Technical Documentation

## 1. System Architecture

### Main Pipeline:

* Module 1 : Audio capture with silence detection.
* Module 2 : Speech recognition (Speech-to-Text).
* Module 3 : Translation (Text-to-Text).
* Module 4 : Speech synthesis (Text-to-Speech).
* **Inter-module communication** via a queue.

## 2. Technologies Used

### Google APIs:

* speech\_v1p1beta1.SpeechClient: Speech recognition.
* translate\_v2.Translate.Client: Translation.
* texttospeech.TextToSpeechClient: Speech synthesis

**Complementary Libraries:**

* sounddevice: Audio capture and playback.
* numpy: Audio data manipulation.
* html: Decoding entities in translated texts.

## 3. Architecture Diagram

….

## 4. Deployment

### Step 1: Google Cloud Configuration

* Enable the required APIs in the Google Cloud project.
* Download the JSON service keys from Google Cloud.

### Step 2: Dependency Installation

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

### Step 3: Run the Script

python translator.py

## 5. Testing Plan

### Unit Tests:

* Verify the transcription of short and long sentences.
* Test translation quality with complex sentences.

### Integration Tests:

* Measure the total latency of the pipeline.
* Simulate network interruptions and validate recovery.

### User Tests:

* Collect user feedback on translation clarity and smoothness.

## 6. Future Applications

* Leverage Gemini to enhance translation contextualization.
* Deploy on portable or embedded devices.
* Add an online collaborative transcription mode.