# Documentation Technique

## 1. Architecture Système

### Pipeline Principal :

* Module 1 : Capture audio avec détection de silence.
* Module 2 : Reconnaissance vocale (Speech-to-Text).
* Module 3 : Traduction (Text-to-Text).
* Module 4 : Synthèse vocale (Text-to-Speech).
* Communication inter-module via une file d’attente (queue).

## 2. Technologies Utilisées

### Google APIs :

* speech\_v1p1beta1.SpeechClient: Reconnaissance vocale.
* translate\_v2.Translate.Client: Traduction.
* texttospeech.TextToSpeechClient: Synthèse vocale.

### Bibliothèques Complémentaires :

* sounddevice: Capture et lecture audio.
* numpy: Manipulation des données audio.
* html: Décodage des entités dans les textes traduits.

## 3. Schéma d’Architecture

….

## 4. Déploiement

### Étape 1 : Configuration Google Cloud

* Activer les APIs nécessaires dans le projet Google Cloud.
* Télécharger les clés JSON du service Google Cloud.

### Étape 2 : Installation des Dépendances

pip install google-cloud-speech google-cloud-translate google-cloud-texttospeech sounddevice numpy

### Étape 3 : Lancer le Script

python translator.py

## 5. Plan de Tests

### Tests Unités :

* Vérifier la transcription de phrases courtes et longues.
* Tester la qualité de la traduction avec des phrases complexes.

### Tests Intégration :

* Mesurer la latence totale du pipeline.
* Simuler des interruptions réseau et valider la reprise.

### Tests Utilisateurs :

* Recueillir les retours des utilisateurs sur la clarté des traductions et la fluidité.

## 6. Applications Futures

* Utiliser Gemini pour améliorer la contextualisation des traductions.
* Déployer sur des dispositifs portables ou embarqués.
* Ajouter un mode de transcription collaborative en ligne.