iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** V **Month of publication:** May 2025

DOI: https://doi.org/10.22214/ijraset.2025.69750

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

PromptX: An AI-Powered Personal Assistant

Varun Dixit[1], Yash Gupta[2], Mohammad Ahmed Ansari[3], Bhanu Tekwani[4]
[1]*Department of Information Technology, Vidyalankar Institute of Technology (VIT), Mumbai, Maharashtra, India*
[2]*Professor, Department of Information Technology, Vidyalankar Institute of Technology (VIT), Mumbai, Maharashtra, India*

***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.*

I. INTRODUCTION

The search for artificial intelligence (AI) systems that exist in harmony The desire to assist people in their day-to-day lives has largely driven cannot invest in research and development, and hence the progress of Intelligent Personal Assistants (IPAs) [6]. Since their invention, explorations of speech recog- nition to advanced assistants embedded in smartphones and smart home devices, IPAs aim to enhance user efficiency and usability [6, 17]. These systems are a long-standing ideal in AI: creating things that perceive their environment, make decisions, and take actions [18]. However, despite progress demonstrated by commercial options, classic IPAs tend to struggle with restrictions on actually comprehending sophisticated user intent, to conform to various contexts [2], creating intricate patterns of activity, and properly improving their capabilities without requiring systematic manual coding or template building [6].

Their operations often depend on established guidelines or specific domains- specific training, which prevents their flexibility and their capacity to act as truly independent and responsive contributors to online presence [6]. The advent of Large Language Models (LLMs) such as Google's Gemini [9] signify a a turning point, introducing unparalleled functionalities in natural language comprehen- sion, production, and advanced thought derived from large data sets [19]. These innovations have gave rise to LLM-based autonomous agents – AI systems where the LLM is the core cognitive unit, capable of perceiving environments, planning sequences of activities, alongside the employment of external resources to attain objectives [15, 7, 18]. This agent paradigm, discussed in recent surveys [15, 7,]] 6], has promising potential for creation of the coming generation of IPAs. This article presents PromptX, a personal A basic framework constructed from the Gemini API [9], i.e., intended to address the specific needs of *individual* support, a field studied thoroughly by Li et al. [6]. PromptX stands out with extensive integration with a user's personal virtual environment—files, mails, device settings, and personal data—trying to provide individualized, context-dependent support [6]. It is modular, multi-agent architecture where certain agents (means), i.e., an Email Agent or File Agent, perform certain actions under the orchestration of the original Gemini method, that takes into consideration user intention and the management of planned activities [8]. The plan intends to control process complicated, potentially multimodal directives (utilizing Gemini's strength and vision of multimodal agent studies [1, 14, 20]), enhance organizational procedures through adaptive tool choice [8, 16], and perform safely and transparently, utilizing robust frameworks like OAuth 2.0 [3, 4] and principles based on research in the observability of AI agents and governance [5]. This paper presents the PromptX framework, elucidating its architecture, its strengths derived from its Gemini central and specialized agents, along with its methodology regarding the essential challenges of developing effective, efficient, and reliable AI- intelligent personal assistants [6, 16, 17]

II. LITERATURE REVIEW

The PromptX development is informed by breakthroughs in ments across several key research domains. Understanding This context is required in order to justify the design decisions and issues with creating an LLM-based individual.

©IJRASET: All Rights are Reserved

A. Author Contributions

The PromptX project discussed above is the output of intensive development efforts by the core team. Varun Dixit led the unification of the LLM core functionalities, with emphasis on Gemini API necessitates planning and intent recognition and became pivotal in specifying the overall flow of the system. Yash Gupta focused on the creation and consolidation of the single-threaded agent modules, particularly the Email Agent such as OAuth 2.0 security features, and the OS Agent with its sandboxing needs; he also made signific- icantly to adapting LangChain for agent orchestration. Mo- hammad Ahmed Ansari underscored the persistence of data and re- trieval facets, designing the Document Agent with Qdrant for storage and retrieval of vectors, and to the The development and deployment of the web agent, and the secure. Audit logging mechanism. Prof. Bhanu Tekwani delivered useful guidance and oversight throughout the project, providing ex- pert knowledge of AI agent theory, task automation strategies, system architecture optimization, and maintaining the research methodology consistent with best practices today. All members contributed actively while undergoing the review of applicable literature and preparation of this manuscript.

