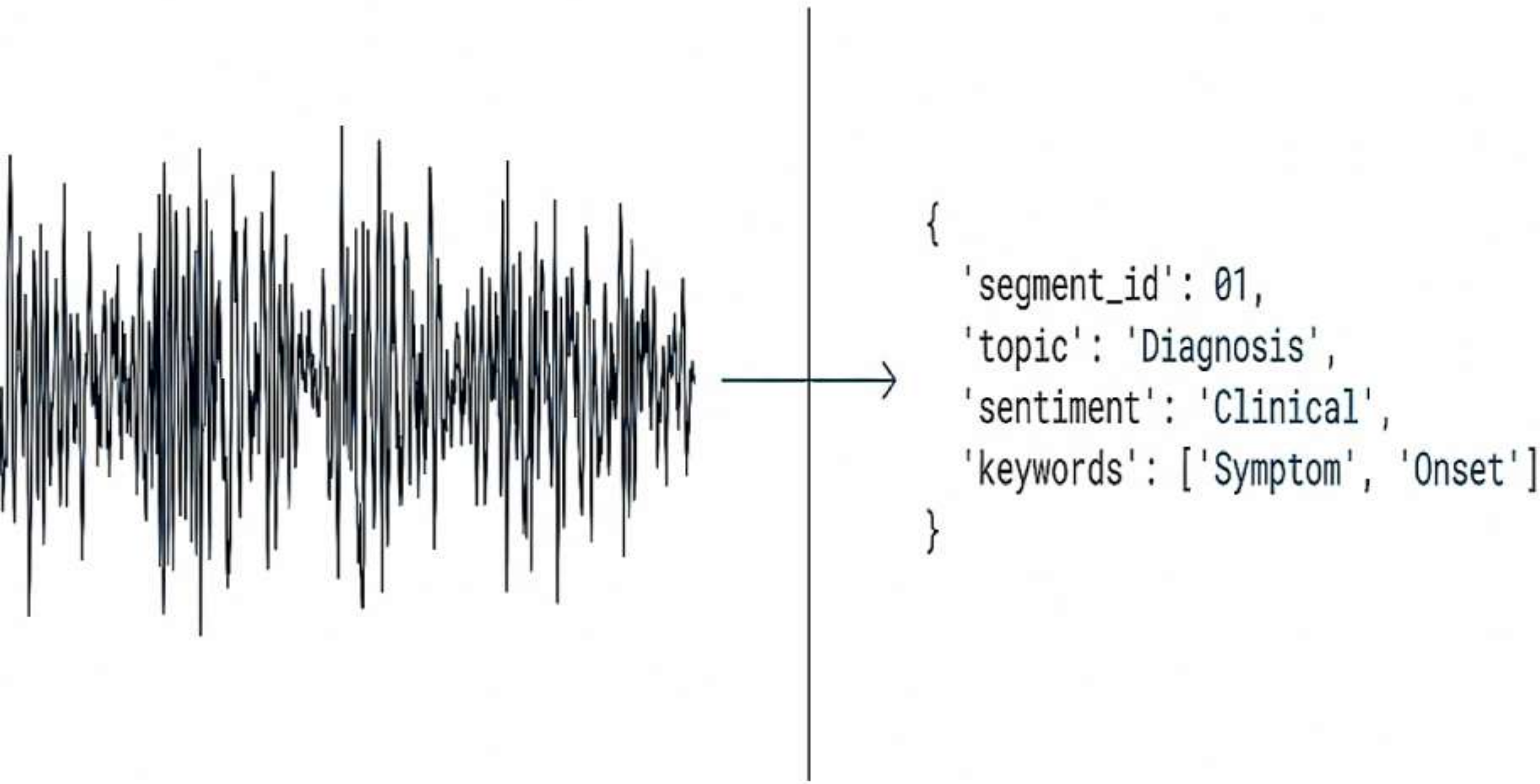
# Automated Medical Podcast Transcription and Topic Segmentation

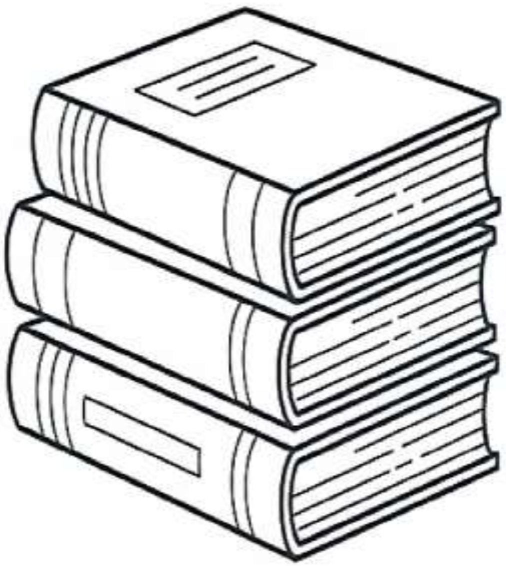
Presented by
Kunchakuri Bharadwaj

# Automated Medical Podcast Transcription and Topic Segmentation

Unlocking unstructured medical knowledge through AI-driven analysis.



