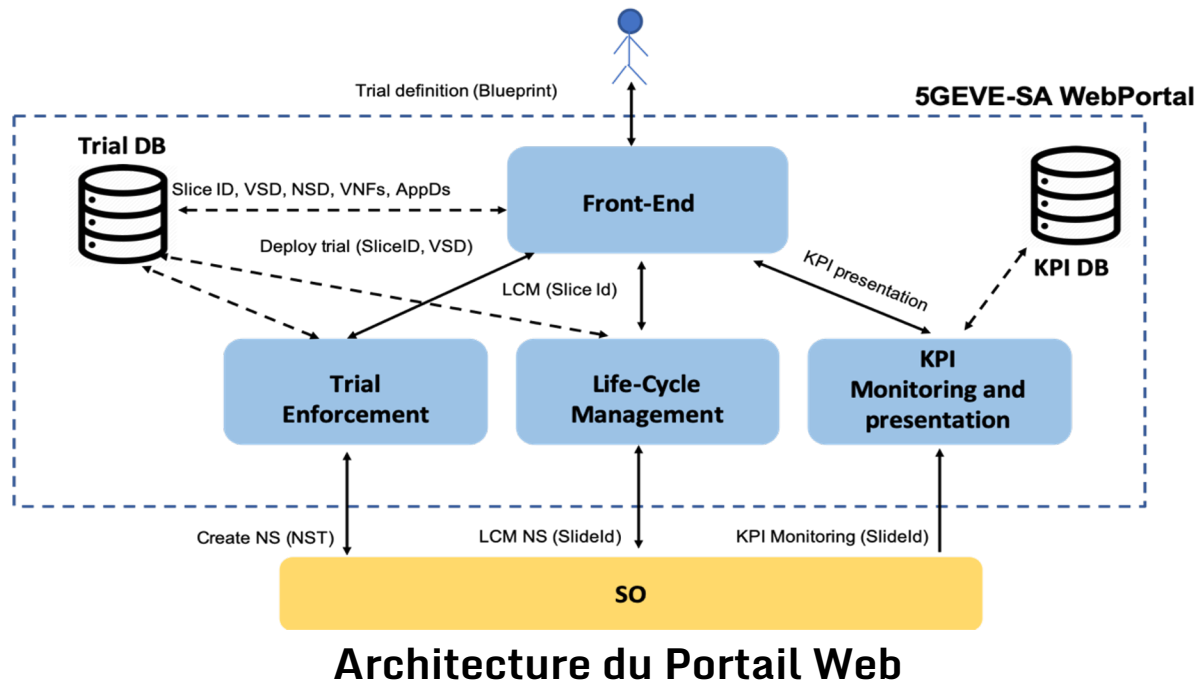
Le Slice Orchestrator doit permettre de :

