

# INF3995 - Projet de conception d'un système informatique

## Readiness Review

---

### Équipe 100

Laurent Bolduc

Jacob Brisson

Jordan Lecourtois

Matthew Paoli

Philippe Savard

Simon Tran

Le 19 avril 2021

# Plan de la présentation

## ➤ **Introduction**

- ◊ Sujet de l'appel d'offre
- ◊ Méthodes de gestion de projet

## ◊ **Architecture du système**

- ◊ Architecture globale
- ◊ Logiciel embarqué
- ◊ Station au sol
- ◊ Interface de contrôle
- ◊ Fonctionnement général du système

## ◊ **Démonstrations**

- ◊ Démonstration de la simulation
- ◊ Démonstration des drones réels

## ◊ **Analyse du déroulement du projet**

- ◊ Réussites et échecs
- ◊ Fonctionnement de l'équipe

## ◊ **Conclusion**

- ◊ Tâches futures
- ◊ Apprentissages réalisés
- ◊ Analyse de la démarche de conception initiale

# Sujet de l'appel d'offre

Objectifs :

1. Réalisation d'une preuve de concept pour un système aérien minimal pour exploration
2. Réalisation d'un prototype fonctionnel de niveau de maturité NMS 4

Exigences :

1. Réalisation de la tâche en un temps maximal de 630 heures-personnes
2. Exploration autonome d'un bâtiment moyen d'une aire maximale de 100 m<sup>2</sup>
3. Interface de contrôle WEB accessible sur le réseau de la station au sol

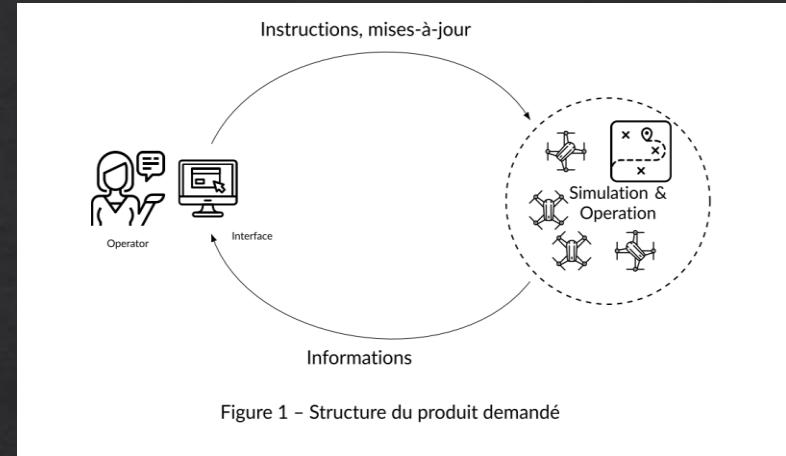


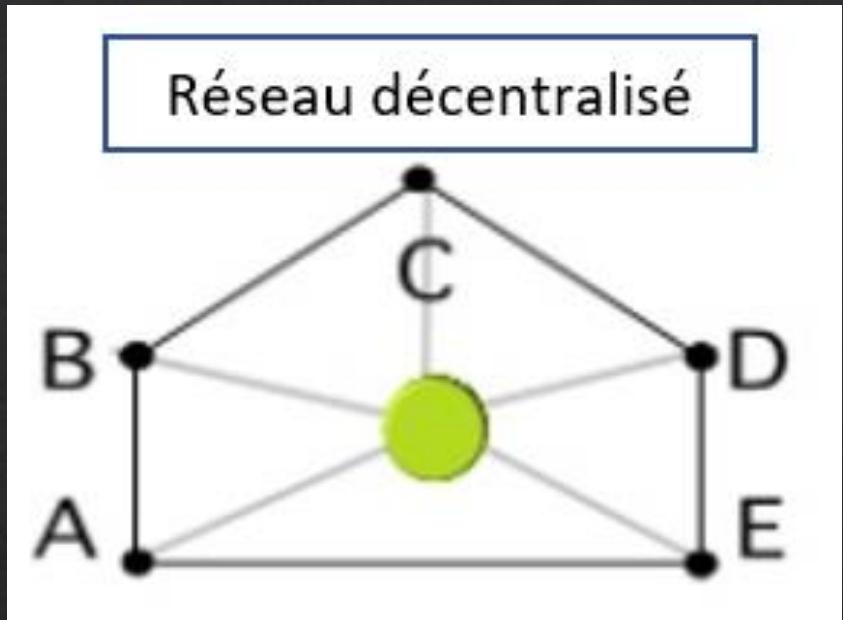
Figure 1 – Structure du produit demandé

Source : Exigences techniques.pdf

# Méthodes de gestion de projet

Rapports d'avancements hebdomadaires

- Établissement des tâches
  - Actualisation du temps de travail
- Organisation décentralisé avec un coordonnateur
- Deux équipes :
    1. Simulation et algorithmes
    2. Interface utilisateur et drones réels
  - Partage de l'avancement des travaux
  - Rétroactions agiles par les pairs



Source : Note de cours Polytechnique

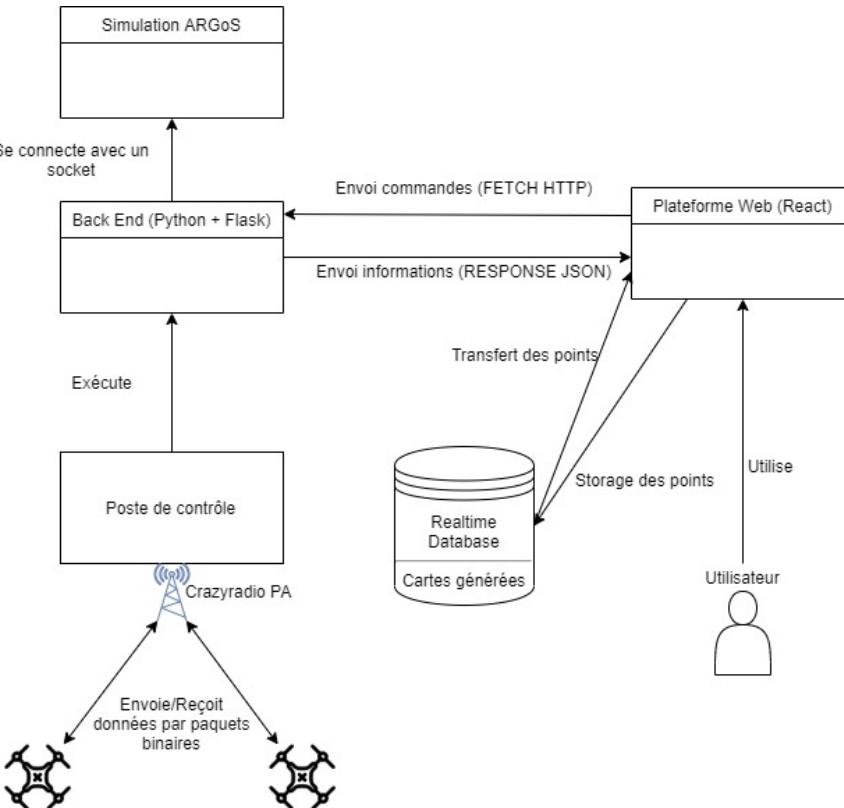
# Plan de la présentation

- ❖ Introduction
  - ◊ Sujet de l'appel d'offre
  - ◊ Méthodes de gestion de projet
- **Architecture du système**
  - ◊ Architecture globale
  - ◊ Logiciel embarqué
  - ◊ Station au sol
  - ◊ Interface de contrôle
  - ◊ Fonctionnement général du système
- ❖ Démonstrations
  - ◊ Démonstration de la simulation
  - ◊ Démonstration des drones réels
- ❖ Analyse du déroulement du projet
  - ◊ Réussites et échecs
  - ◊ Fonctionnement de l'équipe
- ❖ Conclusion
  - ◊ Tâches futures
  - ◊ Apprentissages réalisés
  - ◊ Analyse de la démarche de conception initiale