# The Unstructured Wealth of Medical Audio



Static Knowledge

## The Resource

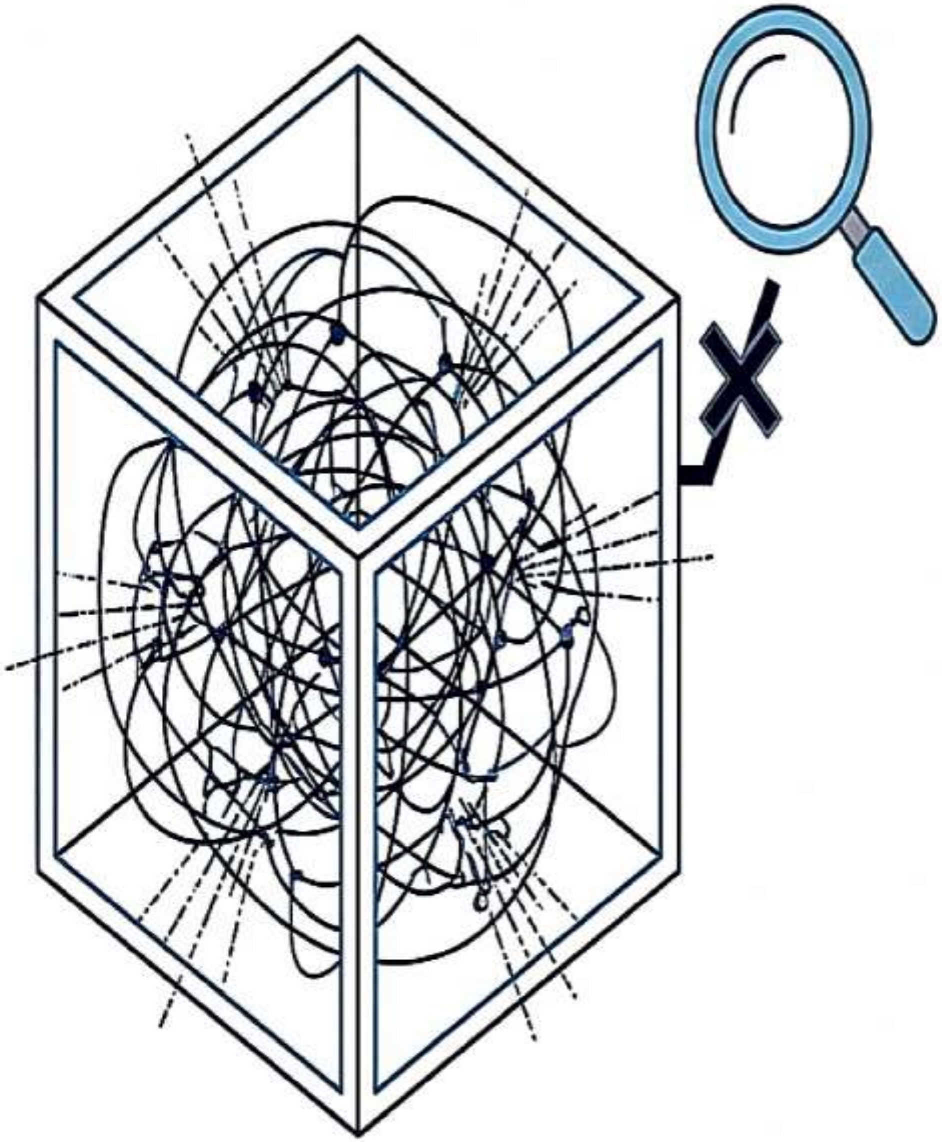
Medical podcasts have emerged as critical information sources for students and professionals. Unlike static textbooks, they offer conversational, real-world insights, covering diverse topics from diagnostic methods to clinical experiences.

Real-world Clinical Insight

## The Conflict

Despite their value, these assets remain locked in linear, audio-only formats. This makes rapid reference and academic review inefficient compared to text-based resources.

# The Accessibility Bottleneck



## The "Black Box" Problem

Medical podcasts are often lengthy and unstructured. Accessing specific information requires listening to entire episodes.