- **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.**

2. Portail Web

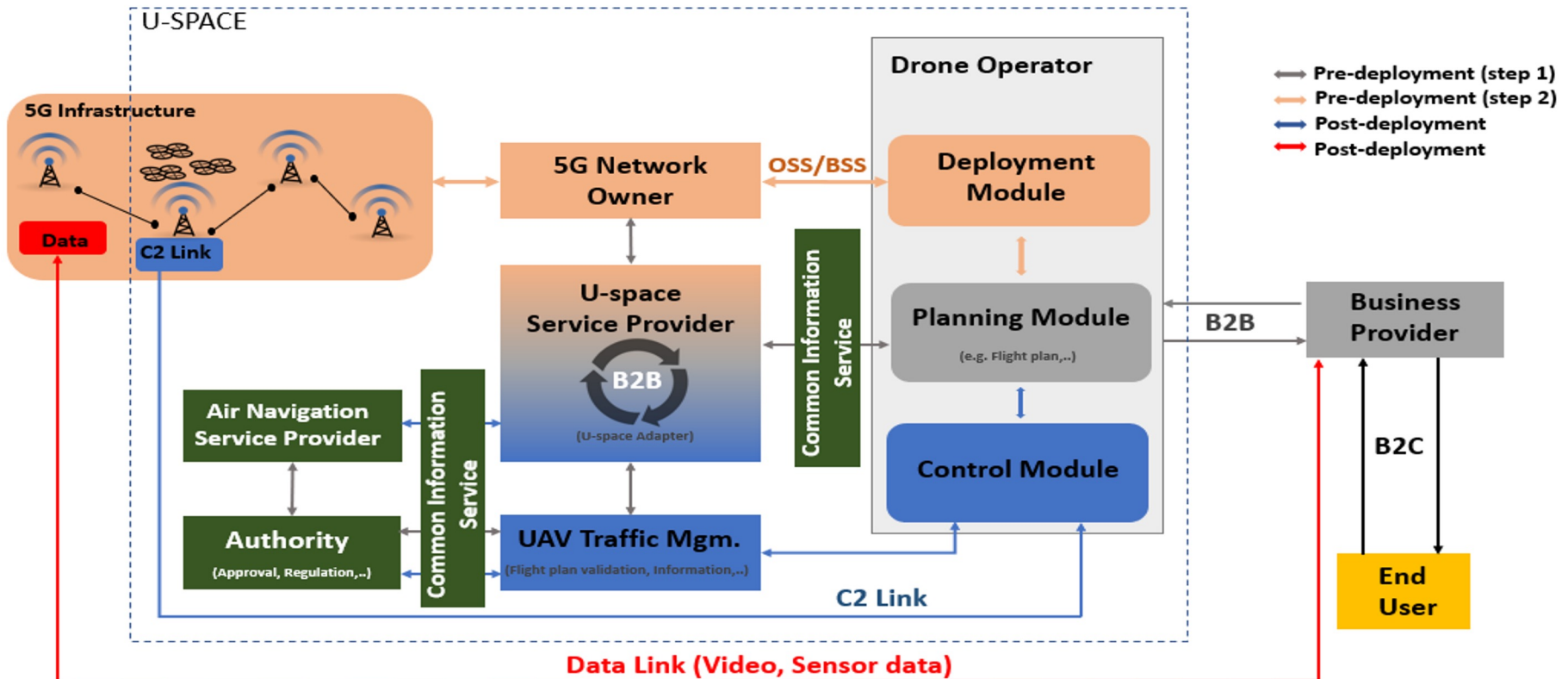
- **Rôles du Portail Web :**

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