# Architecture globale

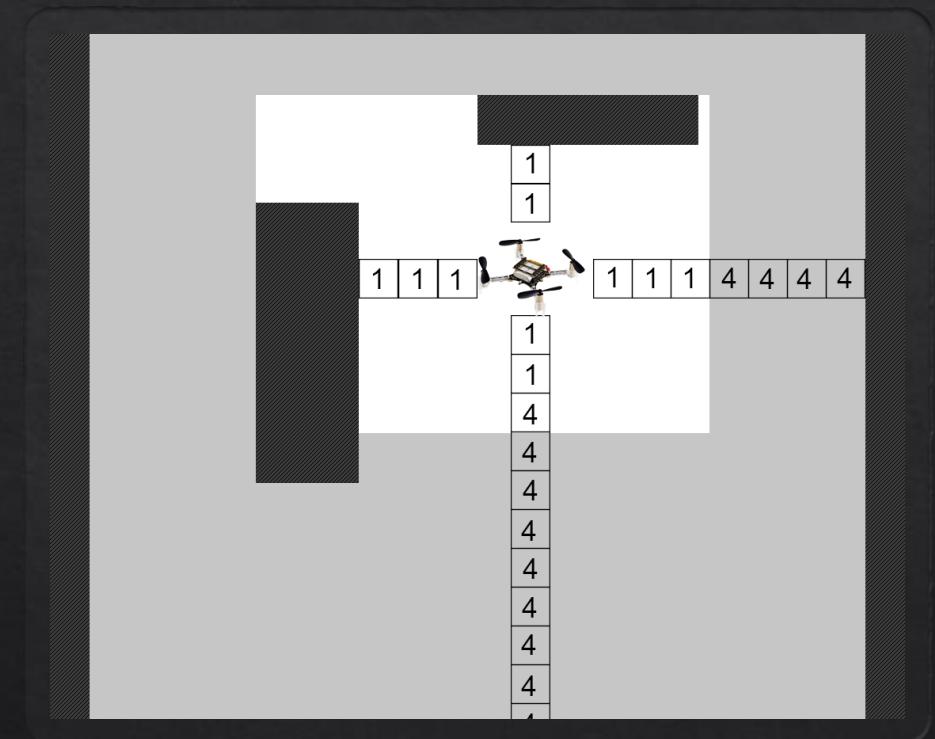
Quatre composants principaux :

- Simulation
- Application interne
- Interface utilisateur
- Micrologiciel embarqué



# Logiciel embarqué

- Algorithme d'exploration sur la simulation
  - Utilise une carte interne de 50 par 50 pour représenter l'arène
  - L'algorithme se divise en deux temps :
    1. Remplissage de la carte (trois valeurs possibles)
    2. Détermination de la prochaine direction permettant le plus grand gain d'informations



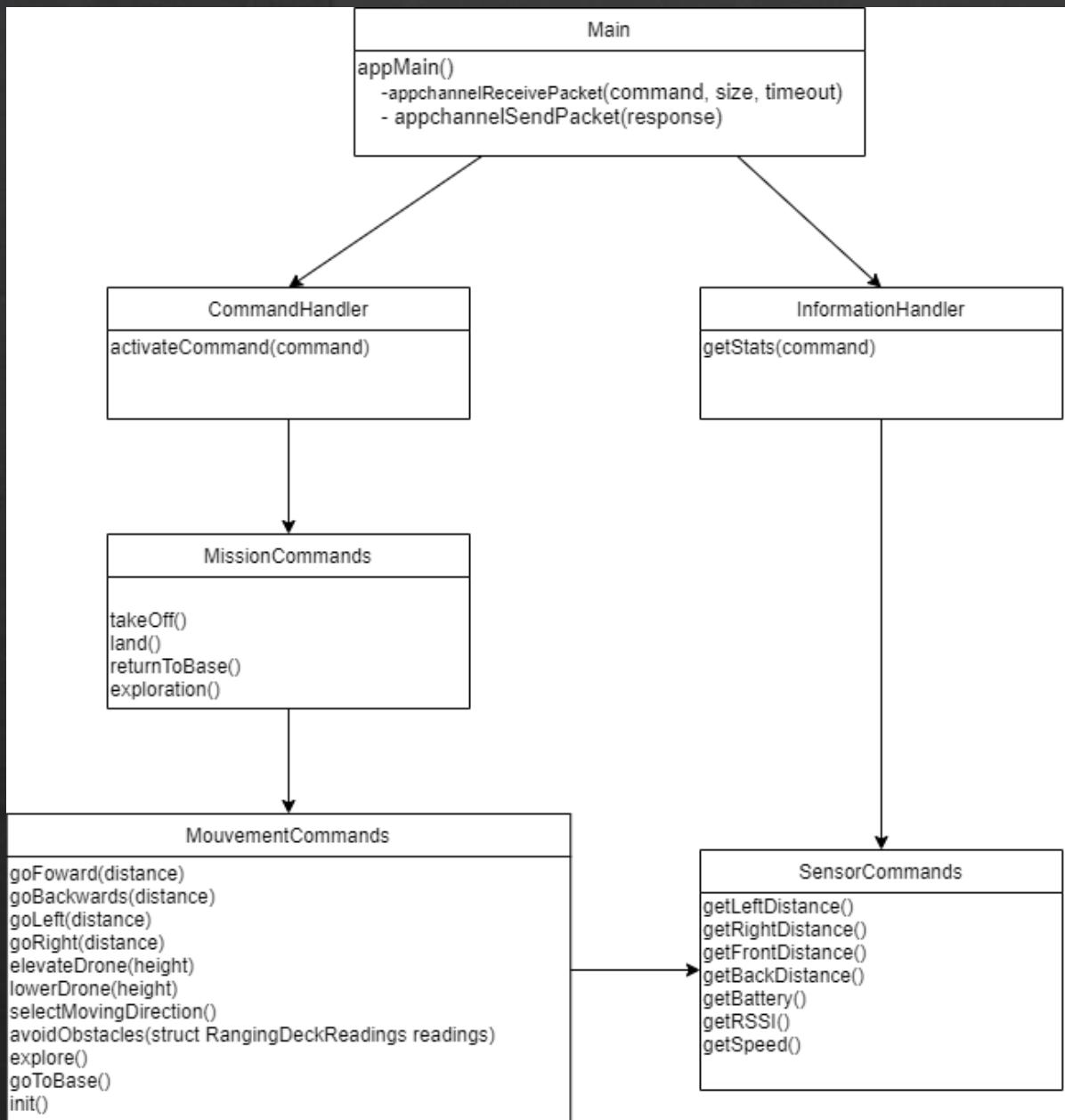
										-1
										-1
										-1
										-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
										-1
										-1
										-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