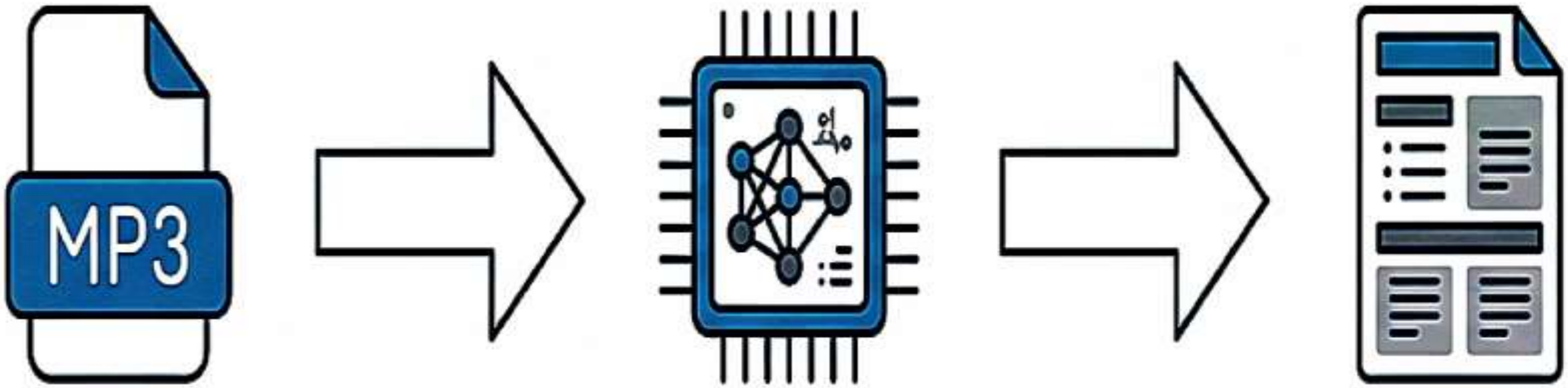
## The Consequence

Valuable clinical data remains unsearchable, time-consuming to extract, and difficult to reuse for research.

## The Objective

To transform raw audio into a format that is segmented, searchable, and actionable.

# Intelligent Transformation of Medical Audio




**Input** — **Process** — **Output**

**Transcription**

Converting speech to raw text.

**Topic Segmentation**

Grouping semantically related sentences into coherent sections.

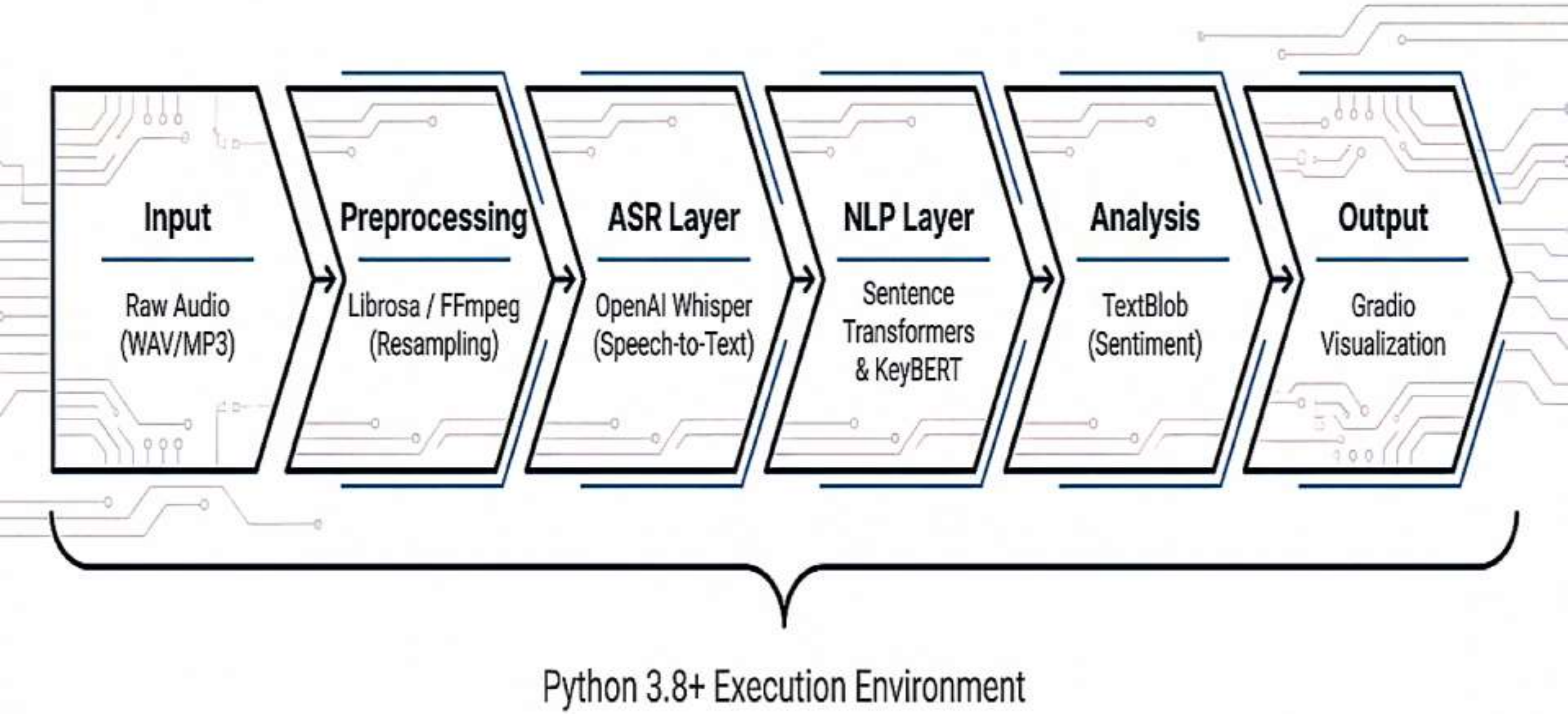
**Insight Extraction**

Identifying medical keywords and sentiment analysis.

**Interface**

Interactive dashboard for navigation.