3. Optimisation d'architecture

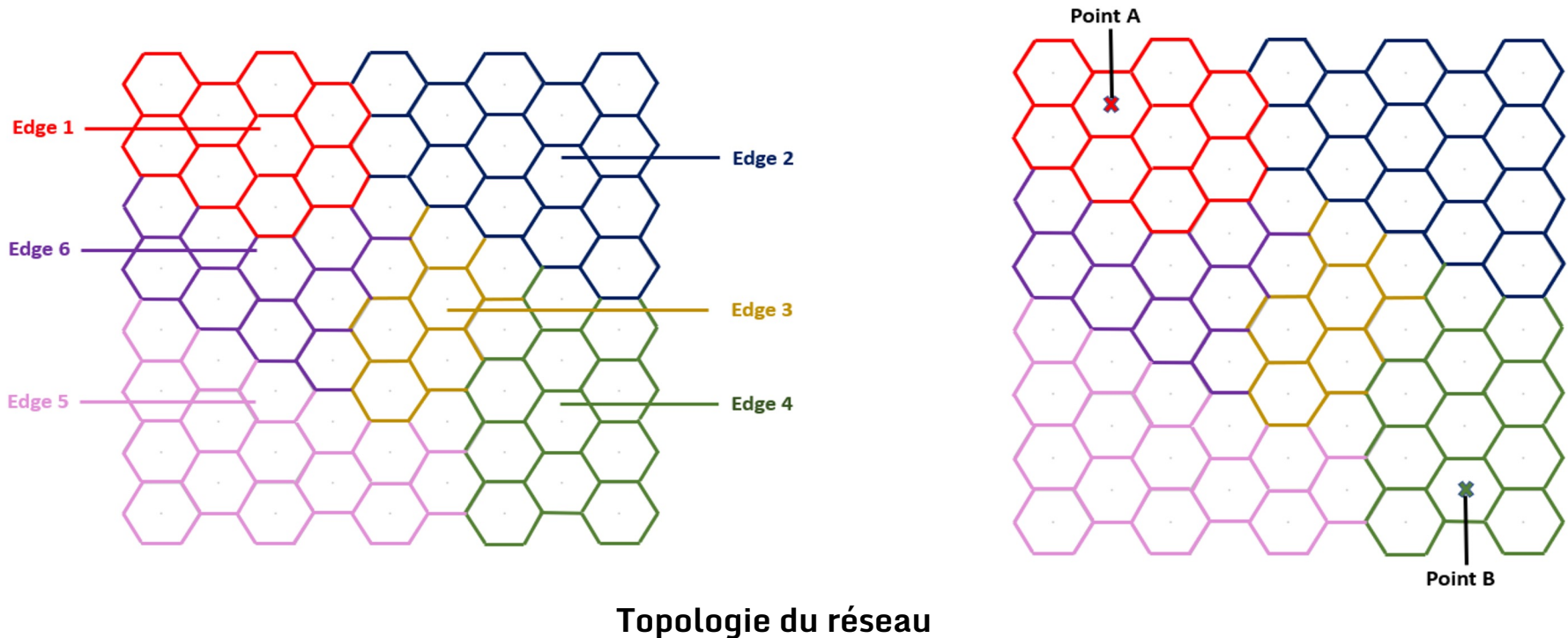
- Architecture permettant la planification et gestion de vols de drones sur des réseaux 5G et faisant intervenir l'ensemble des parties prenantes.