B. Review of Related Work

PromptX draws on the wide foundation established by previous studies. Foun- dational intelligent agent design princi- ples, such as autonomy and reactivity, borrowed from founda- tional texts such as Wooldridge and Jennings [18] recognize the underlying facilitating technology, LLMs, is recognised through exhaustive questionnaires outlining The architectures, training protocols, and capabilities demonstrated by Naveed et al. [19] utilize Google's Gemini overtly. API [9] enables PromptX to employ sophisticated reasoning capa- abilities. The agent paradigm, particularly with LLMs, is a a rapidly emerging field. Xi et al.'s [15] and Guo et al. [7] give a relevant background description regarding LLM-based agent architectural designs, including multi-agent approaches relevant to PromptX's modular design, and outline the current progress and challenges. Our focus on a *personal* assistant is in keeping immediately in relation to the specific factors mentioned in the survey by Li et al. [6], which addresses the specialist requirements of user-specific adaptation, on- device efficiency, and security for agents deeply integrated with personal data. Managing varied inputs and changing environments is criti- The research conducted by Durante et al. [1] on multimodal interaction in Agent AI gives context to PromptX's vision for cross- modal alignment, while the study by Chi et al. [14] offers a specific case of multimodal processing of artificial intelligence, albeit in a to another domain. Also, the ability of agents to adapt to changing situations is explored in the survey on Context- Human- aware Multi-Agent systems by Du et al. [2]. Effective task completion often entails working with external devices. The study by Ruan et al. [8] has a specialized framework for LLM agents for task planning and tool utilization, guiding PromptX's orchestration and API selection logic. Practical application benefits documented APIs such as Gemini [9], OpenAI [12], and Gmail [11], as well as LangChain [10] and vector databases such as Qdrant [13]. Security aspects such as OAuth 2.0 [3] and defining proper access scopes [4] are essential for safe external communications. Less complex VPA deployments [17] offer real-world benchmarks. Assuring the reliability and evaluating the practical use The significance of such agents as PromptX is paramount; the evaluation of existing agent benchmarks and the necessity of cost-sensitive measurement by Kapoor et al. [16] stress the significance of PromptX's focus on improvement and effectiveness. Similarly, the call in terms of transparency and governance, as Chan et al. discussed. [5], encourages PromptX's offer of audit logs and user confirmations. Finally, LLM surveys [19] and multimodal LLMs [20] offer fundamental knowledge of the central technology.

III. PROPOSED SYSTEM ARCHITECTURE

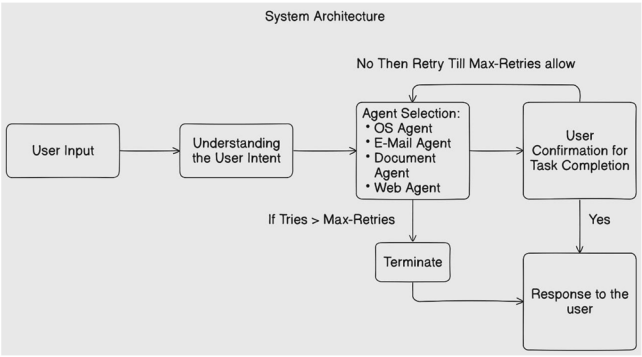


Fig. 1. System Architecture of PromptX (Conceptual Diagram - Replace with actual figure).