# The Processing Pipeline Architecture



# Step 1: Preprocessing and Standardisation

## The Challenge

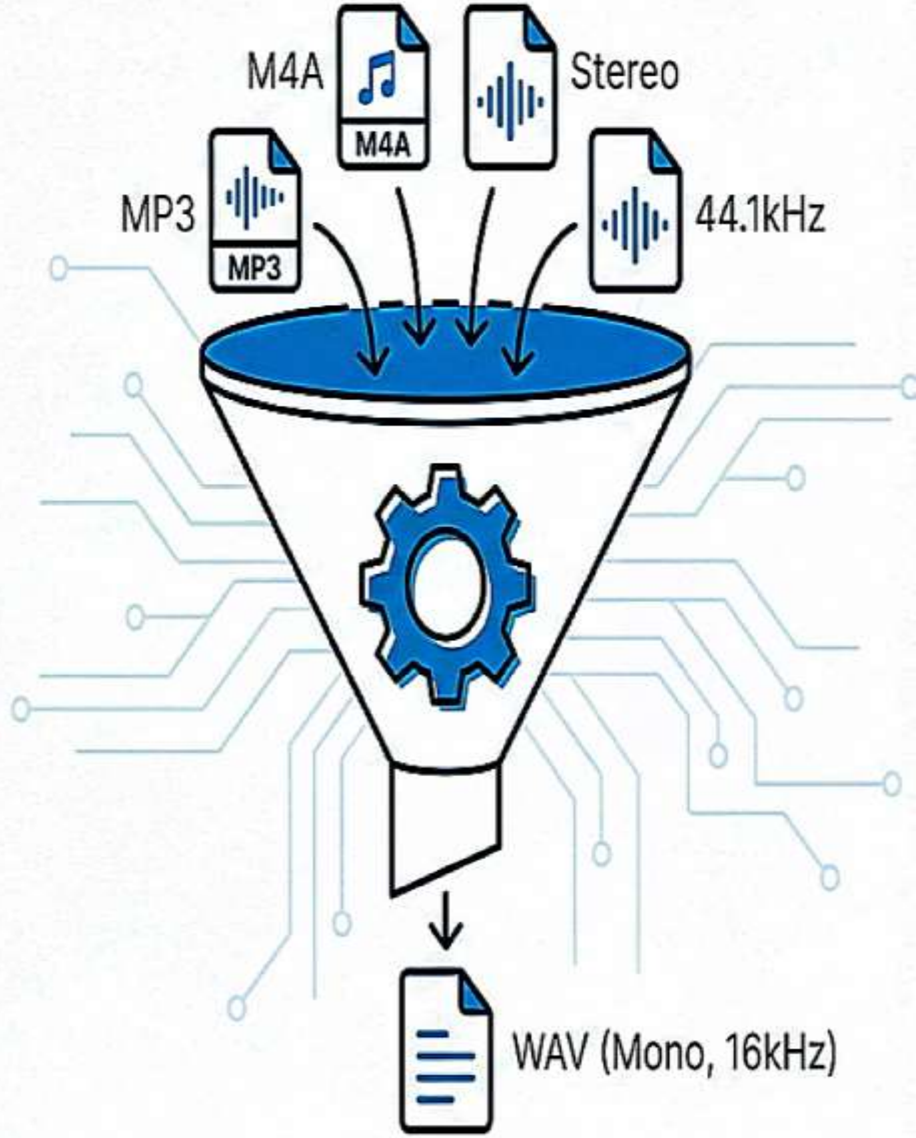
Raw audio files vary in format, quality, and sample rates. Inconsistent inputs lead to failures in speech recognition models.

## The Solution

A rigorous compatibility layer ensures the "canvas is clean" before processing begins.

## Tools & Functions

- **Librosa**: Audio loading, resampling to 16kHz, and normalisation.
- **FFmpeg**: Format conversion (Standardising MP3/M4A to WAV).
- **SoundFile**: Reading/Writing operations.