Architecture de vol

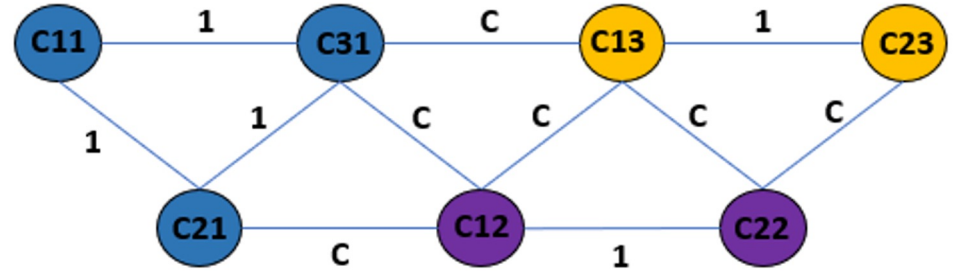
4. Optimisation de plan de vols

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

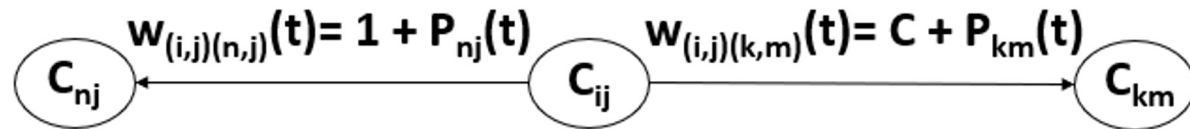


4. Optimisation de plan de vols

- 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

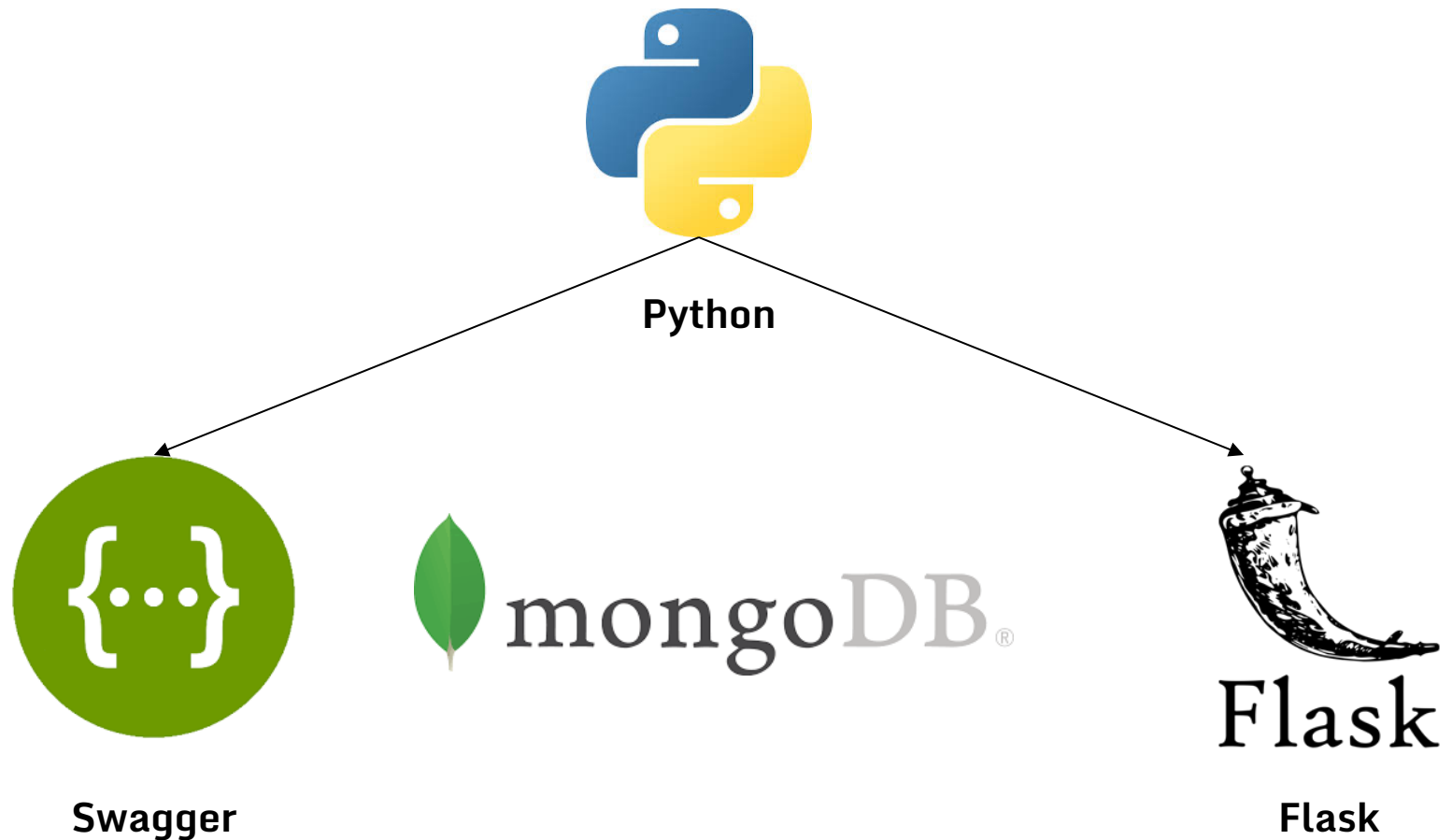
- $0 < \alpha < 1$



Problème du Plus Court Chemin

Réalisation

Environnements de développement



1. 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

2. 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

Identifiant Equipement Utilisateur

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

2. Portail Web

Slice successfully created

Test Slice 1

Provider : Eurecom

Type : URLLC

Region : EURECOM

Creation Time : 0:00:16.204096

Running

Stop Slice

KPI Monitoring

— Succès de la création d'un Slice : Proposition d'arrêt ou de consultation des KPIs.