# Logiciel embarqué

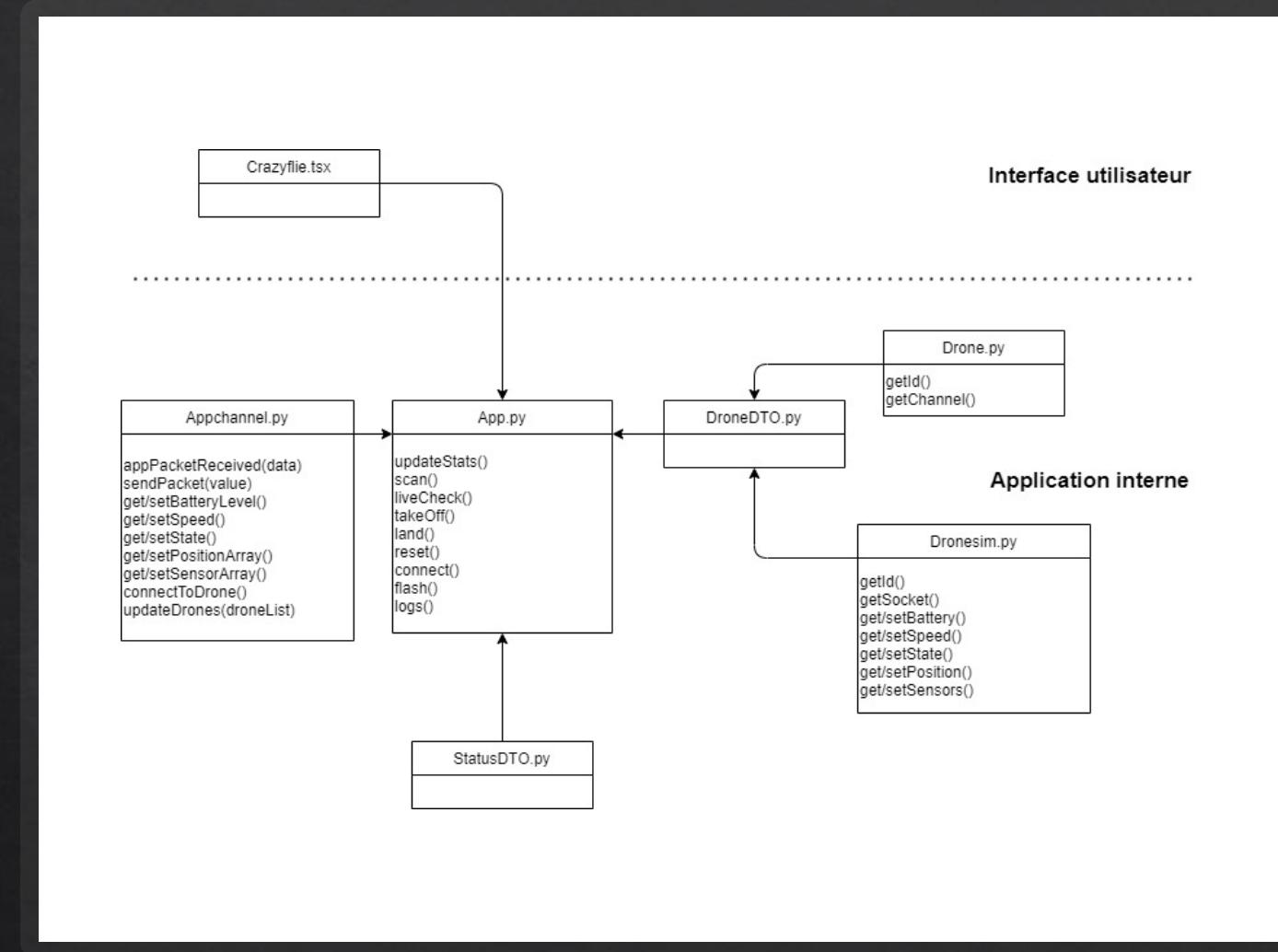
- Algorithme de retour à la base sur la simulation
  - Algorithme de type *Wavefront Planning Algorithm* (propagation par vague)
  - Utilisation des cases adjacentes avec le modèle de Von Neumann
  - Sélection de la prochaine direction de façon à diminuer la distance entre le drone et la base

# Code embarqué

- Reçoit les requêtes de l'application interne
  - Demande d'informations
  - Demande d'activation de commandes
- Implémentation des trois commandes:
  - Décoller
  - Atterrir
  - Retourner à la base
- Algorithme d'exploration et retour à la base:
  - Approche hybride d'une exploration aléatoire
  - Exploration sur un axe et changement de direction selon les informations des capteurs
  - Retour à la base selon une position fixe



