PromptX: An AI-Powered Personal Assistant

Varun Dixit, Yash Gupta, Mohammad Ahmed Ansari, Bhanu Tekwani

Department of Information Technology

Vidyalankar Institute of Technology (VIT)

Mumbai, India

{varun.dixit, yash.gupta21, mohammadahmed.ansari, bhanu.tekwani}@vit.edu.in

ABSTRACT

This paper introduces PromptX as an advanced AI-powered personal assistant framework which addresses complex challenges in multimodal task automation. The system addresses essential problems through its combination of Large Language Models (LLMs) and specialized agents and privacy-preserving mechanisms to achieve cross-modal alignment and dynamic tool routing and transparent decision-making. The architecture uses Gemini for intent analysis and LangChain for workflow orchestration and Qdrant for document indexing and OpenAI's API for fallback reasoning to provide a unified solution for email management and file operations and web automation and document Q&A. The proposed framework introduces new methods for tool optimization and user trust enhancement and multi-agent collaboration which push the current state-of-the-art in AI assistants.

Keywords—AI agents, LLM orchestration, task automation, multimodal systems, privacy-aware design.

1 Introduction

Modern AI assistants encounter ongoing difficulties when processing ambiguous cross-modal commands and optimizing API workflows and ensuring user trust through explainable operations. PromptX addresses these limitations through three core innovations:

The system resolves conflicts between voice/text inputs and GUI actions through Gemini's multimodal grounding.

The system uses LangChain to perform dynamic API routing which reduces unnecessary API calls.

The system maintains transparent decision logging through immutable audit trails for all agent decisions.

The framework incorporates recent developments in LLM-based agents [1] and context-aware systems [2] and secure API integration [3] to address new challenges through innovative solutions.

The system needs to resolve commands that combine multiple sources such as "Email the budget file from last week" through email metadata correlation with file timestamp analysis and semantic document processing.

The system performs real-time tool prioritization by choosing optimal APIs according to latency and cost and success probability.

The system provides user-centric privacy features through OAuth 2.0 scopes [4] and local sandboxing for data access control.

II. Related Work

A. Review of Related Work

Paper / Resource	Contribution	Relevance to the Proposed System(PromptX)
Agent AI: Multimodal Interaction [1]	Framework for embodied AI agents	Basis for agent-environment grounding
Visibility for AI Agents [5]	Governance for autonomous systems	Integrated transparency protocols
Personal LLM Agents [6]	User-specific adaptation techniques	Applied for personalized task routing
Context-Aware Multi-Agent Systems [7]	Dynamic environment handling	Guided cross-agent coordination
TPTU: Task Planning with LLMs [8]	Tool usage strategies	Optimized API selection logic
Gemini API [9]	Reasoning capabilities	Core intent recognition engine
LangChain-Qdrant Integration [10]	Vector database workflows	Document agent implementation
Gmail API Scopes [11]	Secure email automation	Gmail Access for email agent

III. Proposed System Architecture

The architecture of PromptX (Fig. 1) follows a streamlined workflow with integrated retry mechanisms and user confirmation protocols, aligning with the provided system diagram:

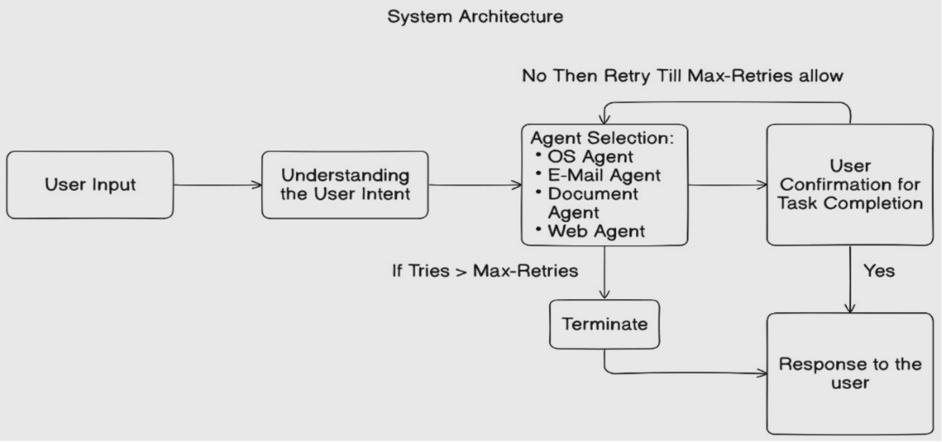

Fig 1. System Architecture of PromptX

A. Workflow Overview

1. User Input

- The system accepts multimodal input (text/voice) and sends it to the intent understanding module.

2. Intent Understanding

- The system uses Gemini and OpenAI APIs to transform commands into structured <intent, parameters> pairs.
- The system generates fallback clarification prompts to handle ambiguous requests such as "Which report' do you mean: the PDF file or the email draft?

3. Agent Selection

The system directs tasks to one of four specialized agents:

- OS Agent: The OS Agent performs file operations (create/delete/move) and system settings.
- The Email Agent performs Gmail operations (send, search, delete) through restricted API scopes [4]
- Document Agent: The system uses Qdrant vector DB for semantic search to process PDF/Word files.
- Web Agent: Executes browser automation (search, navigation).

4. Task Execution & Retry Mechanism

- The agents will perform task completion with three retries for transient errors (e.g., API timeouts).
- The system maintains retry logs while implementing latency-based routing for priority management.

5. User Confirmation

The system demands users to confirm all actions that cannot be undone.