Test Slice 1

Provider : Eurecom

Type : URLLC

Region : EURECOM

Deletion Time : 0:00:04.047821

Deleted

Redeploy Slice

— Succès de la suppression d'un Slice : Proposition de redéploiement.

Slice successfully stopped

Test Slice 1

Provider : Eurecom

Type : URLLC

Region : EURECOM

Creation Time : 0:00:16.204096

Stopped

Resume Slice

Delete Slice

— Succès de l'arrêt d'un Slice : Proposition de reprise ou de suppression.

Gestion des Slices dans le Portail Web

2. Portail Web

NSD Info

Name
gps-server

Version
1

Checksum
132

Application Name
gps-server

Application Description
gps

Virtual Memory Size
120.00

CPU Architecture
x86_64

Number of Virtual CPU
500.00

Container Format
kubernetes

Min Disk number
1.00

Min RAM Size
512.00

Size
512.00

Lien de téléchargement d'application

Software Image URL
http://localhost:9000/download

Operating System
linux

Storage Size
1.00

Nom DNS pour l'accès à l'application

DNS Domain Name
gps-server.

IP Address Type
IP_V4

TTL
0.00

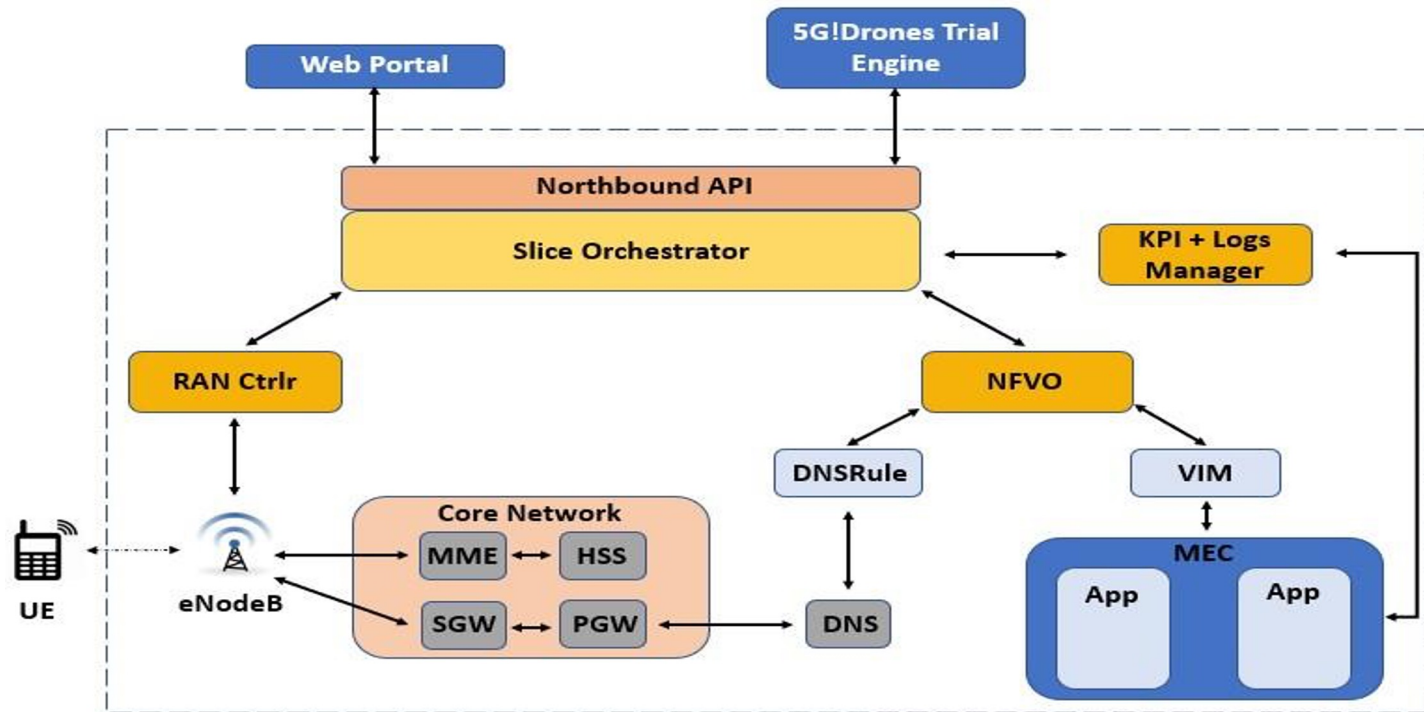
Min RAM Size
100.00

Time Unit
ms

Validate

Formulaire NSD

2. Portail Web



Architecture 5GEVE-Sophia Antipolis

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

2. 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}" |
| > 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': 0.048000000000000015, 'longitude': 7.5500000000000001, 'climb': 0.060000000000000005, 'latitude': 44.89}" |

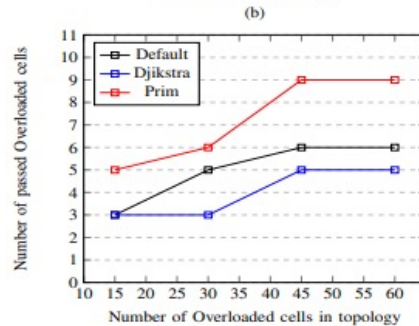
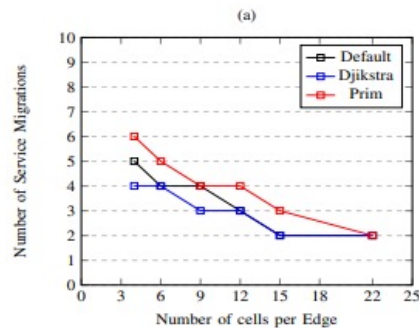
Logs Application GPS

3. Optimisation de vols

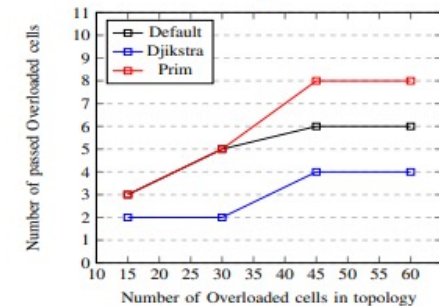
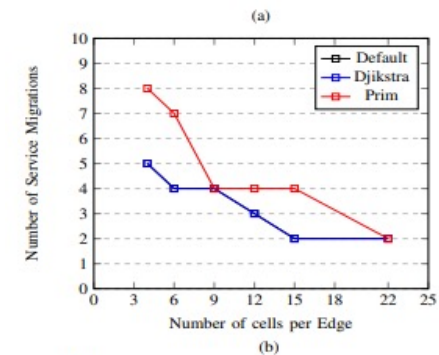
$$W_{(i,j)(k,m)} = \begin{cases} 1 + (1 - \alpha) P_{km}(t) & \text{si } j = m \\ C + \alpha P_{km}(t), & \text{sinon} \end{cases}$$

