MASTER OF SCIENCE
IN ENGINEERING

Cahier des charges - Projet interdisciplinaire

AI Marketplace

Master MSE

Rédigé par : **Andrea Petrucci, Benjamin Pasquier,**
Florian Hofmann, Thibaut Michaud

Supervisé par : Beat Wolf, Jean Hennebert, Sébastien Rumley

Version : 1.0

16 mars 2023

Table des versions

Version	Description
1.0	Version initiale

Table des matières

1 Contexte

Dans le cadre du projet interdisciplinaire du Master MSE, il est demandé aux étudiants de réaliser un projet au sein d'un groupe, combinant plusieurs disciplines de l'ingénierie.

Le projet sélectionné par le groupe est "AI Marketplace". L'objectif principal est la création d'une application de type Marketplace de l'AI basée sur un portefeuille de micro-services encapsulant des modèles Machine Learning. Ces micro-services doivent être facilement interfaçables / assemblables pour créer des applications complètes. Pour donner un exemple concret, l'assemblage d'un module de détection d'objets dans une image, suivi par un module d'image captioning puis par un module text-to-speech permettrait de créer un service « lunettes AI pour malvoyant ».

Les projets doivent inclure différentes dimensions liées au Software Engineering (par exemple le design d'API REST OpenAPI) et au Data Science pour sélectionner et intégrer des modèles d'intelligence artificielle pertinents (NLP, image, parole, etc.). Des aspects « MLOps » seront analysés et développés pour intégrer la mise à l'échelle des services AI (e.g. déploiement Kubernetes) et pour faciliter une intégration continue de ces services (p.ex. MLFlow). Finalement, une interface utilisateur devra être proposée pour permettre à l'utilisateur final de sélectionner et d'assembler des pipelines de traitement des données par IA (p.ex. similaire à Knime).

Chaque groupe doit donc imaginer un cas d'utilisation à mettre en place, représenté par une pipeline (chaînage) constituée de plusieurs micro-services qui peuvent être interfacés les uns aux autres pour effectuer une tâche globale. La pipeline que nous avons choisi de réaliser dans le cadre de ce projet est décrite dans la section suivante.

1.1 Cas d'utilisation

Le cas d'utilisation choisi dans le cadre de notre projet consiste à créer de l'art sous forme d'image à partir d'une musique. Dans un cas concret, cette image pourrait par exemple être utilisée comme pochette de l'album. La pipeline complète composée des étapes intermédiaires pour réaliser ce cas d'utilisation sont décrites par la figure 1.1. Chacune de ces étapes correspond à une tâche qui sera exécutée par un micro-service spécifique, à savoir :

- **Reconnaissance de la parole :** analyse de la voix humaine d'un fichier audio et transcription sous forme textuelle.
- **Analyse de sujet/sentiment :** analyse d'un texte et extraction des sujets et/ou émotions qui y sont exprimés.
- **Détecteur de style de musique :** détection du style musicale d'un fichier audio caractérisant une musique.
- **Générateur d'art :** génération d'image à partir de données textuelles.
- **Extraction de la voix :** extraction de la voix d'un fichier audio en supprimant la musique. Cette tâche n'étant pas primordiale pour l'exécution de la pipeline, le micro-service associé sera développé si le temps le permet.

Notre pipeline pourra être exécutée par l'utilisateur au travers d'une application web lui permettant d'importer un fichier audio à traiter. Il recevra en réponse une ou plusieurs images caractérisant le style de sa musique et, si elle contient des paroles, les sujets ou sentiments qui y sont transmis.

