

DST Airlines

Tracking des vols en temps réel sur
le territoire européen

THALES





Projet de dashboard interactif de suivi en direct des avions en vol sur le territoire européen

❑ Récolter

- Trouver des sources de données pertinentes
- Assurer l'acquisition des données en continue



❑ Structurer

- Enregistrer les données récoltées
- Adapter l'architecture aux types de données



❑ Consommer

- Réalisation d'un dashboard de tracking des vols en direct
 - Carte géographique avec l'ensemble des vols
 - Possibilité de filtrer les données
 - Proposer des statistiques sur les vols passés



❑ Déployer

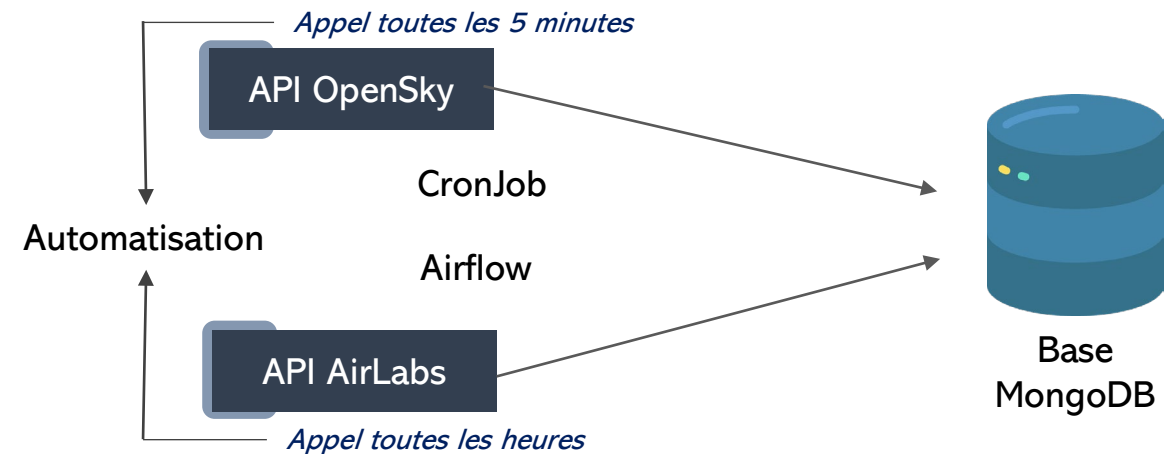
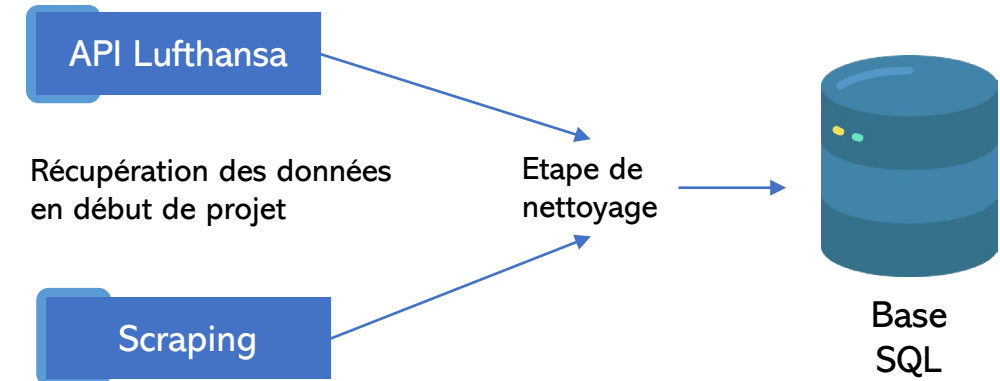
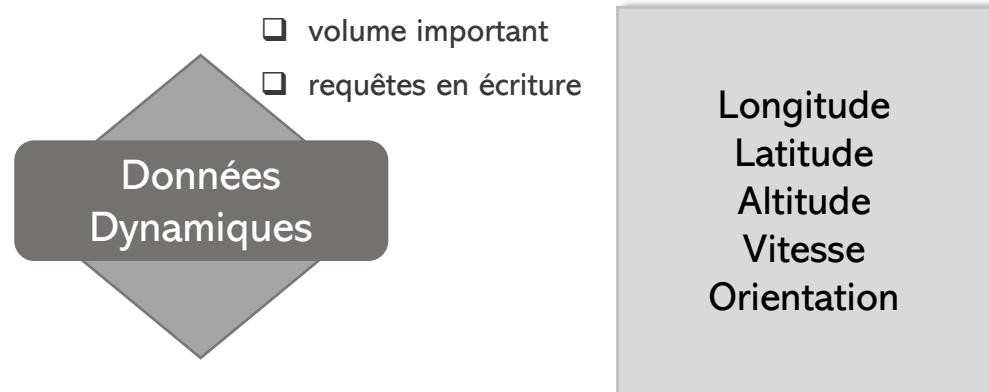
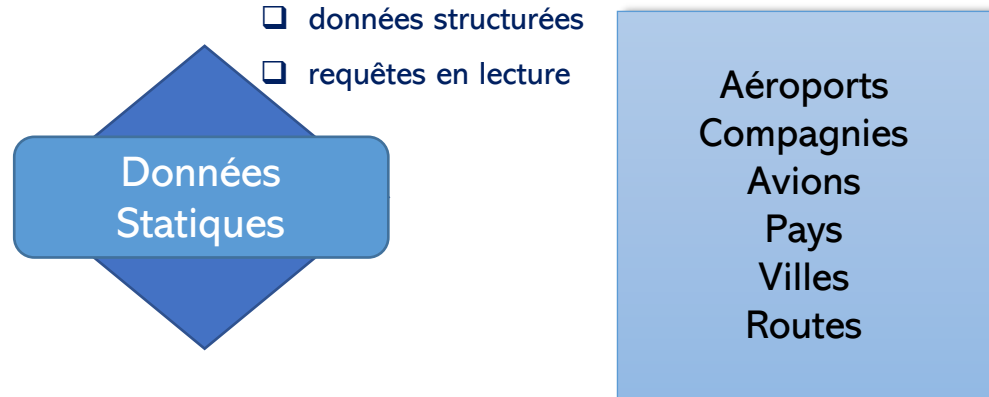
- Proposer une API
 - Récupérer les données
 - Modifier les données
- Dockeriser l'application