The architecture of PromptX (Figure 1) follows a stream- lined workflow with in-built retry features and user confirma- tion processes. It is designed to process user requests, know the target, select appropriate tools (agents), do perform actions securely, and provide results effectively.

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 [9] and OpenAI [12] 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 [13] vector DB for semantic search to process PDF/Word files.
- Web Agent: Executes browser automation (search, navigation).

4) Task Execution & Retry Mechanism
- Agents perform tasks with up to three retries for transient errors (e.g., API timeouts).
- The system maintains retry logs and implements latency-based routing for priority management.

5) User Confirmation
- Mandatory confirmations for irreversible actions: email deletions, file modifications, web form sub- missions.

6) Termination Conditions
- Success: Delivers formatted results through LangChain templates.
- Failure: Stops after three attempts and provides diagnostic information.

B. *Key Innovations*

PromptX introduces several innovations to enhance capabil- ity, efficiency, and trustworthiness:

1) Cross-Modal Alignment and Grounding: Leverages Gem- ini’s multimodal capabilities to ground user references (e.g., “this document”) across text, voice, and GUI con- text [1, 14].

2) Dynamic Tool/API Selection: Prioritizes agents by la- tency and cost, choosing the most efficient execution path (e.g., Qdrant over Gmail when suitable) [8, 16].

3) Transparent and Trustworthy Operation: Combines mandatory confirmations with immutable audit logs (cryptographically hashed) for full visibility [5, 6].

4) Unified Agent Orchestration: Uses Gemini as a central orchestrator for specialized agents (Email, Files, Web, Documents) to handle complex, multi-domain tasks [7, 15].

5) Privacy-Centric Design: Employs OAuth 2.0 with min- imal scopes [3, 4] and sandboxed local operations to protect user data.

6) Optimized API Logic: Integrates performance metrics into LLM-driven tool selection for cost- and time-efficient task planning [8].

7) Context-Aware Foundation: Supports future proactive be- haviors using richer user and system state context [2, 6].

IV. IMPLEMENTATION

The PromptX infrastructure is largely run in Python, that coordinates interaction between the user interface, Google Gemini API [9] (basic LLM core), specialized agent modules as utilities, and many external services and local resources. Gemini is the central player in the workflow, tasked with Comprehending user instructions and task breakdown. plan- ning and action [8], selecting appropriate tools in line with descriptions, constructing arguments, and combining the final response. The OpenAI API [12] serves as a configurable fallback. The orchestration layer leverages the LangChain framework- work [10] for describing agents as ”tools” and regulating the execution loop, with retries and error handling. Custom adapters deal with single OS interactions.

Figure 2 shows the Email Agent reading and dis- reading of unread mail via the Gmail API upon receiving the The instruction ”list the unread emails” showcases the fundamental email reading ability enabled through secure API access

Figure 3 shows The agent is depicted to combine the emails mentioned above in response to a user query. This necessitates Gemini [9] to process the email data obtained and produce a shorter summary, demonstrating the interaction between retrieval and generation capabilities.

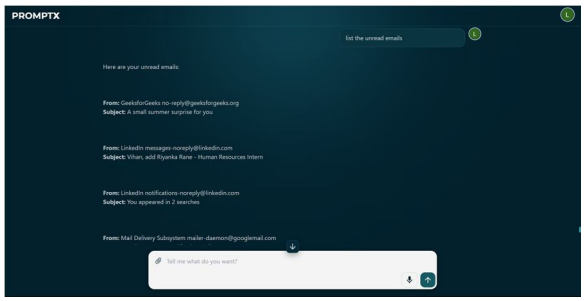


Fig. 2. PromptX interface displaying listed unread emails.