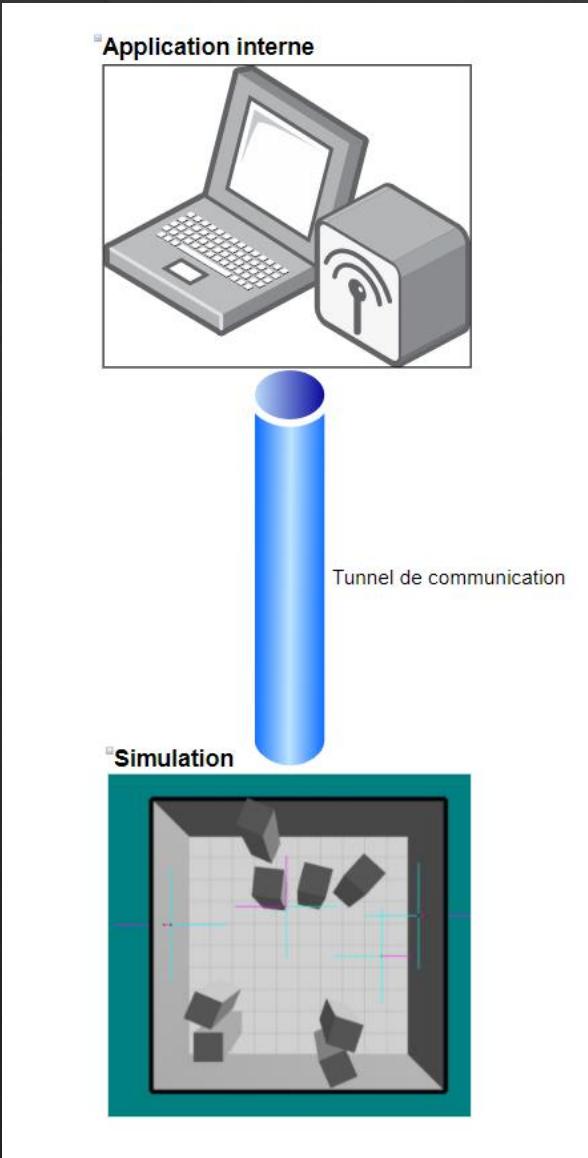
# Station au sol



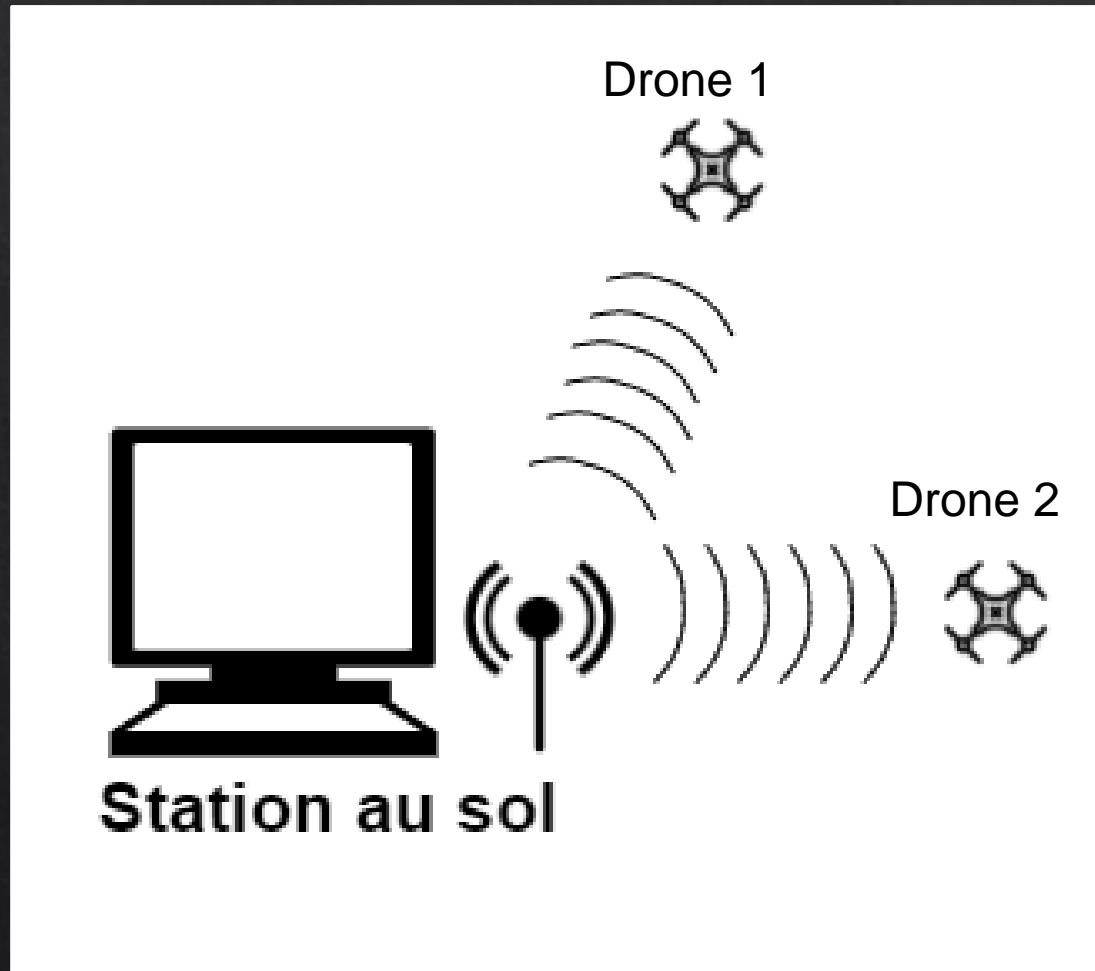
- Interaction entre l'utilisateur et le système
  - Interface de contrôle
  - Séparation entre la simulation et les drones
- Communications application WEB/ application interne

# Communication avec la simulation

- Initialisation du tunnel
  - Définition du bon nombre de drones
  - Envoi périodique des informations concernant les drones
- Communication de commandes vers la simulation
  - Envoi d'un octet de commande
  - Interprétation de la commande



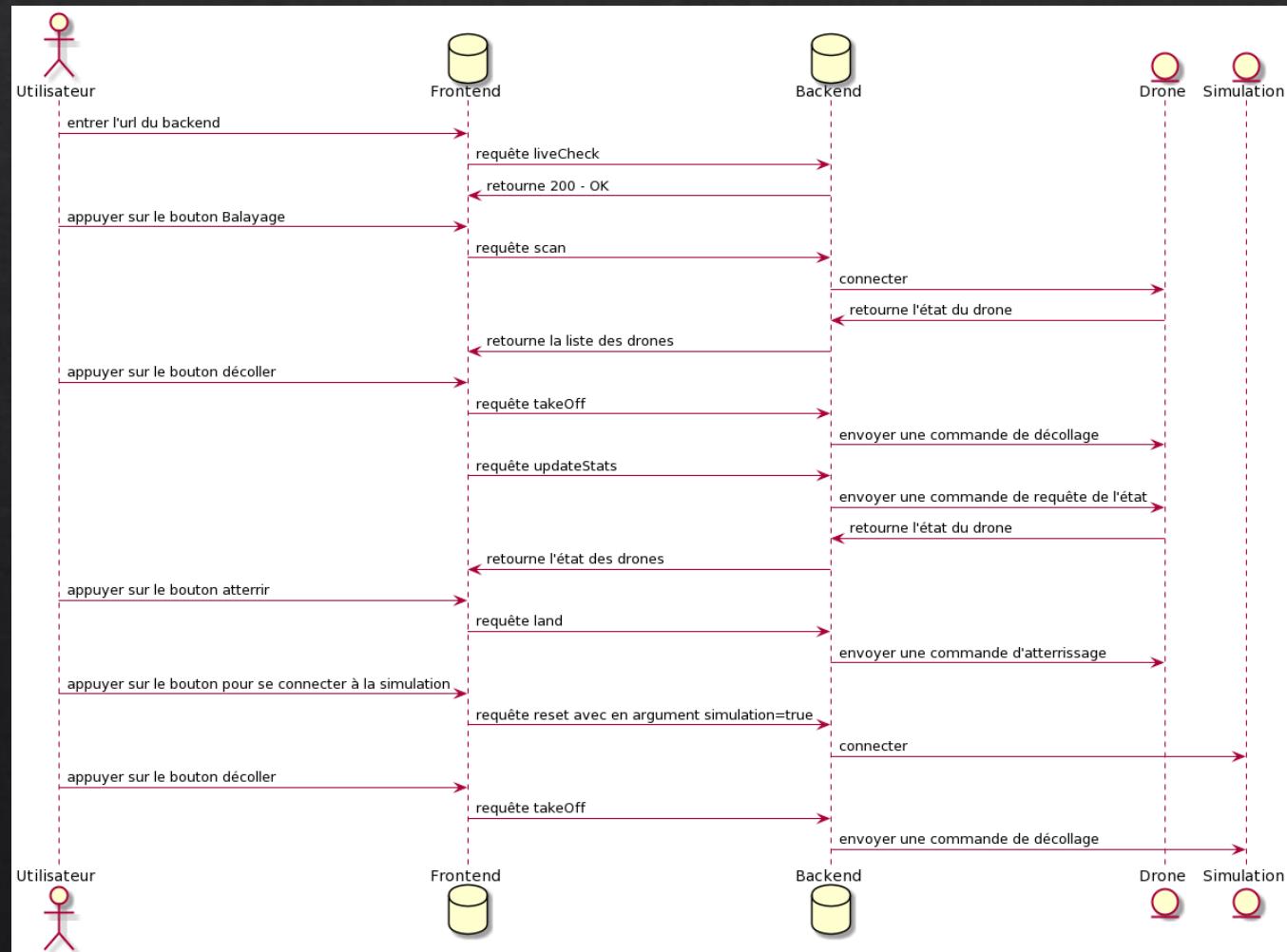
# Communication avec les drones (radio)



## ➤ Fichier *Appchannel.py*

- Fonctions appelées par *App.py*
- Envoi et réception de paquets

# Interface de contrôle



➤ Architecture Client-Serveur

➤ Interface accessible de partout

➤ Interface web, mobile, tablette ...

