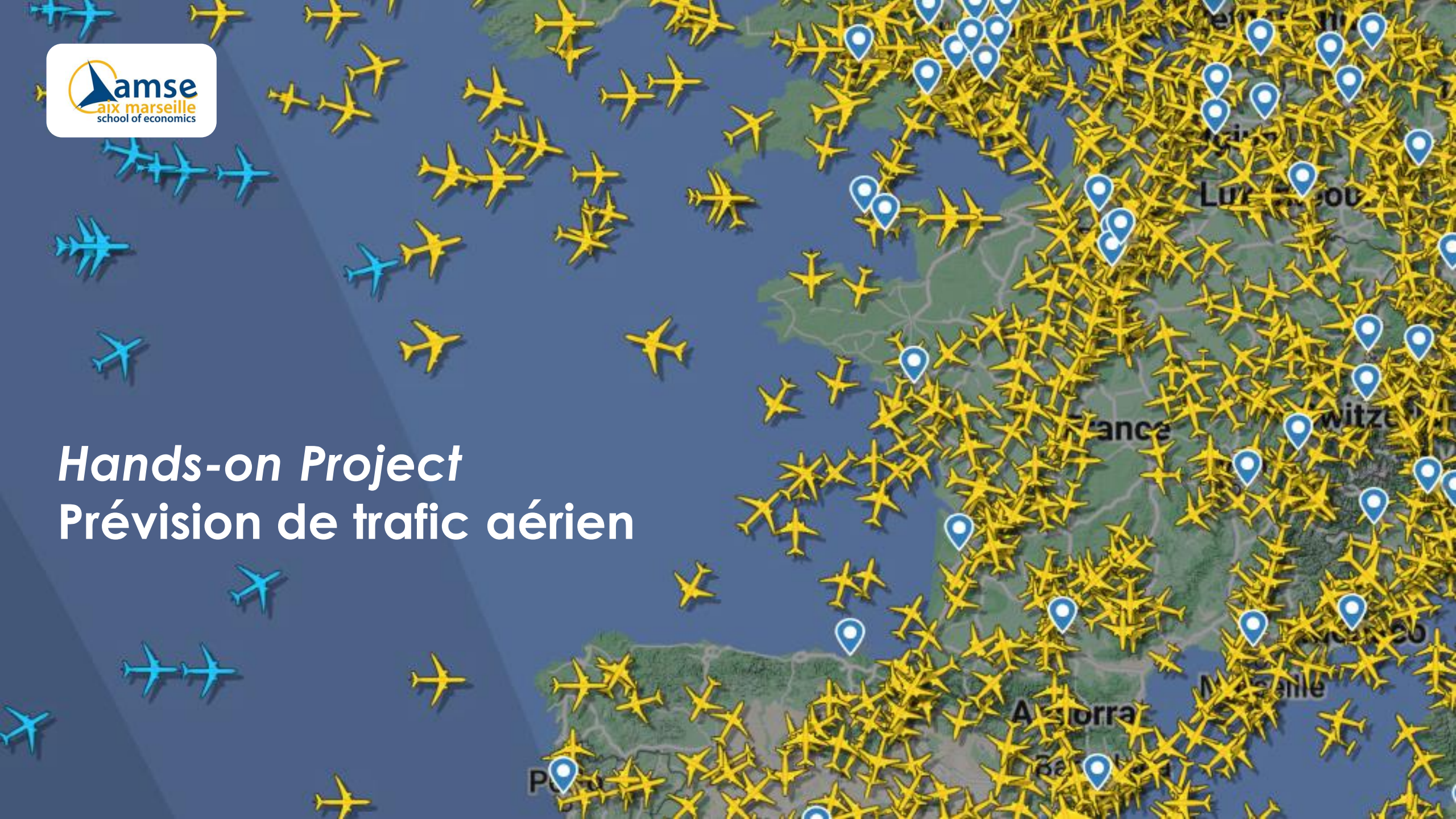


# *Hands-on Project* Prévision de trafic aérien



# Attendus pour le 20 mai 2023

2 (deux) livrables distincts sont attendus :

1. un dossier de présentation de 3 à 4 slides environ (format Powerpoint paysage, fichier pptx ou pdf à votre convenance) traitant le sujet présenté ci-après
2. un *repository* sur Github contenant votre version du projet réalisé ensemble avec les compléments que vous aurez réalisé (liste ci-après), ainsi qu'un lien vers une application streamlit déployée sur Streamlit Cloud (instruction ci-après)

Les travaux sont à réaliser individuellement ou en binôme (pas de groupe de plus de deux personnes).

Vous pouvez réaliser les travaux avec le même binôme ou avec un binôme différent pour chacun d'eux, à votre convenance.