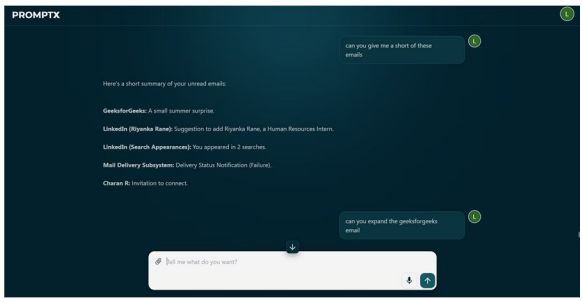


Fig. 3. PromptX providing a short summary of emails.

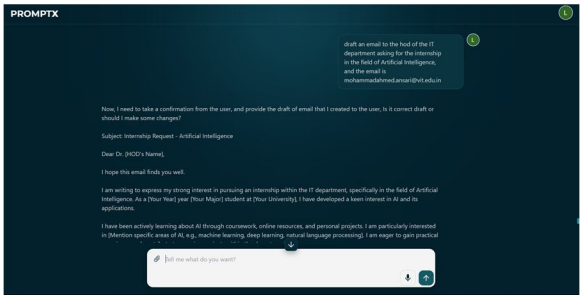


Fig. 4. Drafting an email and requesting user confirmation.

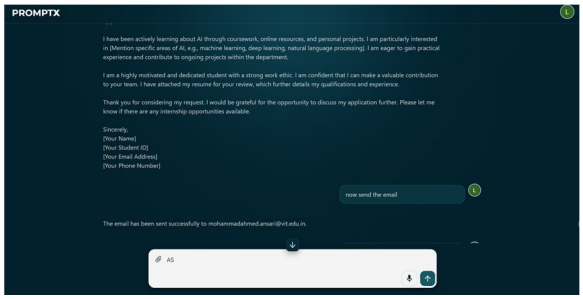


Fig. 5. Confirmation of successful email dispatch.

Figure 4 illustrates the Email Agent generating an email from user input (recipient, subject, general text) and presenting it to the user for approval prior to sending. This is a critical trust and safety feature [5, 6]. When it receives the instruction ”now send the email” (user

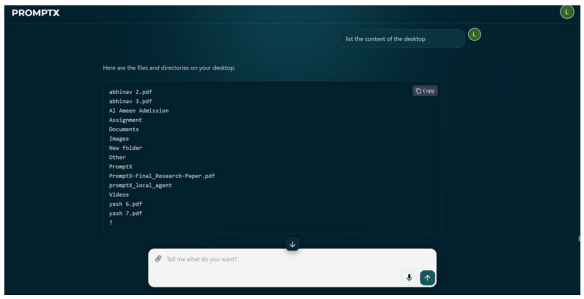


Fig. 6. OS Agent listing files and directories on the desktop.

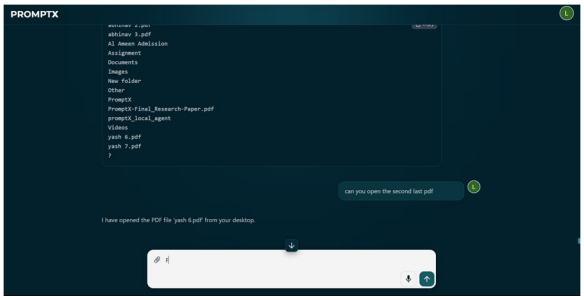


Fig. 7. OS Agent opening a specified PDF file from the desktop list.

confirmation), the agent uses the Gmail API via secure OAuth2.0 [3] to send the message and then provides confirmation to the user, as conceptually shown in Figure 5.

The OS Agent interacts with the local file system within its defined sandbox. Figure 6 shows the agent listing the contents of the user's desktop directory in response to the command "list the content of the desktop".

Following the listing, Figure 7 demonstrates the OS Agent performing a file operation with relative reference ("can you open the second last pdf"), opening the named file ('yash 6.pdf'). This indicates intent recognition and execution on local files. The same flows apply to creating and removing files, with confirmation always requested for removal.

The Document Agent utilizes Qdrant [13] for RAG opera- tions, orchestrated via LangChain [10]. The Web Agent uses standard Python libraries. Security confirmations and audit logs [5] are integrated into the workflow.