➤ Architecture REST

# Centre de contrôle des Crazyflies

INF3995 - Équipe 100

Type	Nom	Date de dernière modification	Actions
✈	04/14/2021, 5:10 PM	jeudi 15 avril 2021 à 12:15:16	🗑
✈	04/15/2021, 11:42 AM	jeudi 15 avril 2021 à 11:42:05	🗑
✈	04/15/2021, 7:19 AM	jeudi 15 avril 2021 à 10:19:43	🗑
✈	04/14/2021, 10:13 PM	mercredi 14 avril 2021 à 22:13:33	🗑
💻	04/14/2021, 9:56 PM	mercredi 14 avril 2021 à 22:13:20	🗑

NOUVELLE EXPLORATION...

Nouvelle Exploration

Nom  
04/15/2021, 1:01 PM

Simulation

[ANNULER](#) [CONFIRMER](#)

# Centre de contrôle

## ➤ Historique d'exploration

- Informations sur l'exploration (nom, type)
- Date de l'exploration
- Option pour supprimer une entrée

## ➤ Nouvelle exploration

- Simulation ou exploration réelle

# Exploration

## 1. État des drones

- Batterie
- Vitesse
- ID

## 2. Visualisation de la carte

- Drones (points bleus)
- Obstacles (carrés rouges)
- Espace exploré (blanc)
- Agrandissement et déplacement de la carte