Les éléments sont à envoyer à : [herve.mignot@equancy.com](mailto:herve.mignot@equancy.com)



# Attendu #1 : dossier « Risques & bénéfices de l'IA générative »

Briefing : vous venez de rejoindre une équipe de data science et votre manager vous demande de préparer une présentation des risques et bénéfices de l'utilisation de l'**IA générative** au sein de l'entreprise (*Large Language Models*, génération d'images à partir de texte et inversement, etc.) . Vous ne devez pas faire d'état de l'art, l'équipe connaît déjà le sujet mais votre manager souhaite en savoir plus sur ce qui va être important dans le futur proche.

Vous rédigerez et préparerez 3 à 4 slides pour cette présentation (fictive), selon le plan qui vous semblera le plus adéquat.

Critères d'évaluations :

- Pertinence des enjeux identifiés et présentés
- Cohérence de la présentation (présentation des enjeux et solutions envisageables)

## Attendu #2 : projet prévision de trafic aérien – compléments

Briefing : vous complétez le projet réalisé en session ensemble d'un certain nombre de fonctionnalités parmi la liste ci-après. Des points sont obligatoires, un facultatif qui contribue à une bonification de l'évaluation du travail rendu.

Votre livrable est un lien vers votre repository Github qui doit contenir l'ensemble des éléments que vous avez développé (même si c'est habituellement déconseillé, vous pouvez y joindre le fichier de données).

Ce *repository* pourra être sur l'un ou l'autre des comptes personnels des membres d'un binôme, à votre convenance.

De plus, vous indiquerez un lien vers l'application Streamlit déployée sur Streamlit Cloud (<https://streamlit.io/cloud#plans>, prendre le plan Starter, gratuit).

Critères d'évaluation :

- Réalisation des points obligatoires ci-après
- Qualité du code (respect des conventions vues ensemble lors des sessions)

## Attendu #2 : projet prévision de trafic aérien – compléments

Statut	ID	Description
Obligatoire	#1	Constituer un <i>repository</i> Github contenant l'ensemble des éléments nécessaires, en ajoutant les éléments suivants : <ul style="list-style-type: none"><li>• <code>requirements.txt</code> (cf. la liste des modules installés lors de la création de l'environnement du projet : streamlit, plotly, etc. Utiliser la liste figurant dans les notes ci-après)</li><li>• README.md : mettre à jour avec les informations qui vous semblent importantes (environnement, installation, etc.). Github déborde d'exemples.</li><li>• <code>.gitignore</code> pour empêcher d'ajouter des fichiers et dossiers inappropriés dans un repo (comme les checkpoints de Jupyter)</li></ul>
	#2	<p>Ecrire un notebook d'entraînement d'un modèle <b><i>mettant au propre et reprenant les éléments développés ensemble dans le notebook du repo actuel, vu ensemble</i></b></p> <p>Parmi les fonctions à reprendre et mettre au propre :</p> <ul style="list-style-type: none"><li>• Affichage des données d'une route (homeAirport – pairedAirport) sous forme d'un graphique Plotly</li><li>• Entraînement d'un modèle Prophet pour une route donnée avec une prévision de x jours</li></ul> <p>Ces fonctions sont réutilisées dans le point suivant pour l'application Streamlit.</p>

## Attendu #2 : projet prévision de trafic aérien – compléments

Statut	ID	Description
Obligatoire	#3	<p>Finaliser le développement de votre application <b>streamlit</b> qui devra comporter les éléments suivants :</p> <ol style="list-style-type: none"><li>1. Afficher la série temporelle avec un graphique Plotly comme celui utilisé dans le notebook selon ce qui est sélectionné dans les boutons Home Airport et du Paired Airport de l'application</li><li>2. Ecrire une fonction réalisant une prédiction du trafic, lorsque l'on appuie sur le bouton « Forecast », pour la route sélectionnée (Home Airport – Paired Airport) avec Prophet (reprendre les éléments du notebook pour faire une fonction qui entraîne le modèle)</li><li>3. Afficher le résultat de la prédiction dans le graphique du point 1 : on utilisera une couleur ou une mise en forme (pointillés) pour distinguer les données historiques des données prédites</li></ol> <p>Déployer l'application sur streamlit.io Cloud (cf. introduction) et communiquer le lien vers l'application.</p>

