Live Meeting Copilot: Solution Approach

Team Lusine
Snigdha Paul, Sambit Mallick
Heritage Institute of Technology, Kolkata

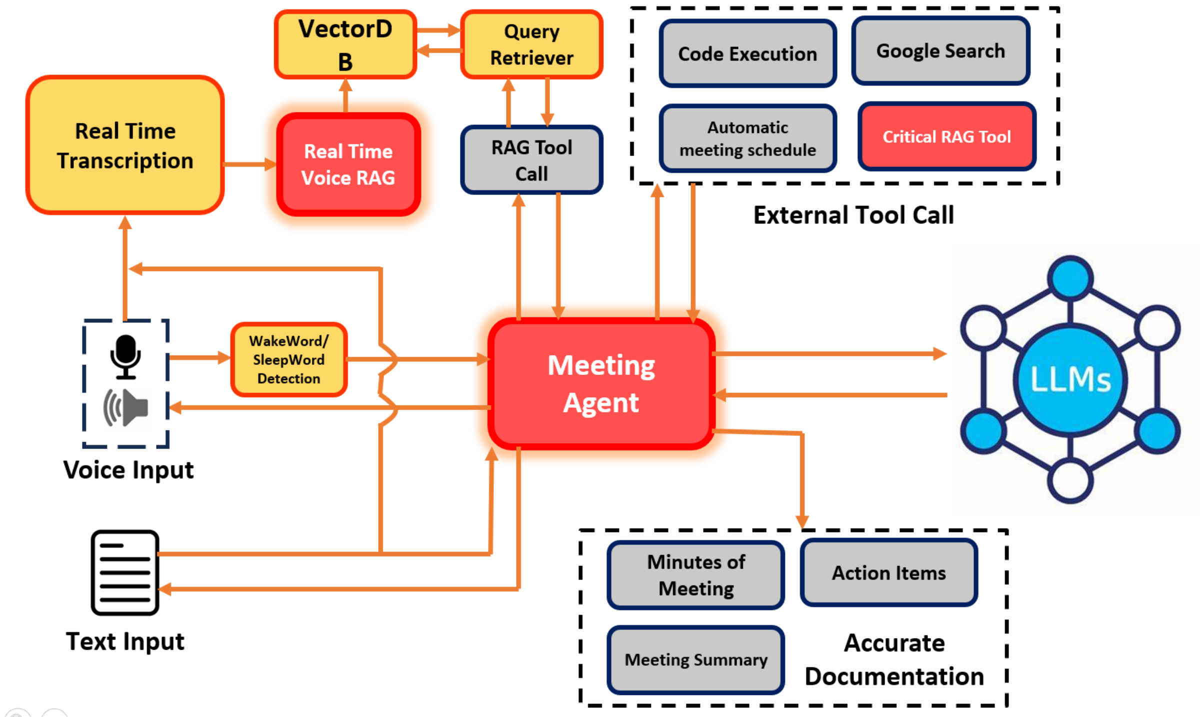
1. Introduction

Following a deep analysis of the cognitive, organizational, and operational problems associated with modern meetings, Team Lusine designed a comprehensive solution: the Live Meeting Copilot. This intelligent system leverages agentic AI principles to transform the way teams interact, document, and follow up during and after meetings.

This report outlines our step-by-step approach to solving the problem, supported by a modular design and integrated technologies that enable real-time intelligence and seamless user interaction.

2. Overview of Solution Architecture

Our system architecture is built around a central agentic AI framework with the following interconnected modules:



- **Input Interfaces**: Real-time voice and text inputs from users.
- **Transcription Engine**: Converts speech to text in real time.
- **RAG Engine**: Retrieves contextual knowledge during conversation using vector-based semantic search.

- **Command Handler**: Parses inputs and determines tool calls (calendar, code exec, search, etc.).
- **Action Generator**: Summarizes meeting content, extracts decisions, creates tasks.
- **Frontend Display**: PyQt5 GUI showing live updates, speaker highlights, and content visualization.

---

3. Key Features and Functional Modules

3.1 Real-Time Transcription

- Integrated **Speechmatics API** provides accurate transcription.

3.2 Voice-Based Interaction with RAG

- Supports wakeword/sleepword for intuitive control.
- Users can ask questions mid-meeting (e.g., "What are today's key decisions?")
- Context-aware responses powered by **LLAMAIndex** and **Gemini Text Embeddings**.

3.3 Semantic Knowledge Retrieval

- Past meeting content stored in **VectorDB**.
- Enables semantic search over previous decisions, summaries, and action items.

3.4 Agentic Task Execution

- External tools triggered through commands:
  - **Calendar Integration**: Schedules follow-ups
  - **Google Search**: Retrieves real-time external information
  - **Code Execution Module**: For live coding tasks or technical explanation

3.5 Action Item & Summary Generation

- Uses real-time conversation context to automatically:
  - Extract decisions
  - Assign responsibilities
  - Generate MoMs and summaries post-meeting

3.6 Visual Explanation Layer

- Diagrams and code snippets are generated based on spoken content
- Allows on-screen augmentation for better understanding during technical meetings

---

4. Technical Stack

Component	Technology Used
Programming Language	Python
Frontend GUI	PyQt5
Voice Processing	Gemini AI Studio, GROQ API, WebSockets
Wakeword Detection	Custom Neural Network
Transcription	Speechmatics API
RAG System	LLAMAIndex + Gemini Embeddings + Sentence Transformer
Calendar Integration	Google Calendar API with OAuth 2.0
Visual Generation	Gemini Vision API + Custom Generative Tools

5. Outcome & Impact

By implementing this modular and intelligent system, Team Lusine was able to:

- **Reduce cognitive load** during meetings through on-screen summaries
- **Capture 100% of critical decisions** and responsibilities
- **Increase follow-up task execution rate** by enabling smart reminders and schedules
- **Improve participant satisfaction** by allowing real-time clarification and visual explanations

This transformation from passive meeting tools to an active, intelligent copilot marks a significant leap in how we approach collaboration.

6. Conclusion

The Live Meeting Copilot effectively bridges the gap between chaotic meeting experiences and organized, intelligent team collaboration. By embedding agentic AI principles and designing an adaptive real-time assistant, we have turned every meeting into a structured, searchable, and impactful event.

Team Lusine believes this innovation not only addresses a pressing problem but also redefines the very fabric of digital collaboration.