V. CHALLENGES AND SOLUTIONS

Construction and deployment of PromptX require overcom- ing major challenges inherent in existing LLM-based agent technology [15, 7, 16, 6, 5], especially for personal assistance. One of the major challenges is Cross-Modal Consistency in in- terpreting user intent that may implicitly or explicitly span text, voice, and visual (GUI) information [1, 14]. Misinterpreting references can result in significant errors. PromptX avoids this by taking advantage of Gemini's [9] multimodal capabilities, enabling it to process linguistic commands in conjunction with contextual knowledge about the system state (e.g., the current application window or selected file), grounding the command more precisely. Another challenge is Tool Selection Optimization for cost and performance [8, 16]. Ineffective naive tool selection is possible. PromptX solves this with Dynamic API routing, employing performance metrics to make smart decisions between capable tools (e.g., using local Qdrant search [13] instead of a slower Gmail API call [11]). The third major challenge is building User Trust due to the agent's access to personal information [5, 6]. PromptX solves this with Mandatory User Confirmations for important actions and by employing Cryptographic Audit Logs based on visi- bility principles [5], offering transparency and accountability about the agent's decision-making process and actions. These solutions offer a solid foundation, though continued research into LLM reliability and security is still required [19, 5].

VI. FUTURE DIRECTIONS

1) Activity Recognition: Enhance context by tracking met- rics like the Number of Apps opened and time spent on apps, similar to agent behavior studies [15].

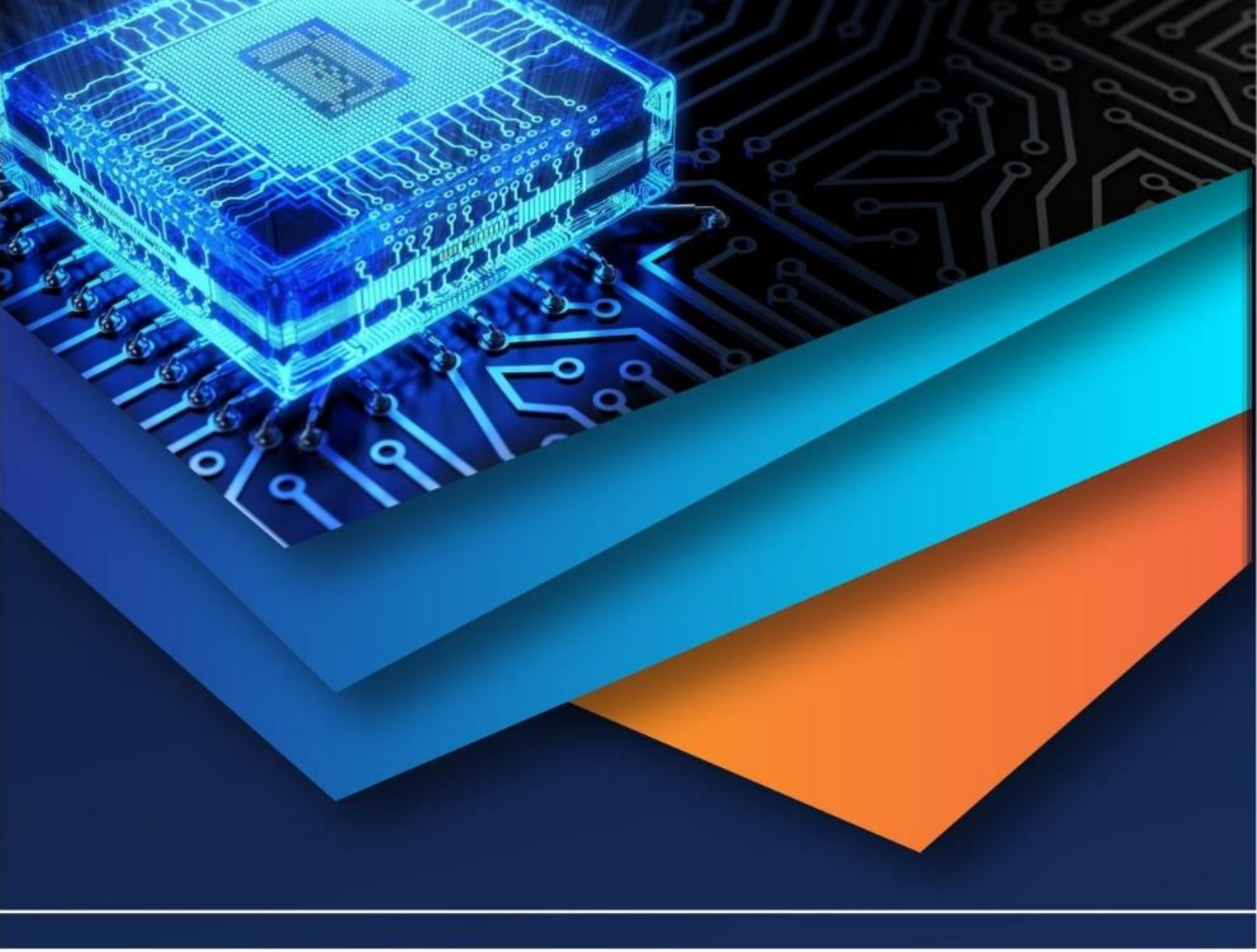
2) Decentralized Execution: Explore routing of more tools across edge devices for improved privacy and perfor- mance, building on concepts from [17].

