

Dossier de cadrage

1. Étude d'art / benchmark :

Dans le cadre de notre projet d'applications nous avons utilisé plusieurs API tels que Groq, Clip Drop et Mistral. Il est maintenant de tant d'étudier leurs utilisations plus en détail.

API	Fonction	coût(dev)	Temps de réponse	Quota	RGPD
Groq	Transcription audio	Gratuit	Très rapide	pas de quota	US
ClipDrop	Génération d'image	0,10 centime par image	5 à 10 secondes par images	100 crédit environ	FR RGPD-compliant
Mistral	LLM résumé des émotions	Gratuit	Très rapides	Limité sans abonnement	US non conforme
DeepL API	Traduction automatique	20€/mois	Très rapide	Gratuit (500K caractères),	Conforme RGPD (serveurs EU)
replicate	Génération d'image via modèle SDXL	Environ 0,02 \$ par image	5 à 20 secondes	Dépend du modèle utilisé	US non conforme
Assembly ai	Transcription audio avancée	0,00025 \$/seconde, soit 0,90 \$/heure En supplément selon l'option choisie (ex.	1 à 2 secondes/audio	5h/mois gratuites, puis payant	US non conforme

		: 0,0002 \$/seconde)			
--	--	-------------------------	--	--	--

2. UML

Avant de concevoir le synthétiseur de rêve il était primordial de pouvoir visualiser et concevoir son développement pour ce faire nous avons conçu un UML.

title Traitement d'un rêve : de l'audio à l'image

actor Utilisateur

participant "Interface Streamlit" as Streamlit

participant "Fichier temporaire" as TempFile

participant "Groq Whisper" as STT

participant "Mistral (Emotion)" as Emotion

participant "ClipDrop API" as ClipDrop

participant "Backend.py" as Backend

Utilisateur -> Streamlit : Upload fichier audio (.m4a/.wav)

Streamlit -> TempFile : Sauvegarde sous temp_audio_file.xxx

Streamlit -> Backend : speech_to_text(temp_audio_file, "fr")

Backend -> STT : Transcription audio avec Whisper

STT --> Backend : Texte du rêve

Backend --> Streamlit : Texte du rêve

Streamlit -> Backend : text_analysis(texte)

Backend -> Emotion : Analyse via Mistral API

Emotion --> Backend : JSON émotions

Backend --> Streamlit : Top 2 émotions avec softmax

Streamlit -> Backend : text_to_image(texte)

Backend -> ClipDrop : Appel API avec prompt

ClipDrop --> Backend : Image (bytes)

Backend --> Streamlit : Image encodée

Streamlit -> Utilisateur : Affichage texte + émotions + image.

Conclusion :

En conclusion Notre application repose sur une chaîne automatisée permettant de transformer un enregistrement vocal en image onirique, en passant par une transcription textuelle et une analyse émotionnelle. Le coût total par utilisateur repose exclusivement sur l'appel à l'API ClipDrop, facturé à 0,10 € par rêve généré. Grâce à l'utilisation d'outils performants et actuellement gratuits comme Groq pour la transcription audio et Mistral pour l'analyse émotionnelle, notre solution reste économiquement viable à petite échelle. Le temps de traitement global, transcription comprise, varie entre 10 et 30 secondes selon la durée de l'audio et la complexité de l'image générée. À terme, la soutenabilité du projet dépendra du maintien de la gratuité des services Groq et Mistral ou, à défaut, d'une stratégie de migration vers des alternatives auto-hébergées ou européennes, plus conformes au RGPD et plus économes à grande échelle.