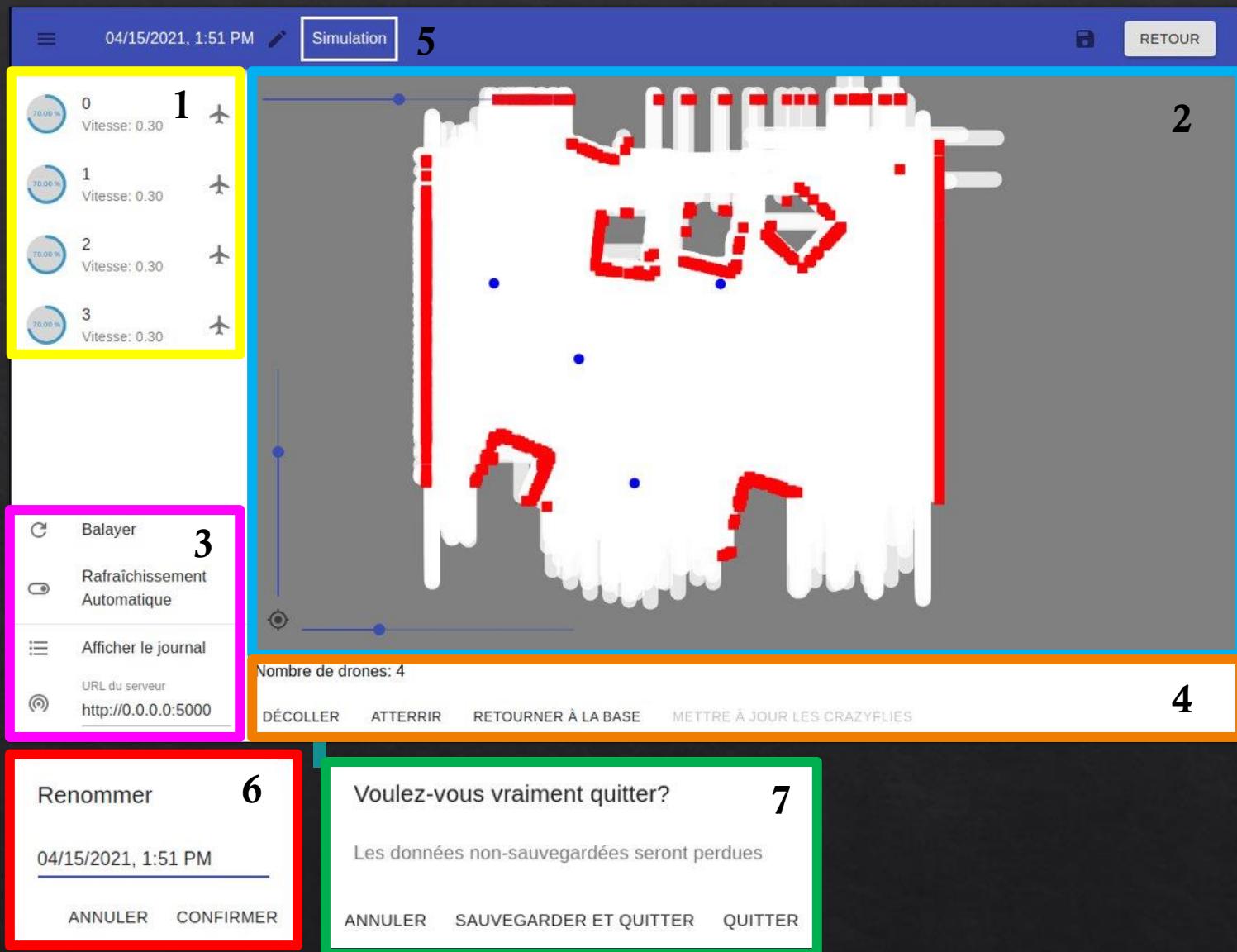
## 3. Configuration

## 4. Panneau de contrôle

## 5. Données sur l'exploration

## 6. Option de renommage

## 7. Option pour quitter



# Initialisation des drones

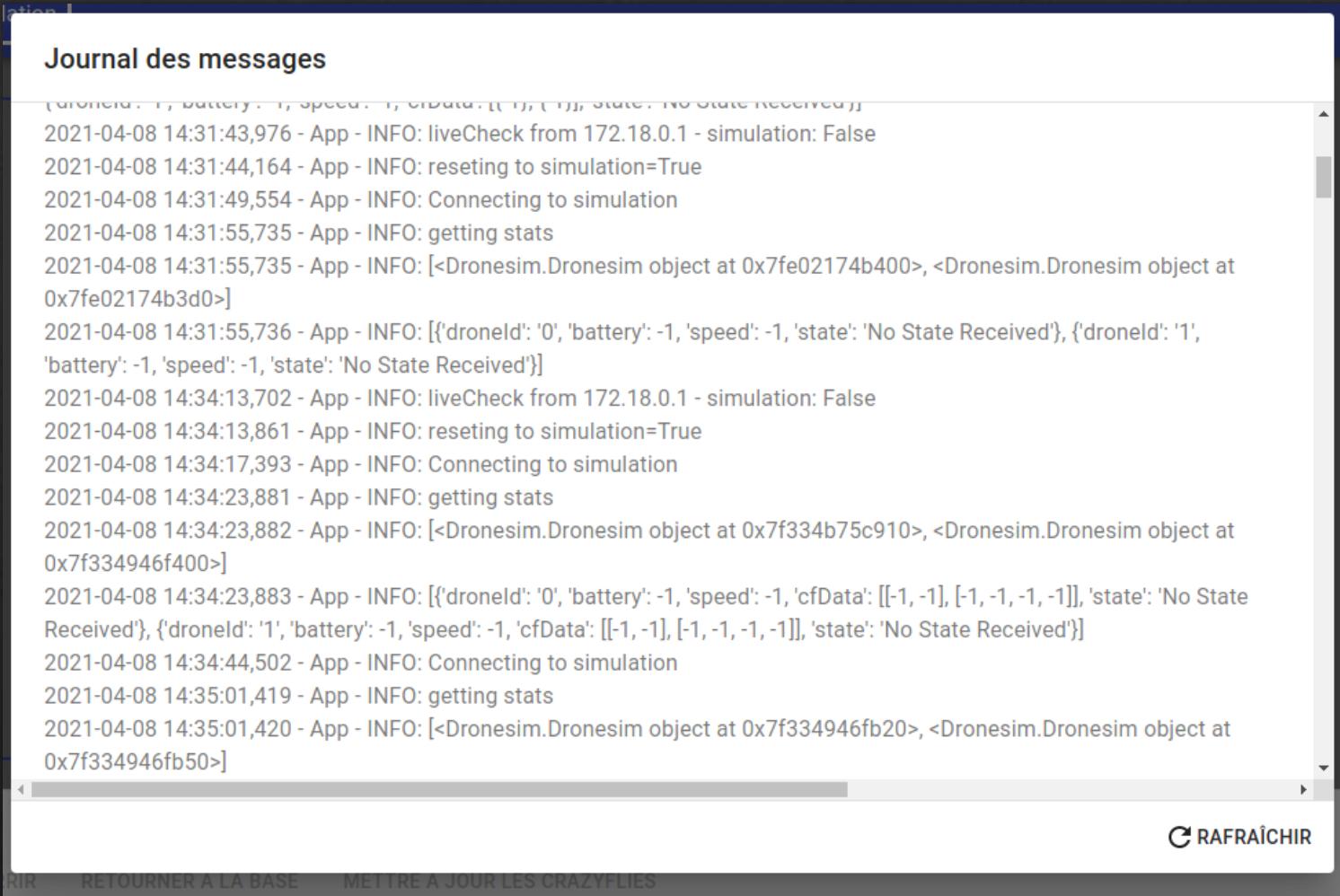
- Position initiale dans l'arène
- Positions en coordonnées réelles (m)
- Positions données dans le journal des messages de la simulation

```
Log
[t=0] Log started.
[t=2] Initial position: id: 7
[t=2] <x: 36.3048, y: 34.5909 >;
[t=2] Initial position: id: 8
[t=2] <x: 44.8118
[t=2] , y: 34.8706 >;
```



# Journal des messages

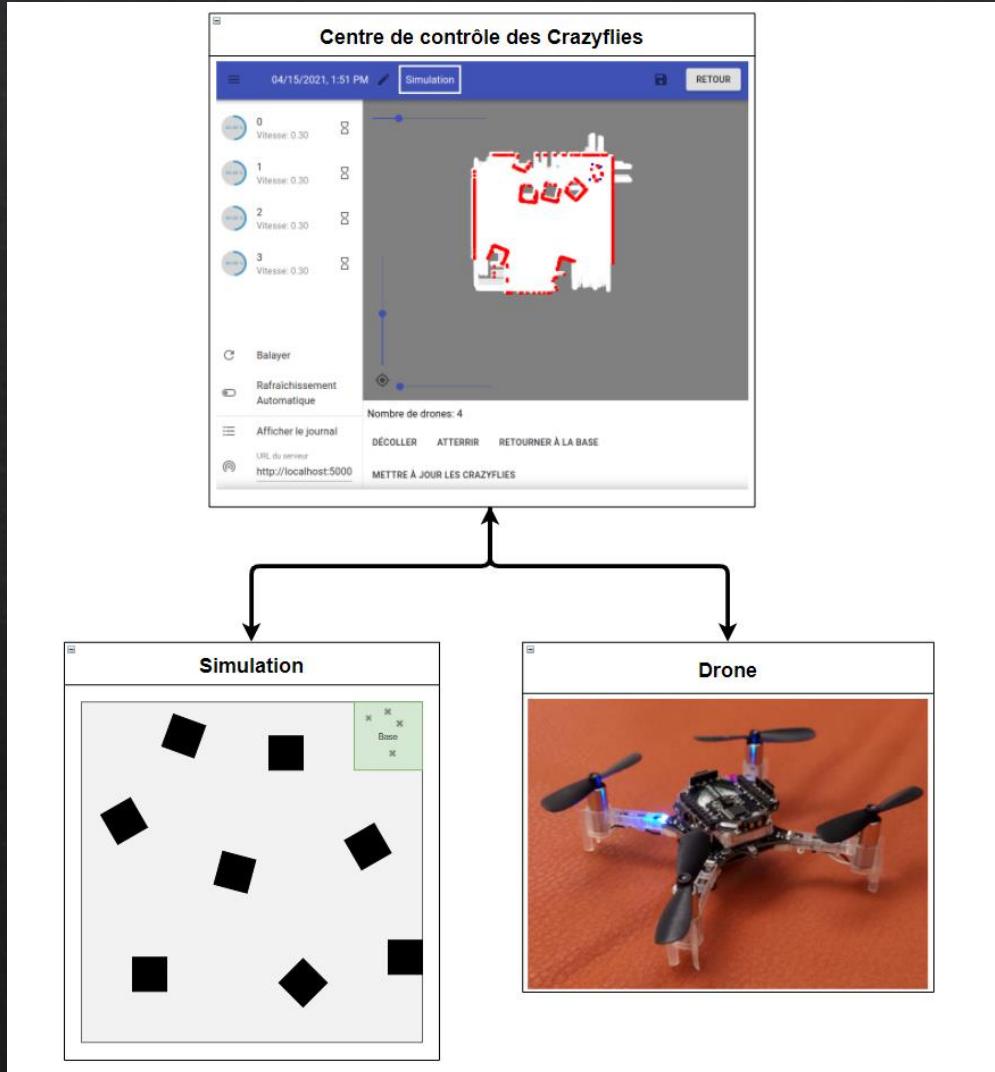
- État de la station au sol
- Historique des commandes
- Autres messages pertinents pour le déverminage



The screenshot shows a terminal window with a dark background and light-colored text. The title bar reads "Journal des messages". The main area contains a list of log entries. At the bottom, there are buttons for "REFRESH", "RETOURNER A LA BASE", and "METTRE A JOUR LES CRAZYFLIES". On the far right, there is a "RAFRAÎCHIR" button with a circular arrow icon.

```
2021-04-08 14:31:43,976 - App - INFO: liveCheck from 172.18.0.1 - simulation: False
2021-04-08 14:31:44,164 - App - INFO: resetting to simulation=True
2021-04-08 14:31:49,554 - App - INFO: Connecting to simulation
2021-04-08 14:31:55,735 - App - INFO: getting stats
2021-04-08 14:31:55,735 - App - INFO: [<Dronesim.Dronesim object at 0x7fe02174b400>, <Dronesim.Dronesim object at 0x7fe02174b3d0>]
2021-04-08 14:31:55,736 - App - INFO: [{"droneId": '0', 'battery': -1, 'speed': -1, 'state': 'No State Received'}, {"droneId": '1', 'battery': -1, 'speed': -1, 'state': 'No State Received'}]
2021-04-08 14:34:13,702 - App - INFO: liveCheck from 172.18.0.1 - simulation: False
2021-04-08 14:34:13,861 - App - INFO: resetting to simulation=True
2021-04-08 14:34:17,393 - App - INFO: Connecting to simulation
2021-04-08 14:34:23,881 - App - INFO: getting stats
2021-04-08 14:34:23,882 - App - INFO: [<Dronesim.Dronesim object at 0x7f334b75c910>, <Dronesim.Dronesim object at 0x7f334946f400>]
2021-04-08 14:34:23,883 - App - INFO: [{"droneId": '0', 'battery': -1, 'speed': -1, 'cfData': [[-1, -1], [-1, -1, -1, -1]], 'state': 'No State Received'}, {"droneId": '1', 'battery': -1, 'speed': -1, 'cfData': [[-1, -1], [-1, -1, -1, -1]], 'state': 'No State Received'}]
2021-04-08 14:34:44,502 - App - INFO: Connecting to simulation
2021-04-08 14:35:01,419 - App - INFO: getting stats
2021-04-08 14:35:01,420 - App - INFO: [<Dronesim.Dronesim object at 0x7f334946fb20>, <Dronesim.Dronesim object at 0x7f334946fb50>]
```