# Step 2: High-Fidelity ASR with OpenAI Whisper

## The Engine

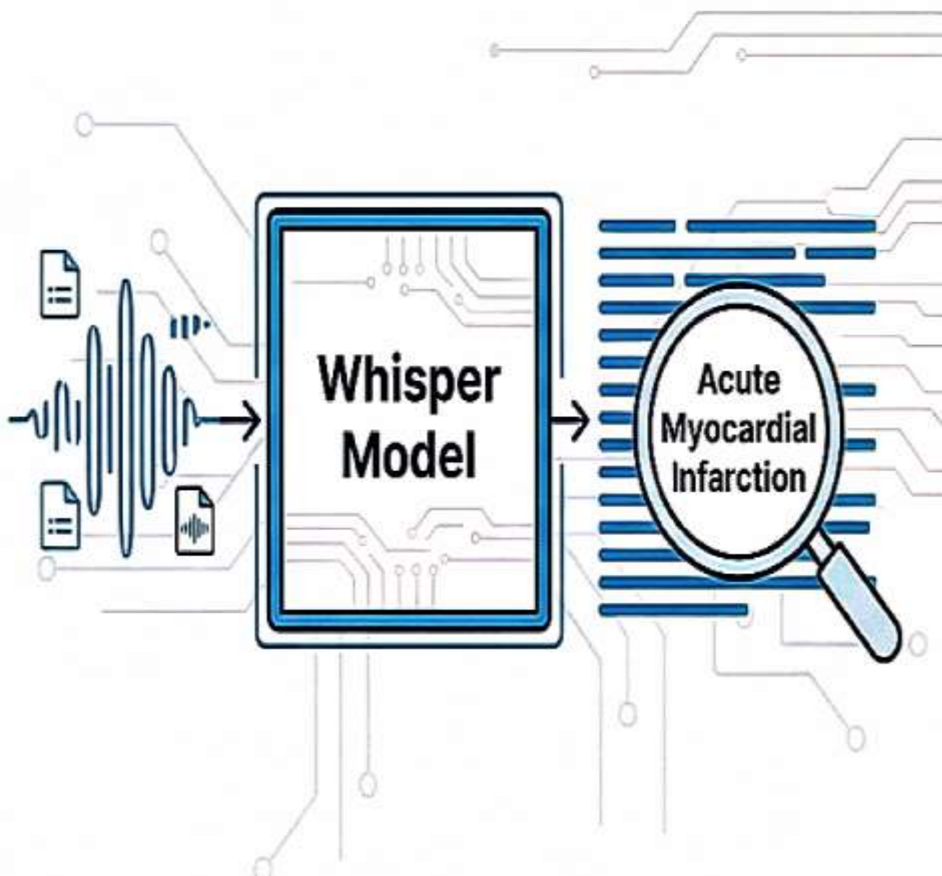
The system utilises the OpenAI Whisper (small model) architecture for Automatic Speech Recognition.

## Why Whisper?

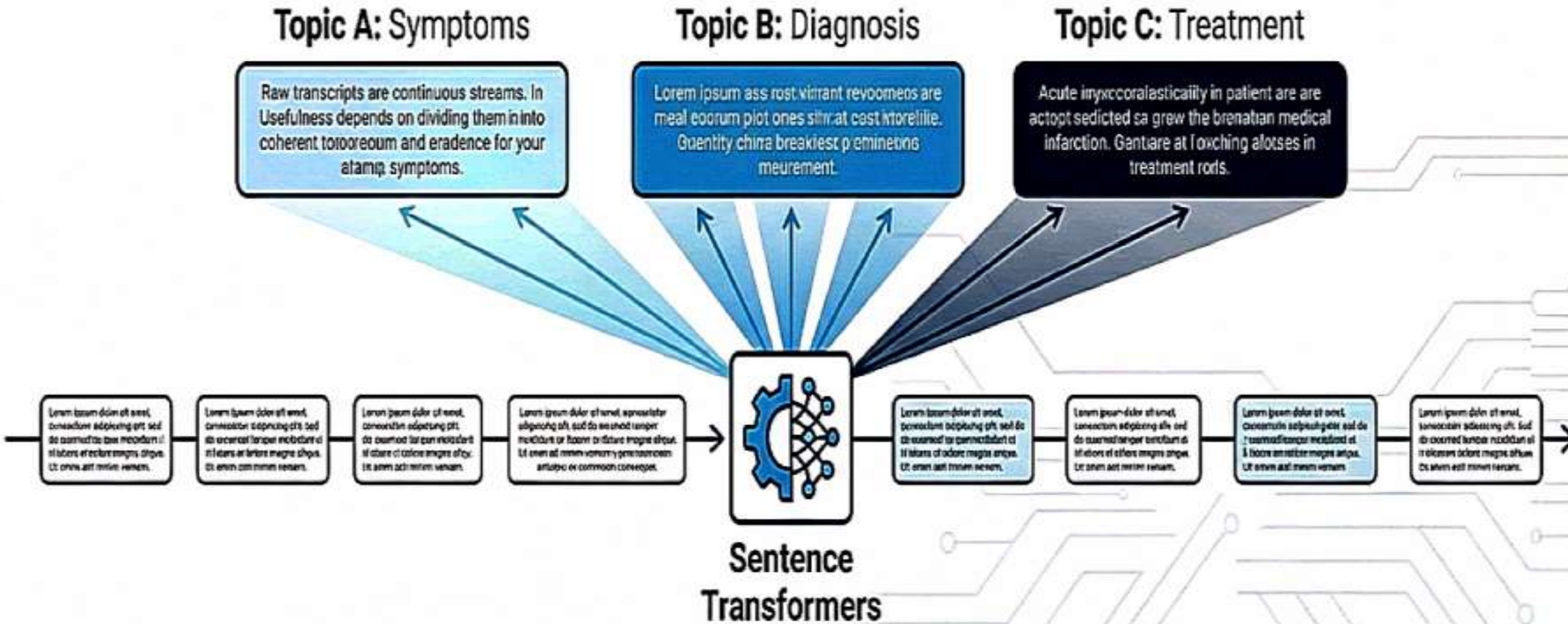
- **Robustness:** Handles long-duration audio effectively.
- **Medical Lexicon:** Superior performance with complex medical terminology (drug names, diseases) compared to standard models.

## Process

Preprocessed audio is ingested and transcribed into raw text, ready for semantic analysis.