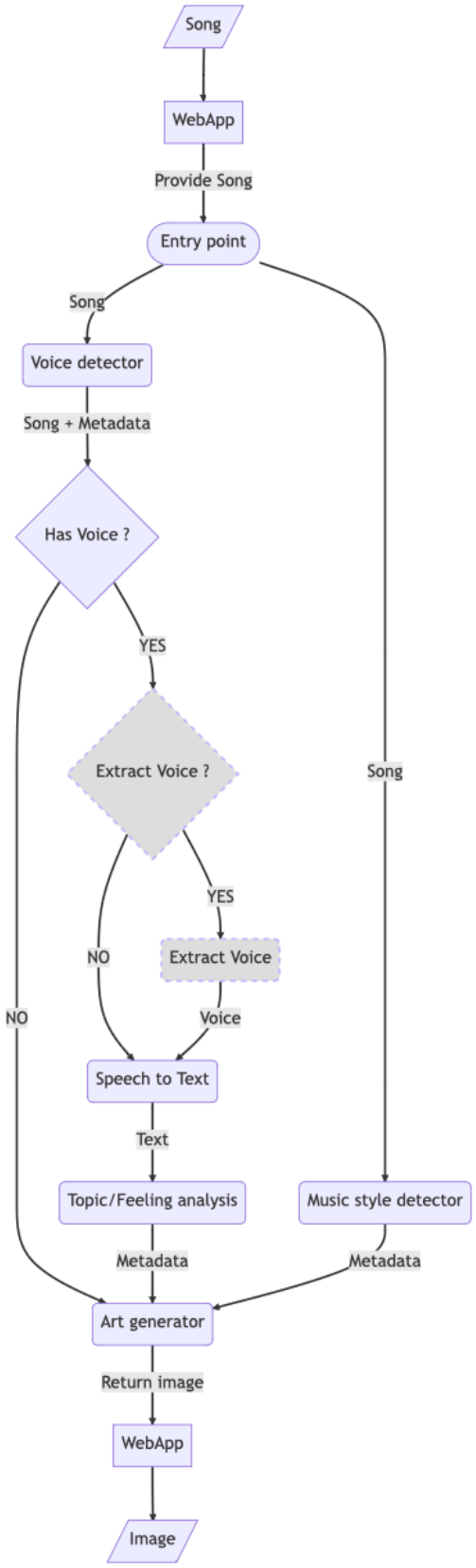

Figure 1.1 – Flowchart du use case

2 Objectifs

En plus des objectifs fixés par le Master, le projet en comporte certains plus spécifiques :

- Avoir un ou plusieurs workflow(s) démontrable(s) en fin de projet ; un workflow est composé au minimum de 2 microservices ;
- Utiliser des outils CI/CD pour le déploiement ;
- Utiliser Kubernetes comme plate-forme de déploiement - ou autre plate-forme permettant la distribution et mise à l'échelle des calculs ;
- Au moins un des modèles ML doit pouvoir être ré-entraînable automatiquement à travers les outils CI/CD (i.e. MLOps) ;
- Proposer une organisation du travail en groupe ;
- Individuel : proposer un ensemble de compétences à développer lors du projet ;

Pour le dernier point, les membres du groupe ont choisi de développer les compétences suivantes :

- **Andrea**
  - Approfondir les connaissances en Machine Learning.
  - Découvrir les pratiques MLOps pour la création, le réentrainement et le déploiement d'un modèle.
  - Apprendre à utiliser le framework de frontend VueJS.
- **Florian**
  - Consolider les compétences en DevOps, notamment en utilisant la plateforme GitHub.
  - Approfondir les connaissances en Machine Learning notamment en ce qui concerne la reconnaissance de parole.
  - Découvrir les pratiques MLOps.
- **Benjamin**
  - Découvrir et mettre en place les pratiques de MLOps.
  - Approfondir et consolider les compétences en terme de création, d'entraînement et d'évaluation d'un modèle de Deep Learning ainsi que son encapsulation dans un micro-service.
- **Thibaut**
  - Découvrir et mettre en place un outil d'orchestration de workflow afin de connecter les différents micro-services.
  - Découvrir la génération d'image et approfondir les connaissance en Machine Learning ainsi que l'encapsulation d'un modèle dans un micro-service.

3 Technologies

Ce chapitre décrit les différentes technologies que nous utiliserons pour atteindre les objectifs définis dans notre projet. Les technologies concernant les modèles de Machine Learning utilisées dans les différents micro-services ne figurent pas ici puisqu'elles feront l'état d'une analyse ultérieure.