VII. CONCLUSION

PromptX provides a comprehensive and visionary architec- ture for an AI-driven personal assistant, successfully com- bining the advanced reasoning and multimodal understanding capabilities of Google's Gemini API [9] with a structured, modular agent architecture. By systematically addressing fun- damental challenges identified in recent literature—such as cross-modal alignment [1, 20, 14], optimal use of instruments driven by performance metrics [8, 16], and building user trust through control and transparency [5, 6]—PromptX is an advance over conventional intelligent personal assistants (IPAs) [17] and contemporary large language model (LLM) agent architectures [15, 7]. The addition of expert agents for handling various personal digital tasks from emails [3, 4, 11] to documents (with Qdrant [10, 13]), controlled smartly by platforms like LangChain [10], allows for a more dynamic and potent user experience. The emphasis on security through OAuth 2.0 [3] and privacy through sandboxing and bounded scope requests [4] provides a good foundation for personal deployment. While admitting the current problems regarding LLM reliability, cost-effectiveness, and the security context inherent to personal agents [16, 19, 5], PromptX provides a valuable contribution by providing a concrete architecture and detailing solutions based on recent research outcomes [e.g., 1, 2, 8, 18]. It acts as a needed guide for future development, trying to leverage the potential of personalized, efficient, and reliable AI assistants that are intrinsically embedded with users' daily activities.

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 preprint 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 Direc- tions," arXiv preprint 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 preprint arXiv:2401.13138, 2024. (Accepted to ACM FAccT 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 preprint 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 preprint 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 preprint arXiv:2308.03427, 2023. (NeurIPS-2023 Workshop).
[9] Google LLC, "Gemini API Documentation," 2024. [Online]. Available: https://ai.google.dev/gemini-api/docs
[10] Qdrant, "Qdrant Documentation: LangChain Integration," 2023. [On- line]. 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 Cog- nitive Decline: A Multimodal AI Approach to Dementia Screening from Speech," arXiv preprint 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 preprint arXiv:2309.07864, 2023.
[16] S. Kapoor, B. Stroebl, Z. S. Siegel, N. Nadgir, and A. Narayanan, "AI Agents That Matter," arXiv preprint 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. 784–788, 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 preprint 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 preprint arXiv:2306.13549, 2023.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)