# Step 3: Semantic Topic Segmentation




## The Logic

Raw transcripts are continuous streams. Usefulness depends on dividing them into coherent topics.

## The Mechanism

- **Tokenisation (NLTK):** Breaks transcript into sentences.
- **Embeddings:** Calculates semantic similarity between sentences.

## The Result

Clustering algorithms group related sentences, creating distinct 'topic blocks' rather than a wall of text.

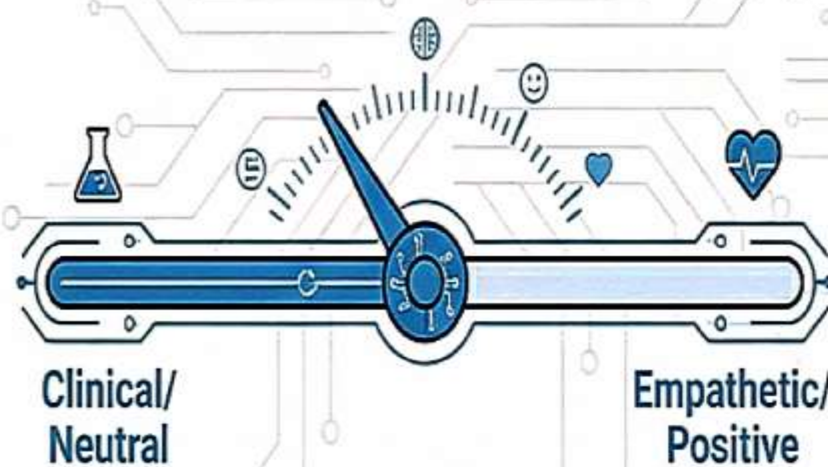
# Step 4: Extracting Insight and Tone

## Context-Aware Keywords

Using **KeyBERT**, we extract specific medical terms relevant to the context of each segmented topic, avoiding generic stopwords.


## Sentiment Analysis

Using **TextBlob**, the system analyses emotional tone. It classifies discussion polarity to distinguish between clinical fact-stating and empathetic patient discussions.



# The Integrated Technology Stack

### Language
Python 3.8+

### Audio Layer
Librosa
FFmpeg
SoundFile

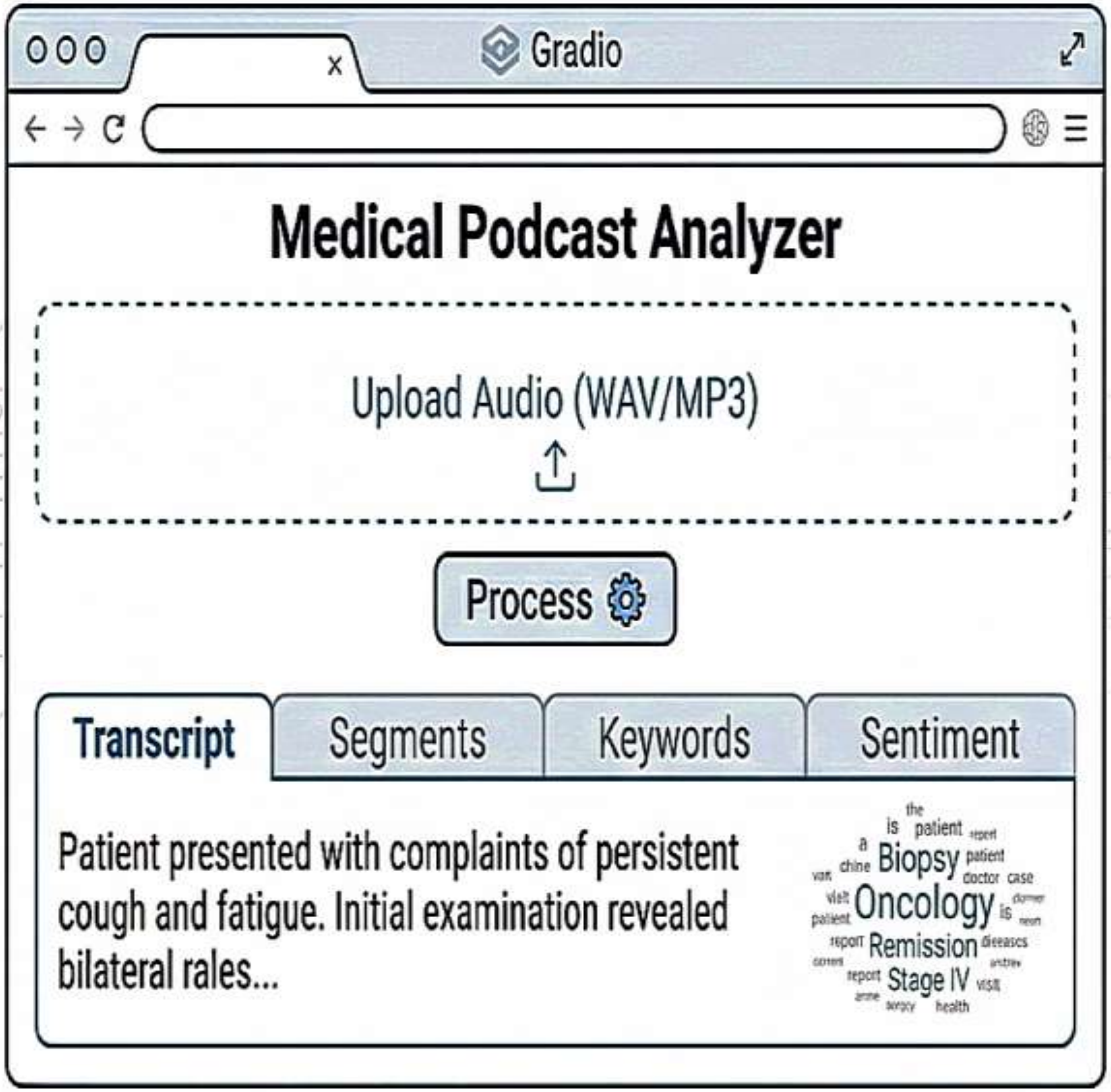
### NLP & ML
OpenAI Whisper (ASR)
Sentence Transformers
KeyBERT