- **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.**

- **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 Scénario 1



Résultats du Scénario 2

4. Matériel utilisé



Drone utilisé

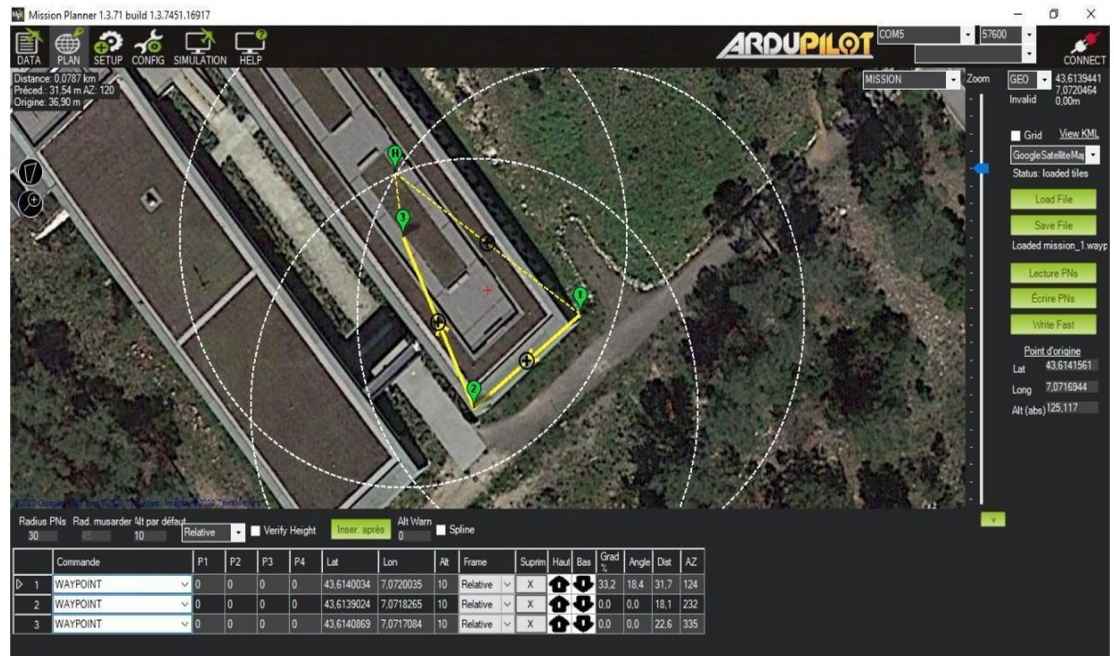


Télécommande de drone

4. Matériel utilisé

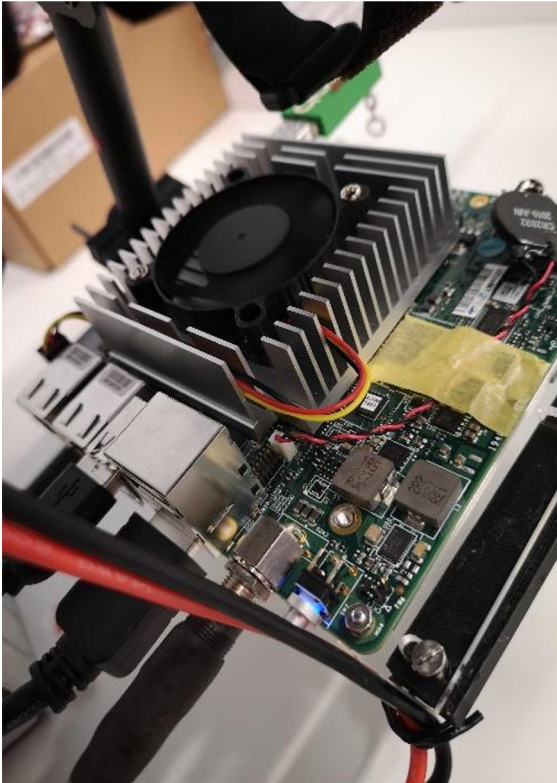


Pixhawk 4 Autopilot



ARDU Pilot Mission Planner

4. Matériel utilisé



Mini-PC



Configuration Mini-PC

4. 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

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.

Synthèse

Contributions et Résultats

- Développement d'un contrôleur de Slice réseau sur la plateforme d'EURECOM.
- Développement du Portail Web pour pouvoir l'utiliser de façon interactive.
- Proposition d'une architecture à adopter afin de permettre la synchronisation entre les parties prenantes des vols de drones.
- Proposition et implémentation d'un algorithme afin d'optimiser les vols de drones sur un réseau 5G.

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, 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’.

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.**

Merci pour votre attention !