❑ Automatiser

- Automatiser l'acquisition des données dynamiques
 - Airflow





- 2 API différentes comme source de données dynamiques car informations complémentaires
- Contraintes imposées par ces API concernant le nombre de requêtes autorisées : 1000 requêtes / jour pour opensky vs 1000 requêtes / mois pour airlabs



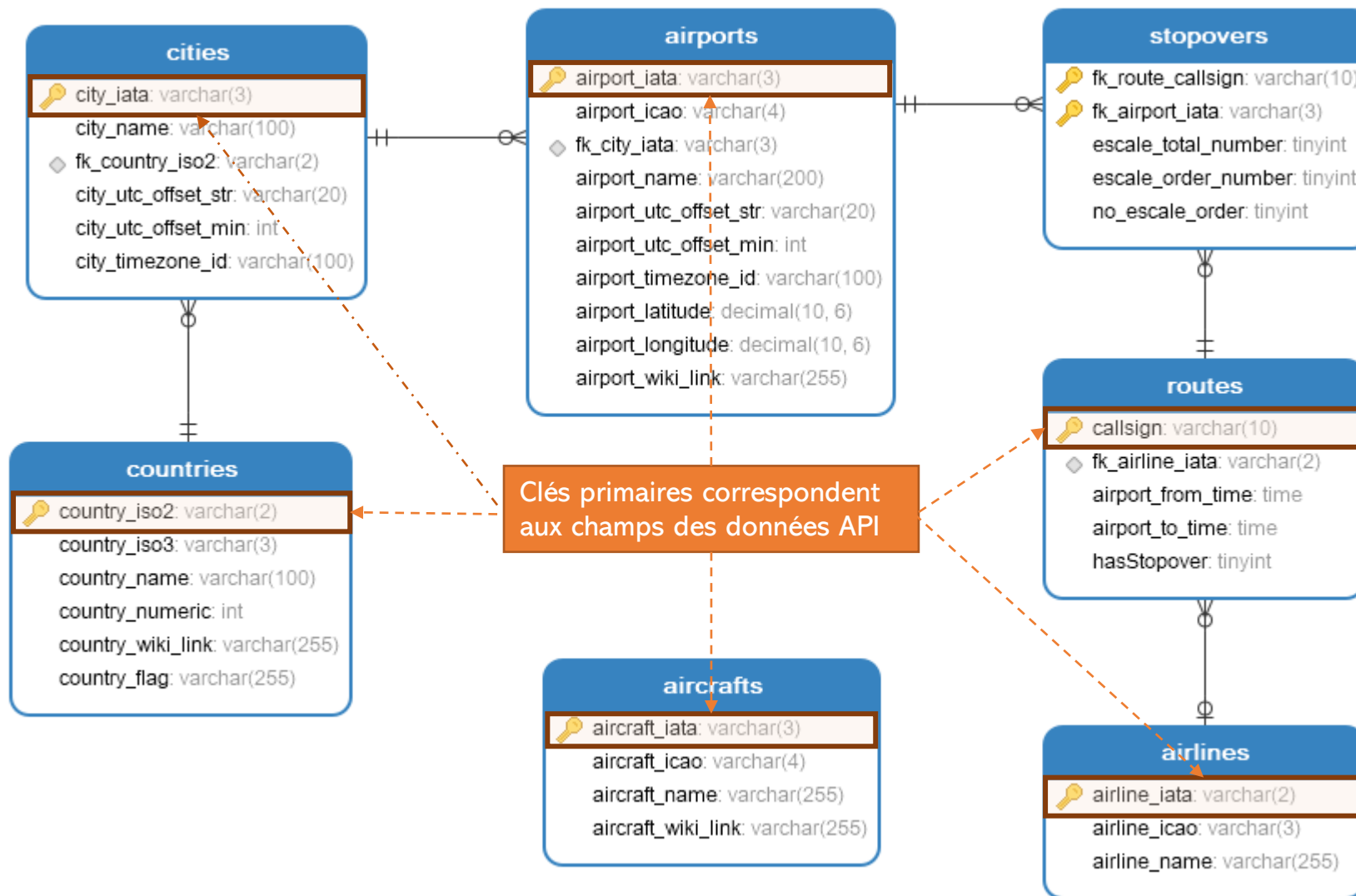
Création de la base
+ Insertion des données