### Analysis
TextBlob (Sentiment)
NLTK (Tokenisation)

### Visualisation
Matplotlib
WordCloud

### UI & Env
Gradio
Jupyter Notebook

# The Interactive User Experience



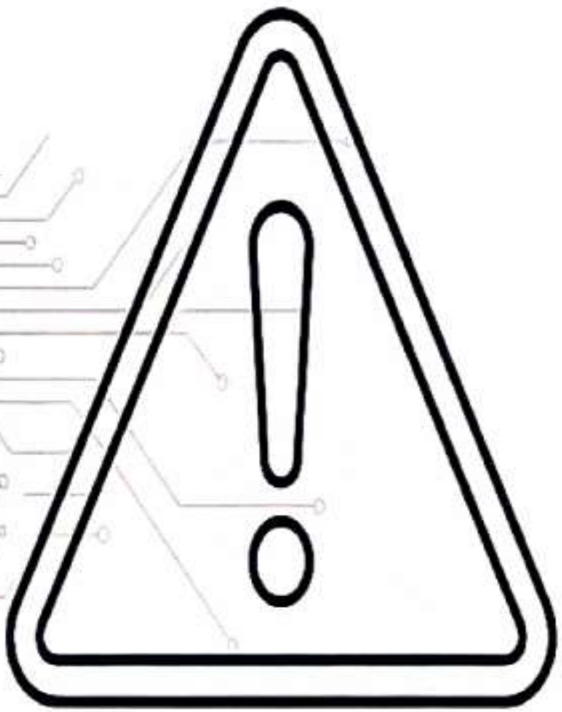

## Workflow

1. **Upload:** User submits audio file.
2. **Process:** System executes pipeline automatically.
3. **Explore:** Users navigate results via tabs for Transcripts, Segments, Keywords, and Sentiment.

# Optimisation and Troubleshooting

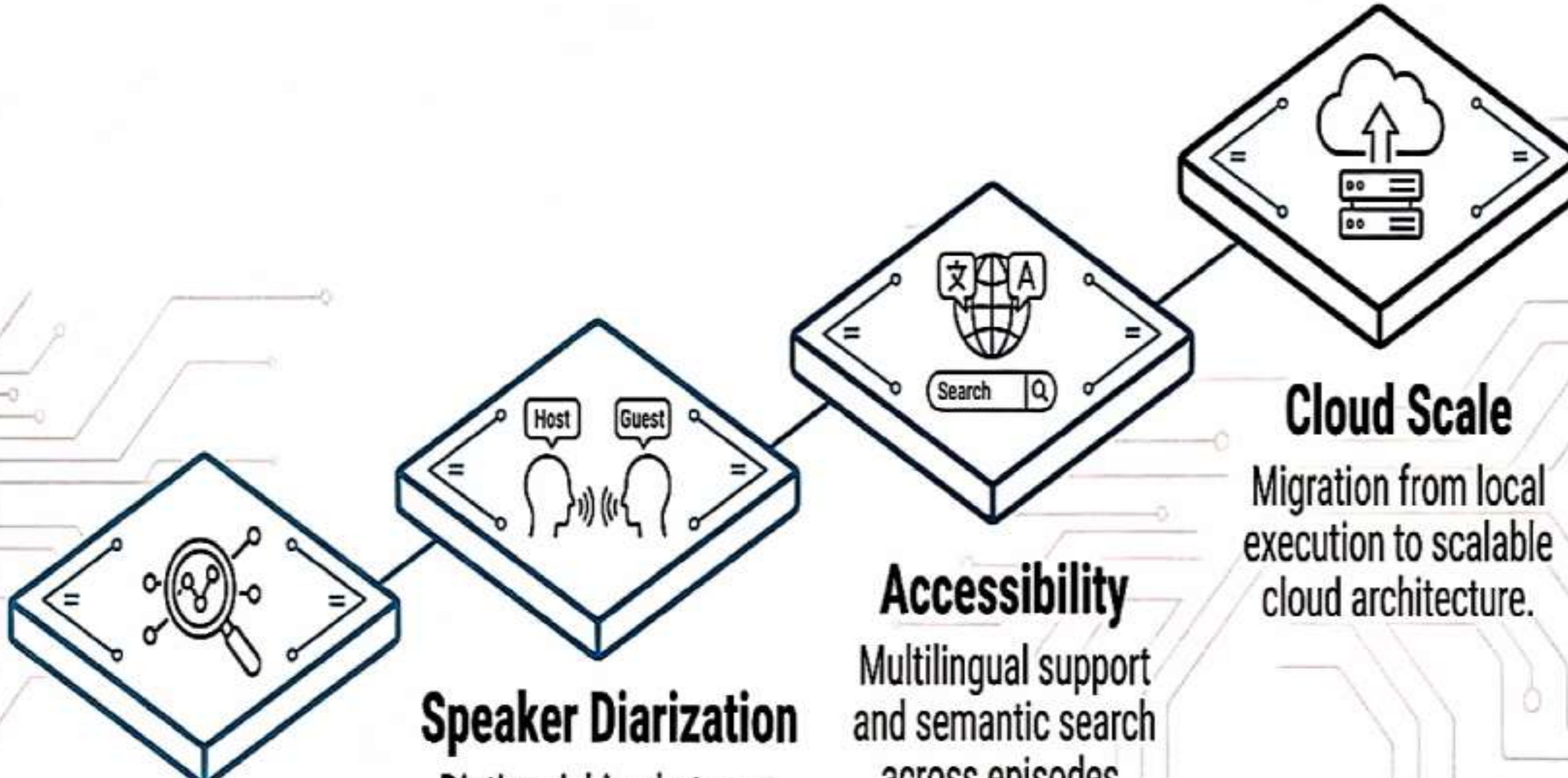
| Challenge | Mitigation Strategy |
| --- | --- |
| Audio Hygiene & Quality | Require standard formats (WAV/MP3). Minimise background noise for better ASR accuracy. |
| Performance Latency | Initial execution includes model downloads (Whisper). Caching is used for subsequent runs. |
| Processing Failures | Integrated logging tracks errors (e.g., insufficient text for word clouds) to assist debugging. |