## Attendu #2 : projet identification d'avions – compléments

Statut	ID	Description
Optionnel	#6	<p>Ajouter un sélecteur dans l'application <b>streamlit</b> permettant de choisir quel type modèle utiliser pour la prédiction.</p> <p>Cela nécessite de :</p> <ul style="list-style-type: none"><li>- rajouter un sélecteur de modèle pour choisir parmi les autres modèles vus ensemble dans le notebook (avec Nixtla, <a href="https://docs.streamlit.io/library/api-reference/widgets/st.selectbox">https://docs.streamlit.io/library/api-reference/widgets/st.selectbox</a>)</li><li>- réécrire la fonction calculant la prédiction pour appeler le modèle choisi dans le sélecteur</li></ul>

# Outillage utilisé durant le projet



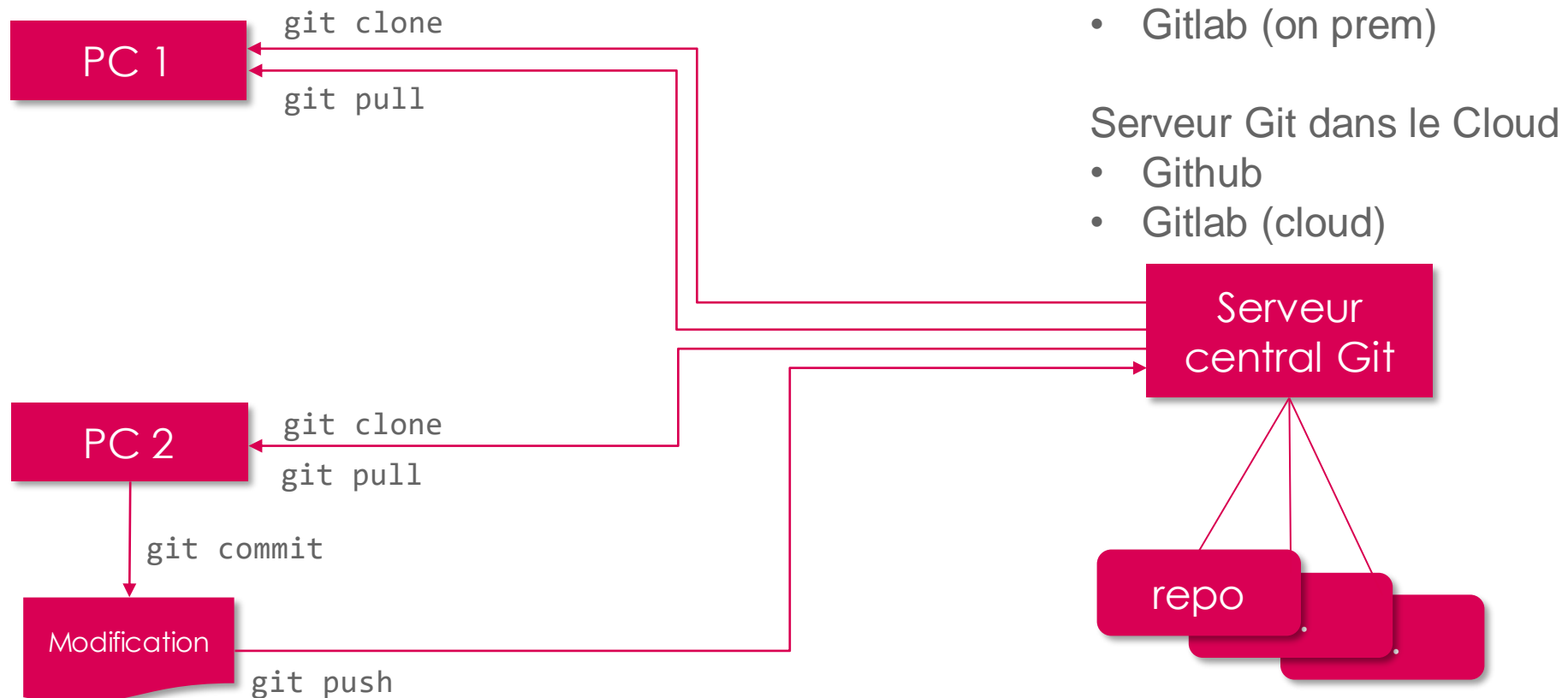




# Architecture git – rappel

Poste local

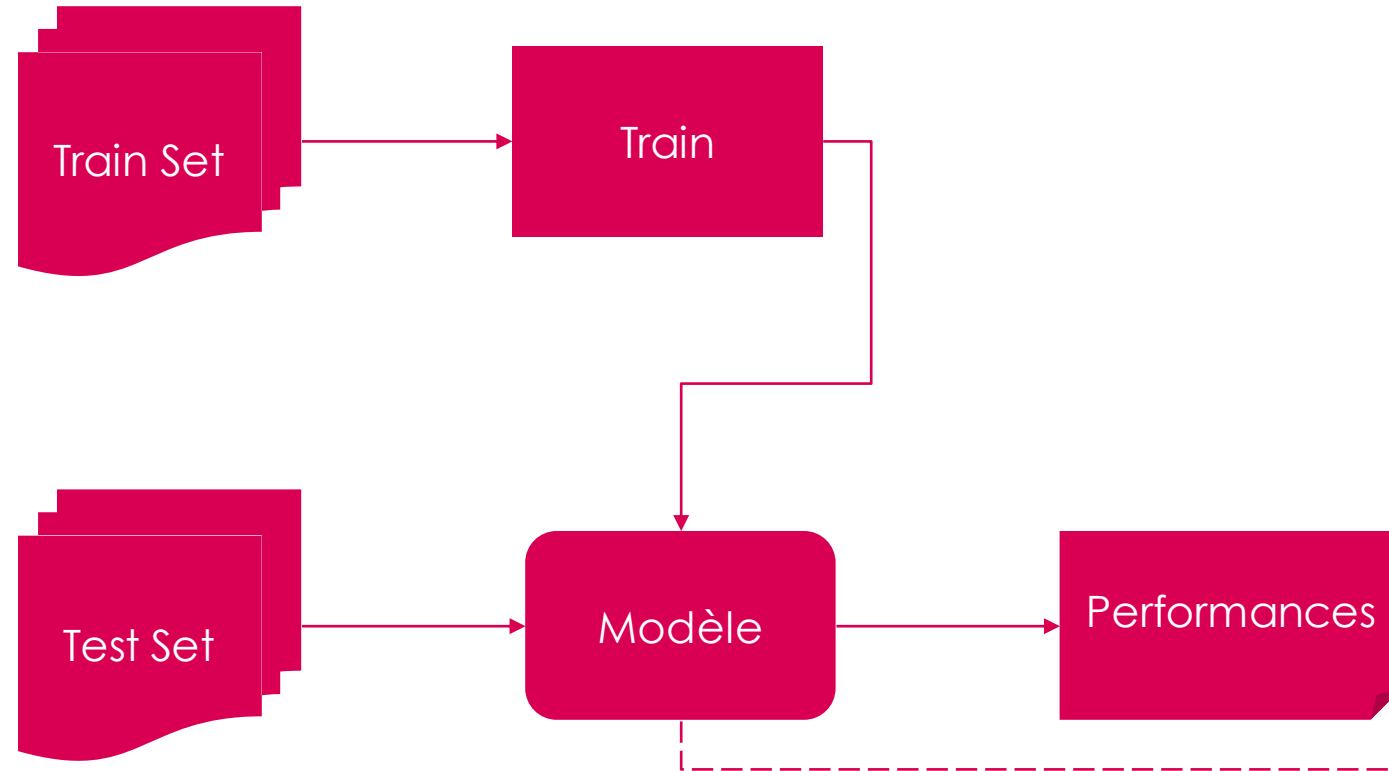
Serveur central



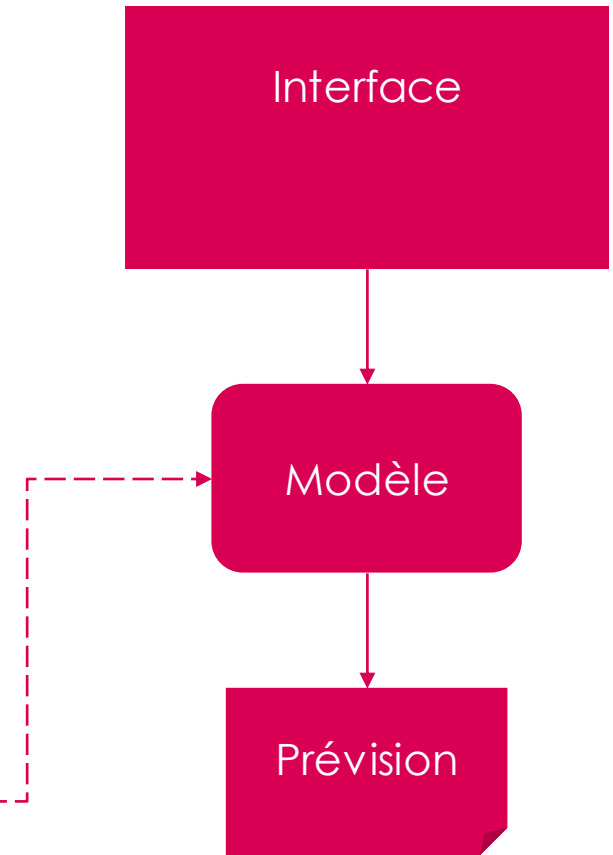


# Première approche de l'architecture

## Entraînement du modèle



## Interface



Questions ?