# Fonctionnement général du système



```
simon@DESKTOP-BMN25LN MINGW64 /q/Work/Sess6/inf3995/inf3995-main (master)
$ ./start.sh
```

- Harmonisation des commandes de configuration
  - Conteneur Docker indépendant pour chaque composant du système
  - Communication entre l'interface et la station au sol par requêtes HTTP
  - Communication entre la station au sol et les drones/la simulation par des tunnels
- Déploiement de la base de données et de l'interface sur Firebase

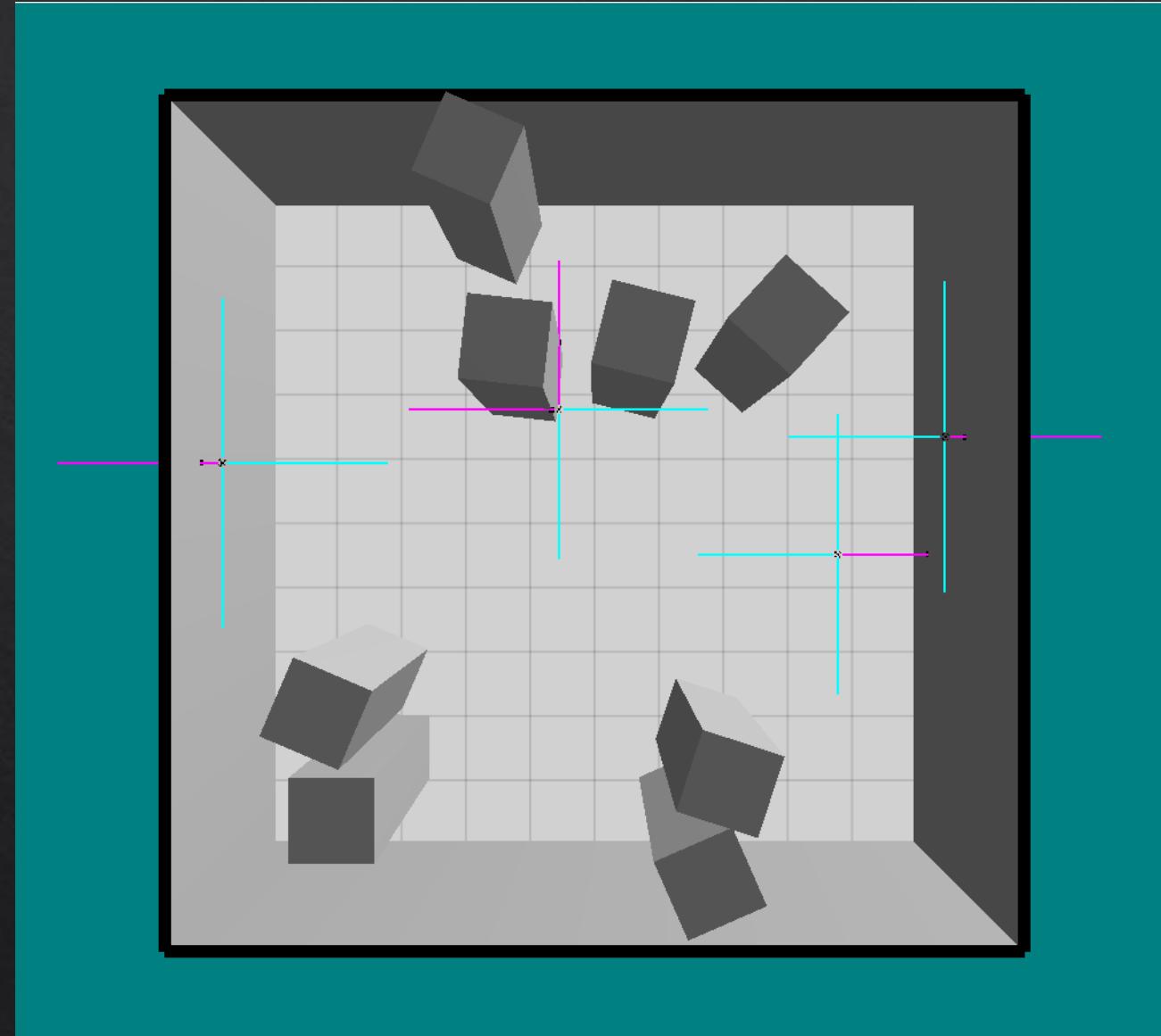


Source : <https://firebase.google.com/>

# Plan de la présentation

- ❖ Introduction
  - ◊ Sujet de l'appel d'offre
  - ◊ Méthodes de gestion de projet
- ❖ Architecture du système
  - ◊ Architecture globale
  - ◊ Logiciel embarqué
  - ◊ Station au sol
  - ◊ Interface de contrôle
  - ◊ Fonctionnement général du système
- **Démonstrations**
  - ◊ Démonstration de la simulation
  - ◊ Démonstration des drones réels
- ❖ Analyse du déroulement du projet
  - ◊ Réussites et échecs
  - ◊ Fonctionnement de l'équipe
- ❖ Conclusion
  - ◊ Tâches futures
  - ◊ Apprentissages réalisés
  - ◊ Analyse de la démarche de conception initiale

# Démonstration de la simulation



# Démonstration des drones réels



# Plan de la présentation

- ❖ Introduction
  - ◊ Sujet de l'appel d'offre
  - ◊ Méthodes de gestion de projet
- ❖ Architecture du système
  - ◊ Architecture globale
  - ◊ Logiciel embarqué
  - ◊ Station au sol
  - ◊ Interface de contrôle
  - ◊ Fonctionnement général du système
- ❖ Démonstrations
  - ◊ Démonstration de la simulation
  - ◊ Démonstration des drones réels
- **Analyse du déroulement du projet**
  - ◊ Réussites et échecs
  - ◊ Fonctionnement de l'équipe
- ❖ Conclusion
  - ◊ Tâches futures
  - ◊ Apprentissages réalisés
  - ◊ Analyse de la démarche de conception initiale

# Réussites

- Informations pertinentes visibles sur l'interface à une fréquence de 1 Hz
- Drones simulés et réels capables d'explorer et de retourner à la base
- Système fonctionnel avec plusieurs drones
- Carte représentative et précise pour la simulation
- Mise à jour à distance possible
- Un journal des opérations est visible pour les différents composants du système