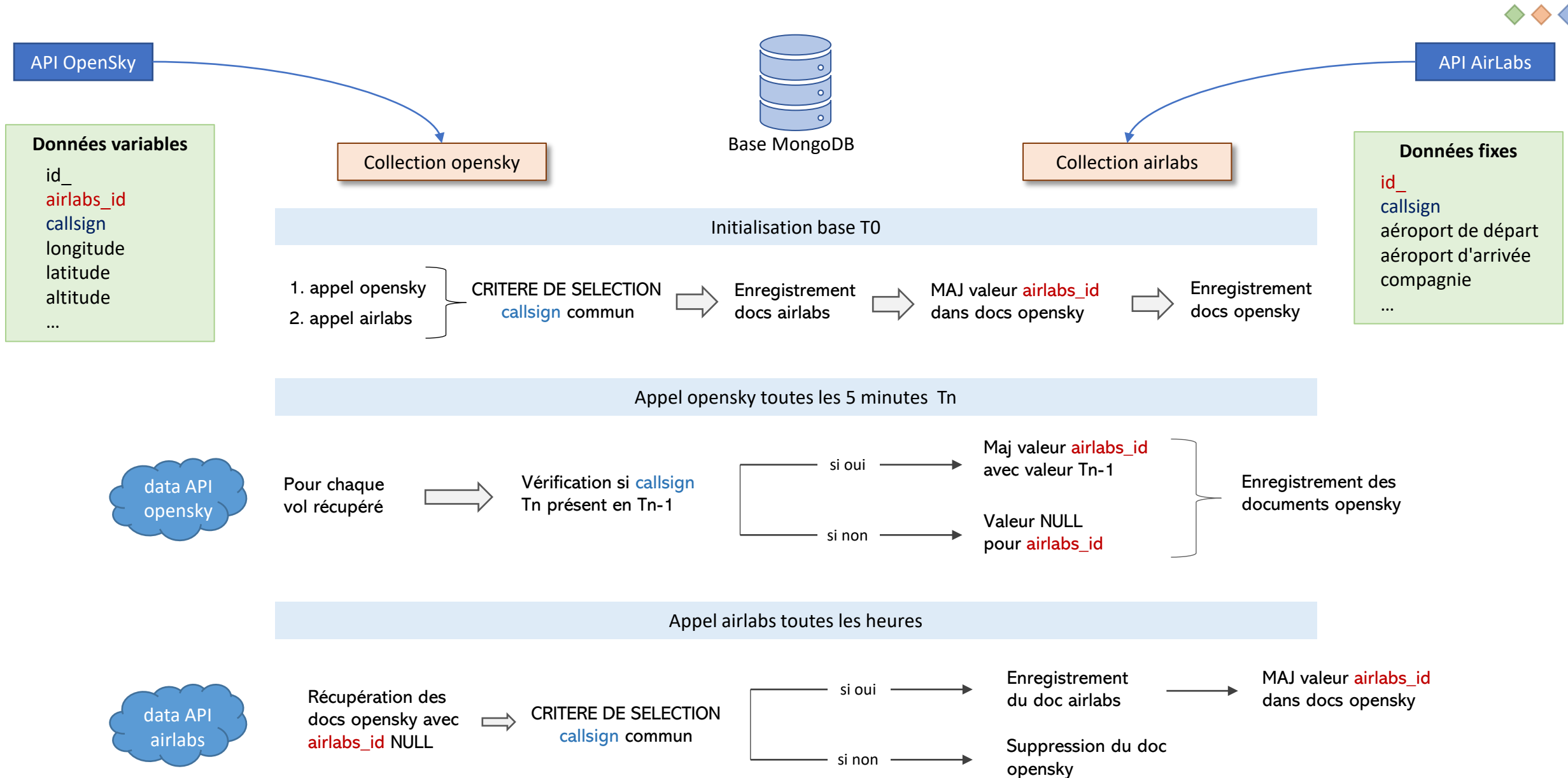


En début de projet

Mise à jour des données
(Insert Update Delete)

Réalisée via une API
Pour admin seulement





Agrégation des données

Objectif : regrouper les données de chaque vol dans un seul document

Exemple : callsign AJT8621 du 2023-05-28

Collection opensky

datetime	airlabs_id	callsign	longitude	latitude
2023-05-28 14:43:02	64735056e6154b1c37746f84	AJT8621	-10,7222	4
2023-05-28 14:48:03	64735056e6154b1c37746f84	AJT8621	-9,8298	4
2023-05-28 14:53:02	64735056e6154b1c37746f84	AJT8621	-8,9159	4
...				
2023-05-28 16:08:08	64735056e6154b1c37746f84	AJT8621	4,4558	5
2023-05-28 16:13:07	64735056e6154b1c37746f84	AJT8621	4,4953	5

Collection airlabs

_id	flight_icao	flight_iata	dep_iata	arr_iata	airline
64735056e6154b1c37746f84	AJT8621	M68621	SJU	BRU	M6

Opération réalisée 1 fois par jour, dans la nuit (par cronjob / airflow)



Base MongoDB

1 document enregistré
toutes les 5minutes



on récupère :

- le premier datetime comme datetime_start
- le dernier datetime comme datetime_end

Merge des données
agrégées avec Airlabs

Collection data_aggregated

_id	callsign	datetime_start	datetime_end	airline_iata	arr_iata
64735056e6154b1c37746f84	AJT8621	2023-05-28 14:43:02	2023-05-28 16:13:07	M6	BRU

→ un vol = une ligne



<http://193.70.85.253:8050/>





- Mise en pratique des connaissances acquises durant la formation de Data Engineer de DataScientest
 - ➔ Bases de données MongoDB, Docker, Airflow
- Projet de formation qui reste un projet de travail sans vocation à être diffusé
 - ➔ Sécurisation nécessaire (protocole https a minima)
 - ➔ Scalabilité de l'application (avec Kubernetes par exemple)
- Appréhender de manière plus concrète certains aspects métiers
 - ➔ Préparer au mieux mon entrée en poste prochaine

Remerciements :

- à l'équipe DataScientest pour la formation
- à l'équipe Thales pour leur attention

QUESTIONS ?