# Critical Limitations and Constraints



- **No Real-Time Processing**: The system is designed for recorded audio files only; no live stream support.
- **Accuracy Dependencies**: Performance relies heavily on speaker clarity and pronunciation.
- **Contextual Nuance**: Segmentation may struggle with the undefined boundaries of highly casual conversation.
- **Disclaimer**: This tool is for educational and informational purposes only. It does not provide medical diagnoses.

# Future Roadmap and Scalability




**Medical NER**
Integration of Named Entity Recognition to automatically tag drugs and diseases.

**Speaker Diarization**
Distinguishing between multiple speakers (Host vs. Guest).

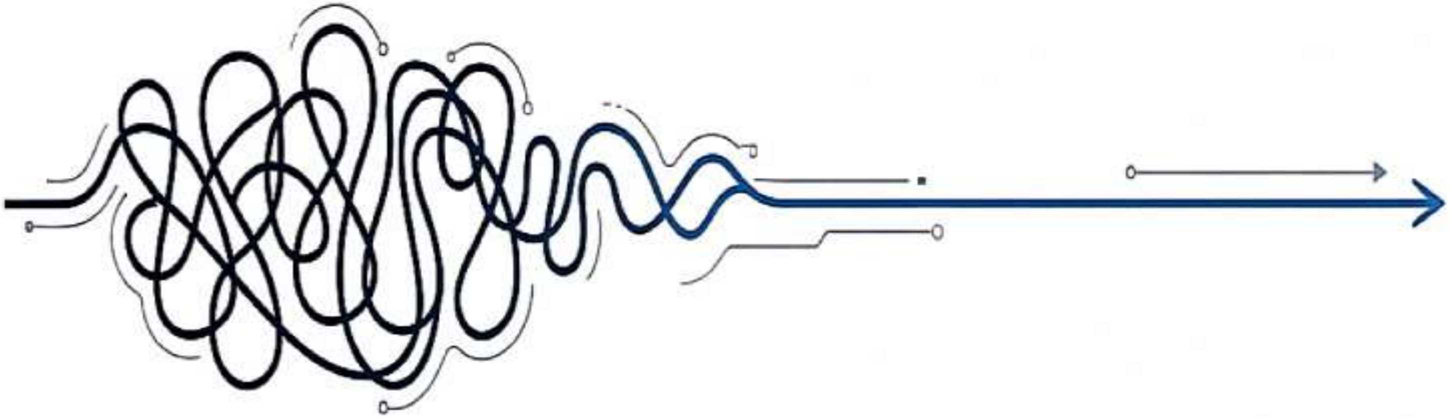
**Accessibility**
Multilingual support and semantic search across episodes.

**Cloud Scale**
Migration from local execution to scalable cloud architecture.

# The Intelligent Transformation is Complete

## Summary

By leveraging ASR and NLP, we have successfully converted unstructured audio into a navigable, knowledge-rich asset, empowering medical education.



## Key References

* OpenAI Whisper Documentation
* HuggingFace Transformers & Sentence Transformers
* KeyBERT & Librosa Documentation
* TextBlob & NLTK Resources

# Thank You