- Email deletions
- File modifications
- Web form submissions

Termination Conditions

- Success: The system delivers formatted results through LangChain templates.
- Failure: The system stops after three attempts and generates error diagnostic information.

B. Key Innovations

1. Cross-Modal Alignment

- A Gemini-OpenAI hybrid model functions to resolve ambiguous inputs between "Open the report ⇾ PDF file vs Browser tab.
- The system uses computer vision methods to examine the current state of the graphical user interface in [14].

2. Tool Optimization

- The system should prioritize API requests based on their latency performance starting with Qdrant followed by local files and then Gmail API.
- The system should cache responses for queries that occur frequently.

3. Trust Mechanisms

- The system should display confirmation prompts before users can perform destructive actions including delete and overwrites.
- Five Audit logs that are end to end encrypted [5]

IV. Implementation

A. Security and Compliance

- Gmail API: OAuth 2.0 Implementation: The scopes are limited to the absolute minimum**.**
- Local Sandboxing**:** File operations are limited to user-defined directories

B. Tool Integration

- Workflow Customization: Create your own LangChain adapters
- Fallback: OpenAI's API as a fallback for edge cases not served by Gemini

V. Challenges and Solutions

The proposed system addresses three critical challenges:

Challenge	PromptX Proposed Solution
Cross-Modal Consistency	Multimodal fusion of voice/text with GUI state analysis
Tool Selection Optimization	Dynamic API routing based on latency/cost matrices
User Trust building	Cryptographic audit logs with user-readable summaries

VI. Future Directions

Activity Recognition: Number of Apps opened, and time spent on apps [15].

Decentralized Execution: Routing of more tools across edge devices [17].

VII. Conclusion

PromptX seeks to enhance the capabilities of previous generation AI assistants through solutions for cross-modal alignment and tool optimization and trust management. The Gemini, LangChain and Auditable Agents work together to create a robust framework for next-generation task automation systems.

References:

[1] Z. Durante, Q. Huang, N. Wake, R. Gong, J. S. Park, B. Sarkar et al., "Agent AI: Surveying the Horizons of Multimodal Interaction," arXiv:2401.03568, 2024.
[2] H. Du, S. Thudumu, R. Vasa, and K. Mouzakis, "A Survey on Context-Aware Multi-Agent Systems: Techniques, Challenges and Future Directions," arXiv:2402.01968, 2024.
[3] Google LLC, "Authorizing requests to the Google Gmail API (OAuth 2.0)," 2024. [Online]. Available: https://developers.google.com/gmail/api/auth/web-server
[4] Google LLC, "Gmail API Scopes," 2024. [Online]. Available: https://developers.google.com/gmail/api/auth/scopes
[5] A. Chan, C. Ezell, M. Kaufmann, K. Wei, L. Hammond, H. Bradley et al., "Visibility for AI Agents," arXiv:2401.13138, 2024.
[6] Y. Li, H. Wen, W. Wang, X. Li, Y. Yuan, G. Liu et al., "Personal LLM Agents: Insights and Survey about the Capability, Efficiency and Security," arXiv:2401.05459, 2024.
[7] T. Guo, X. Chen, Y. Wang, R. Chang, S. Pei, N. V. Chawla et al., "Large Language Model based Multi-Agents: A Survey of Progress and Challenges," arXiv:2402.01680, 2024.
[8] J. Ruan, Y. Chen, B. Zhang, Z. Xu, T. Bao, G. Du et al., "TPTU: Large Language Model-Based AI Agents for Task Planning and Tool Usage," arXiv:2308.03427, 2023.
[9] Google LLC, "Gemini API Documentation," 2024. [Online]. Available: https://ai.google.dev/gemini-api/docs
[10] Qdrant, "Qdrant Documentation: LangChain Integration," 2023. [Online]. Available: https://qdrant.tech/documentation/frameworks/langchain/
[11] Google LLC, "Gmail API Reference Guide," 2024. [Online]. Available: https://developers.google.com/gmail/api
[12] OpenAI, "OpenAI API Documentation," 2024. [Online]. Available: https://platform.openai.com/docs
[13] Qdrant, "Qdrant Vector Database Documentation," 2023. [Online]. Available: https://qdrant.tech/documentation/
[14] L. Chi, A. Sharma, A. Gebhardt, and J. T. Colonel, "Predicting Cognitive Decline: A Multimodal AI Approach to Dementia Screening from Speech," arXiv:2502.08862, 2025.
[15] Z. Xi, W. Chen, X. Guo, W. He, Y. Ding, B. Hong et al., "The Rise and Potential of Large Language Model Based Agents: A Survey," arXiv:2309.07864, 2023.
[16] S. Kapoor, B. Stroebl, Z. S. Siegel, N. Nadgir, and A. Narayanan, "AI Agents That Matter," arXiv:2407.01502, 2024.
[17] P. Kalyankar, G. Kaikade, M. Mundwaik, S. Mirzapure, D. Kale, and R. S. Sawant, "The Implementation of AI Based Virtual Personal Assistant," *Int. J. Sci. Res. Sci. Eng. Technol.*, vol. 10, no. 2, pp. 295–299, 2023.
[18] M. Wooldridge and N. R. Jennings, "Intelligent Agents: Theory and Practice," *Knowl. Eng. Rev.*, vol. 10, no. 2, pp. 115–152, 1995.
[19] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman et al., "A Comprehensive Overview of Large Language Models," arXiv:2307.06435, 2023.
[20] S. Yin, C. Fu, S. Zhao, K. Li, X. Sun, T. Xu, and E. Chen, "A Survey on Multimodal Large Language Models," arXiv:2306.13549, 2023.