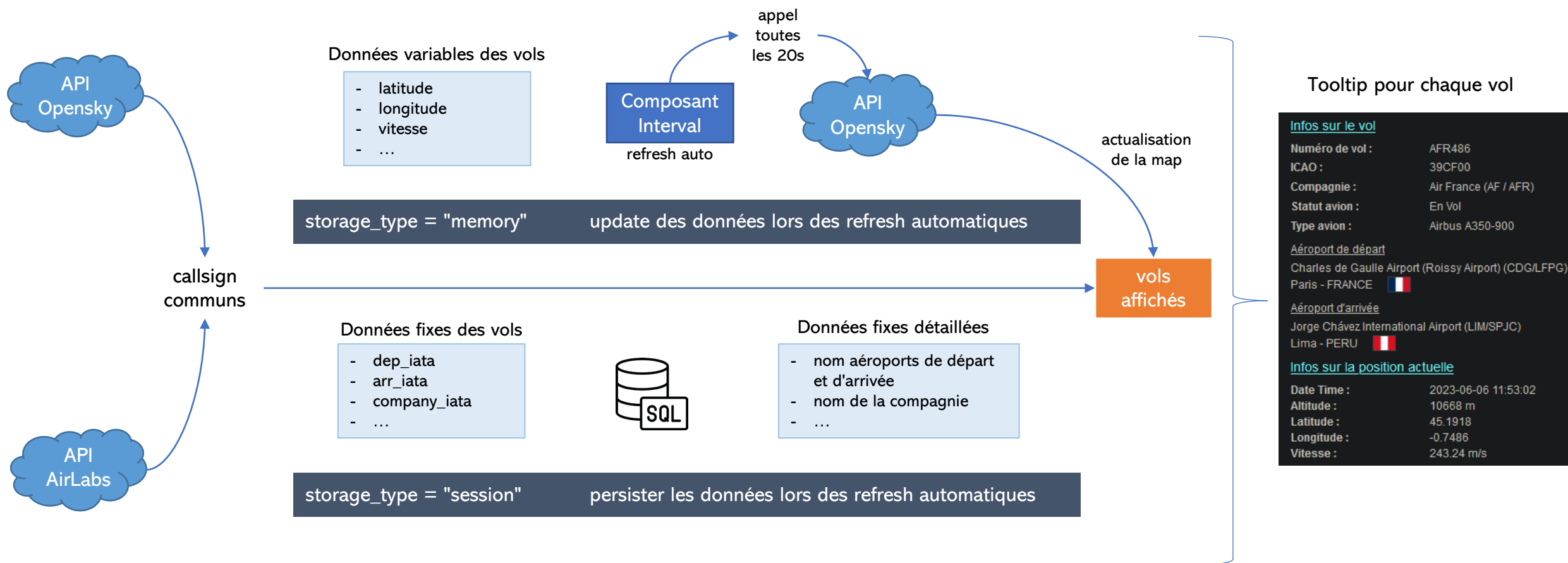
- [Annexe 1 : Dashboard - Live Map](#)
- [Annexe 2 : Dashboard - Statistics Data et Statistics Map](#)
- [Annexe 3 : API Publique et Privée](#)
- [Annexe 4 : Déploiement Docker](#)
- [Annexe 5 : Automatisation des appels API : Airflow](#)

Page Map Live

- Objectif
- Moyen

Visualiser sur une carte les vols en direct sur l'Europe
Component Dash Leaflet

mise à jour automatique
présence de filtres



Page Statistics Data

• Partie 1 – Statique : Statistiques journalières

count → graphiques
average → cards

• Partie 2 – Dynamique : Statistiques journalières

Statistiques filtrées par compagnies, aéroports de départ, d'arrivée...