# Échecs

- Tests pour chacun des composants
- Envoi de messages par pair-à-pair pour l'évitement
- Faire fonctionner la simulation avec un grand nombre de drones
- Précision de la carte avec les drones réels
- Certains requis plus spécifiques



# Fonctionnement de l'équipe

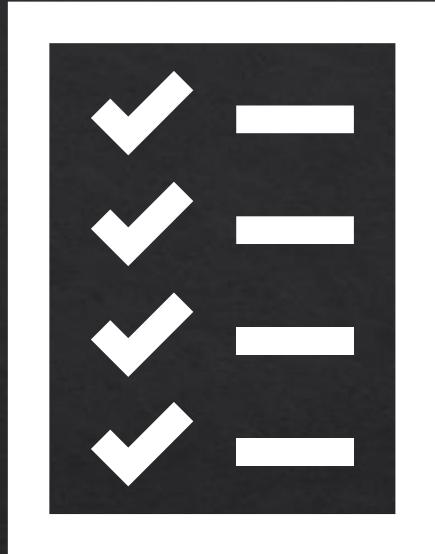
- ✓ Satisfaction des exigences techniques du mandat initial      ✗ Estimation du temps de travail
- ✓ Comportement global du projet réussi      ✗ Rapports d'avancements non fiables
- ✓ Respect de l'échéancier et bonne distribution des tâches      ✗ Problèmes techniques
- ✓ Mode de fonctionnement décentralisé      ✗ Changement des algorithmes au cours du projet
- ✓ Équipe dynamique favorisant l'entraide entre les membres      ✗ Implémentation tardive des algorithmes sur les drones

# Plan de la présentation

- ❖ Introduction
  - ◊ Sujet de l'appel d'offre
  - ◊ Méthodes de gestion de projet
- ❖ Architecture du système
  - ◊ Architecture globale
  - ◊ Logiciel embarqué
  - ◊ Station au sol
  - ◊ Interface de contrôle
  - ◊ Fonctionnement général du système
- ❖ Démonstrations
  - ◊ Démonstration de la simulation
  - ◊ Démonstration des drones réels
- ❖ Analyse du déroulement du projet
  - ◊ Réussites et échecs
  - ◊ Fonctionnement de l'équipe
- Conclusion
  - ◊ Tâches futures
  - ◊ Apprentissages réalisés
  - ◊ Analyse de la démarche de conception initiale

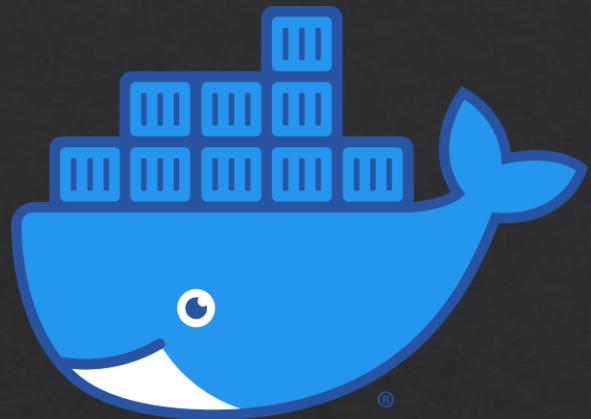
# Tâches futures

- Afin de respecter les exigences :
  - Nombre arbitraire de drones pour la simulation et les drones réels
  - Architecture de l'application interne
  - Précision de la carte avec les drones réels
  - Implémenter les algorithmes de la simulation sur les drones
- Recommandations d'ajouts :
  - Envoi de commandes uniquement à certains drones
  - Création d'une carte en trois dimensions de l'environnement visité
  - Utilisation de capteurs plus précis sur les drones et d'un système de géolocalisation
  - Ajouter une caméra sur les drones afin de voir des images du lieu visité en direct



# Apprentissages réalisés

- Comment conceptualiser un système informatique du début à la fin
- Utilité des « Docker Containers »
- Pertinence de tester la logique d'exploration sur une simulation avant d'implémenter sur les drones réels.
- Concept de réseautique dans un système
  - Tunnel entre applications
  - Communication avec les drones
- Synchronisation du système global



source : <https://jackmckew.dev/intro-to-docker.html>

# Analyse de la démarche de conception initiale

## Points faibles :

- Réfléchir plus longtemps sur l'impact des choix technologiques
- Meilleure planification initiale et estimation de temps des tâches à réaliser
- Analyse initiale plus poussée des algorithmes d'exploration et de retour à la base
- Intégration des tests lors du développement

## Points forts :

- L'architecture générale du système était bien réfléchie
- Le développement des applications dans des conteneurs « Docker »

# Références

- ◊ [1] Flask. (2021). *Quickstart*. <https://flask.palletsprojects.com/en/1.1.x/quickstart/>
- ◊ [2] Bitcraze. (2021). *Crazyflie-lib-python*. <https://github.com/bitcraze/crazyflie-lib-python/tree/master/cplib>
- ◊ [3] Bitcraze. (2021). *Crazyradio PA*. <https://www.bitcraze.io/products/crazyradio-pa/>
- ◊ [4] Python. (2021). *socket — Low-level networking interface*. <https://docs.python.org/3/library/socket.html>
- ◊ [5] GeeksforGeeks. (2019). *Socket Programming in C/C++*. <https://www.geeksforgeeks.org/socket-programming-cc/>
- ◊ [6] Bitcraze. (2021). *Crazyflie-firmware*. <https://github.com/bitcraze/crazyflie-firmware/tree/1a4cc9acf3edefb3833b72e05095be9c8b0fc887>
- ◊ [7] Bitcraze. (2020). *Peer to Peer API*. [https://www.bitcraze.io/documentation/repository/crazyflie-firmware/2020.02/p2p\\_api/](https://www.bitcraze.io/documentation/repository/crazyflie-firmware/2020.02/p2p_api/)
- ◊ [8] Facebook Open Source. (2021). React.js (Version 17.0.2) [Logiciel]. <https://reactjs.org/docs/getting-started.html>
- ◊ [9] Material-UI. (2020) Material-UI (Version 4.11.1) [Logiciel]. <https://material-ui.com/>
- ◊ [10] Microsoft. (2021). TypeScript (Version 4.2) [Logiciel]. <https://www.typescriptlang.org/>
- ◊ [11] lodash. (2016). Lodash (Version 4.0.0) [Logiciel]. <https://lodash.com/>
- ◊ [12] Facebook Open Source. (2021). Jest (Version 26.6.3) [Logiciel]. <https://jestjs.io/>
- ◊ [13] Koss, S. (2021). Date-fns (Version 2.21.1) [Logiciel]. <https://date-fns.org/>
- ◊ [14] Dodds, K. C. (2021). Testing Library [Logiciel]. <https://testing-library.com/docs/react-testing-library/intro/>
- ◊ [15] Typicode. (2021). Husky (Version 6.0.0) [Logiciel]. <https://typicode.github.io/husky/#/>
- ◊ [16] Okonetchnikov, A. (2021). Lint-Staged (Version 10.5.4) [Logiciel]. <https://github.com/okonet/lint-staged>
- ◊ [17] Prettier. (2020). Prettier (Version 2.2.1) [Logiciel]. <https://prettier.io/>
- ◊ [18] Varshavskaya, P. Wavefront Planning Algorithm. Tufts University. <https://www.cs.tufts.edu/comp/150IR/labs/wavefront.html>