3.1 MLOps

DVC - `https://dvc.org/`
— Versioning des données
— Pipeline ML

Weights & Biases - `https://wandb.ai/site`
— Tracking et versioning des expériences

MinIO - `https://min.io/`
— Stockage des données transitant entre les micro-services et celles des modèles

3.2 DevOps

Github - `https://github.com/`
— Stockage et versioning du projet
— CI/CD et tests

Prefect - `https://www.prefect.io/`
— Orchestration des micro-services

Kubernetes - `https://kubernetes.io/`
— Déploiement

3.3 Application web frontend

VueJS - `https://vuejs.org/`
— Framework frontend

4 Tâches

Ce chapitre décrit les différentes tâches permettant d'atteindre les objectifs fixés dans ce projet. Elles sont catégorisées dans les sections suivantes, selon le domaine auquel elles sont liées.

4.1 DevOps et MLOps

- Mettre en place les fichiers de déploiement Kubernetes.
- Mettre en place une pipeline CI/CD exécutant une série de tests ainsi que le déploiement des services et de l'application web frontend sur Kubernetes.
- Mettre en place le versioning et le stockage des données nécessaires au modèle entraîné.
- Mettre en place le tracking, le versioning d'expériences liés à l'entraînement du modèle.
- Mettre en place une pipeline MLOps pour l'entraînement et le déploiement du modèle conçu pour l'un des micro-services.

4.2 Micro-services

- Sélectionner un modèle de reconnaissance de parole et développer le micro-service l'encapsulant.
- Sélectionner un modèle d'analyse de sujets/émotions et développer le micro-service l'encapsulant.
- Sélectionner un modèle de génération d'art et développer le micro-service l'encapsulant.
- Créer, entraîner et évaluer le modèle de détection de style musical.
- Développer le micro-service encapsulant le modèle de détection de style musical.

4.3 Intégration des services

- Orchestrer les services pour former la pipeline qu'illustre la figure 1.1.
- S'assurer que tous les services développés respectent les spécifications requises pour l'intégration à la plateforme CSIA-PME.

4.4 Application web frontend

- Concevoir des maquettes d'interface utilisateur.
- Développer l'application web servant d'interface utilisateur.

5 Planning

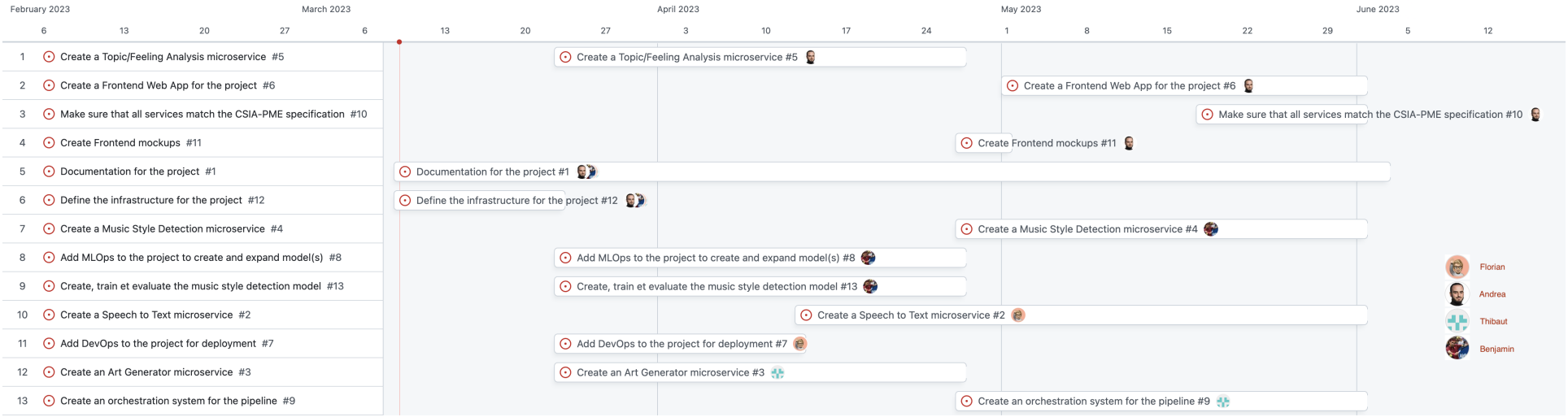

Figure 5.1 – Gantt