→ résultat sous forme d'une table HTML avec :

- des infos statiques issues de la base MySQL
- des données agrégées issues de la base MongoDB

• Partie 3 – Dynamique : Statistiques des vols passés

Affichage des différents vols enregistrés pour un aéroport de départ et d'arrivée donné

→ comparer les mêmes vols à différentes dates



Page Statistics Map

• Partie 1 – Routes principales

Pour chaque aéroport de départ



Routes les plus fréquentes
(15 au maximum)

• Partie 2 – Route d'un vol

Pour tous les vols du jour



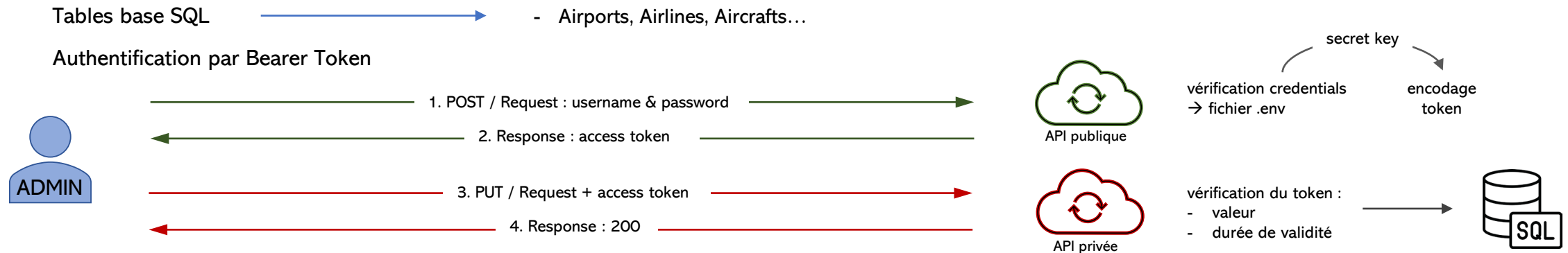
Affichage des différentes
positions enregistrées



• Récupération des données : routes GET -----> API publique

Data Statiques	<ul style="list-style-type: none"> - Table complète - ou filtrée par une ou plusieurs valeurs de la clé primaire 	Exemple : table airport → récupérer data des aéroports CDG et BOD
Data Statistiques	<p>Eléments obligatoires :</p> <ul style="list-style-type: none"> - la variable de recherche (callsign, compagnie, aéroport de départ...) - la ou les valeurs recherchées <p>Elément optionnel :</p> <ul style="list-style-type: none"> - date recherchée 	Exemple : variable city → récupérer data <ul style="list-style-type: none"> - de la ville PAR - pour la date du 2023-06-06
Data Dynamiques	<ul style="list-style-type: none"> - Ensemble des données dynamiques à l'instant T (ou filtrées par un élément) - Positions d'un vol du jour - Les routes principales d'un aéroport de départ donné 	Exemple : Ensemble des données dynamiques des avions actuellement en vol : <ul style="list-style-type: none"> - dont l'aéroport de départ est CDG

• Modification des données : routes POST, PUT, DELETE -----> API privée



• Test des différentes routes

dossier test : /appli_dash/tests

\$ python3 /appli_dash/tests/run_tests_api.py





Remarques

- présence d'un fichier `.env.example` contenant toutes les variables d'environnement à définir pour pouvoir exécuter le docker-compose
- présence d'un fichier `instructions.txt`
- présence d'un script shell `setup.sh` regroupant les différentes commandes à effectuer

4 services :

• MySQL

- Utilisation d'un dockerfile
- Dossier du container `./data_cvs_init/` data des tables en csv
- mysql contient : `./init_sql/sql_files/` scripts SQL pour : créer la base
insérer les données
- Création d'un volume nommé `mysql_data` pour persister les données

• MongoDB

- Utilisation de l'image officielle
- Partage du dossier local `/init_mongodb` fichier d'initialisation `mongo-init.js`
→ création des collections et des index
- Création d'un volume `mongo_data` pour persister les données

• Dash

- Utilisation d'un dockerfile
- Dossier local `appli_dash` comme Workdir `/app`

• Airflow

dockerfile / docker-compose

```
COPY ./data_cvs_init/ /var/lib/mysql-files/data_csv_init/
```

```
COPY ./sql_files/ /docker-entrypoint-initdb.d/
```

```
mysql_data:/var/lib/mysql
```

```
./init_mongodb/mongo-init.js:/docker-entrypoint-initdb.d/mongo-init.js
```

```
mongo_data:/data/db
```

```
WORKDIR /app
```

```
COPY . .
```



• DAGS

