

# Voice-to-Text Meeting Minutes Generator - Full System Architecture

## System Architecture: Voice-to-Text Meeting Minutes Generator

### 1. User Interaction Layer (Frontend / CLI)

#### Function:

- Upload .mp3/.mp4 audio files
- Validate and send to S3
- Notify users about upload status

#### Technologies:

- Python CLI or Flask/Streamlit
- Output: File + Metadata (filename, uploader, timestamp)

### 2. Storage Layer - Amazon S3

#### Buckets:

- Upload Bucket (raw audio)
- Transcript Output Bucket (AWS Transcribe JSON)
- Clean Transcript Bucket (Post-processed)
- Summary Output Bucket (TXT, PDF, JSON)

### 3. Event-Driven Processing with AWS Lambda

Lambda 1: Trigger Transcribe on upload

## Lambda 2: Trigger post-processing on transcript output

### Responsibilities:

- Automate Transcribe job creation
- Trigger post-processing and summarization pipeline
- Monitor job status or failures

## 4. Transcription Engine - AWS Transcribe

- Converts audio to text with speaker labels
- Outputs timestamped JSON
- Configurable language and format

## 5. Transcript Post-Processing Module

### Responsibilities:

- Clean filler words like "uh", "um", "you know"
- Remove noise tags like [inaudible], [crosstalk]
- Fix punctuation and formatting
- Add speaker labels and timestamps
- Format as paragraphs or bullet points
- Output: Cleaned JSON, TXT

### Technologies:

- Python (NLTK, spaCy, regex)

## 6. Summarization Engine (NLP)

#### Responsibilities:

- Generate summaries, action items, decisions
- Extract participants, duration, keywords
- Apply NLP to clean transcript

#### Techniques:

- TextRank (graph-based extractive)
- spaCy/NLTK (NER, keyword extraction)
- GPT API (abstractive summarization)

#### Output:

- summary.json
- summary.pdf (using ReportLab or FPDF)
- summary.txt

### 7. Analytics Dashboard

#### Responsibilities:

- View/search/download summaries and transcripts
- Visualizations: speaker talk-time, keyword clouds
- Filter by date, keywords, speaker
- Download as PDF/TXT/JSON

#### Technologies:

- Flask or Streamlit
- Optional: Login/auth with Flask-Login

## 8. Data Flow Overview

User Upload -> S3 -> Lambda -> AWS Transcribe -> Transcript Bucket ->

Lambda -> Post-Processing -> Clean Bucket -> NLP Summary -> Summary Bucket -> Dashboard

## 9. AI & DS Engineer Role

Focus Areas:

- Module 3: Transcript Post-Processing
- Module 4: Summarization Engine
- Optional: Analytics Dashboard (speaker stats, NER analysis)

Tasks:

- NLP cleaning, formatting
- Summary generation using TextRank/spaCy/GPT
- Action item and decision detection
- Format outputs for UI consumption