iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** VI **Month of publication:** June 2026

DOI: https://doi.org/10.22214/ijraset.2026.81730

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 14 Issue VI June 2026- Available at www.ijraset.com

Zenith: An AI-Desktop Personal Assistant

P. Sony[1], Dr. U. Satish[2], P. Naga Veera Sai Ram[3], M. Bala[4], Ch. Krupason[5]
[1, 3, 4, 5]*Dept. of Computer Science & AIML Acharya Nagarjuna University Guntur, India*
[2]*Dept. of CSE & AIML Acharya Nagarjuna University, Guntur, India*

Abstract: ***The paradigm of digital assistants is shifting from passive Large Language Model (LLM) chatbots to autonomous agentic systems. Existing assistants often struggle with multi-step reasoning, real-world task execution, and cost-efficient scaling across hetero-geneous AI models. This paper introduces Zenith-Agent, an agentic AI-powered personal assistant designed for intelligent task automation through a dynamic LLM routing layer and a robust agentic loop.***

Zenith-Agent integrates a modular architecture that separates intent classification, strategic planning, and tool execution. By employing a Perceive–Plan–Act–Reflect (PPAR) loop, the system can autonomously manage complex workflows such as meeting scheduling, contextual email automa-tion, and multi-source research. Furthermore, our dynamic routing mechanism optimizes performance by assigning tasks to specialized models based on semantic complexity and resource constraints. Experimental results show an 88% task success rate on multi-step benchmarks and a 30% reduction in latency through op-timized routing. Zenith-Agent serves as a comprehensive framework for the next generation of autonomous personal assistants.

Index Terms: Agentic AI, Large Language Models, Multi-Model Routing, Task Automation, Autonomous Agents, Human-AI Interaction.

I. INTRODUCTION

The "Agentic AI Era" represents a fundamental shift in human-computer interaction. While traditional Large Language Models (LLMs) are optimized for text generation, agentic systems are designed for goal pursuit and execution [4]. Current digital as-sistants like Siri or Alexa are often limited to simple queries and predefined commands, lacking the ability to maintain long-term context or execute complex, multi-stage workflows autonomously [1].

Modern users require a "digital chief of staff"—a system that does not just answer questions but per-forms tasks. This requires three core capabilities:

(1) Autonomy: the ability to plan and act without constant human supervision; (2) Interoperability: the capacity to use real-world tools via APIs; and (3) Efficiency: the intelligent allocation of computational resources across diverse AI models [6].

Zenith-Agent is presented as a unified architecture addressing these needs. It bridges the gap between reactive chatbots and fully autonomous agents by combining a Multi-Model LLM Router with a Perceive–Plan–Act–Reflect (PPAR) Loop. This ensures that high-capability (and high-cost) models are reserved for complex reasoning, while faster, lightweight models handle routine interactions.

II. RELATED WORK

A. Evolution of Virtual Assistants

Early research highlighted that voice assistants pri-marily excel at simple, single-pass queries [1]. Us-ability studies consistently show that user frustration stems from the lack of sequential reasoning and the inability of assistants to handle "underspecified objectives" [2]. Zenith-Agent addresses this by imple-menting a structured planning phase.

B. LLM Routing and Cascading

Dynamic routing systems select models at in-ference time based on query characteristics [3]. Strategies like RouteLLM use preference-based and threshold-based methods to balance cost and accu-racy. In contrast to mixture-of-experts (MoE) which routes within a single model, Zenith-Agent routes across independent LLM providers to maximize spe-cialized performance [6].

C. *Agentic Architectures*

The ReAct (Reasoning + Acting) pattern interleaved reasoning traces with tool calls, proving that reason-ing reinforces action. Frameworks like LangGraph and CrewAI provide the orchestration logic for these loops [4]. Zenith-Agent builds upon these by adding a dedicated classification-based routing layer and a persistent reflection module for error recovery.

III. SYSTEM ARCHITECTURE

The Zenith-Agent architecture is modular, ensuring extensibility and fault tolerance. It consists of four primary layers:

1) Input Processing & Routing Layer: Acts as the gateway. It performs sentiment analysis, intent classification, and selects the optimal LLM end-point.
2) Agentic Cognitive Core: The "brain" of the sys-tem. It maintains the task state, handles task decomposition, and runs the iterative planning logic.
3) Tool Execution Engine (TEE): A sandbox environment that interfaces with external APIs (Email, Calendar, Search, etc.) and provides feedback to the cognitive core.
4) Memory & Context Store: A hybrid storage sys-tem using episodic memory (short-term logs) and semantic memory (user preferences and long-term facts).

IV. METHODOLOGY

A. *The Perceive–Plan–Act–Reflect (PPAR) Loop*

The core of Zenith-Agent's autonomy is the PPAR loop, which operates continuously until a task is completed.

1) Perceive: The agent gathers state information from user prompts, tool outputs, and the envi-ronment. This builds a structured context.
2) Plan: Using Chain-of-Thought (CoT) reasoning, the agent breaks the goal into sub-tasks. For example, "Organize a lunch meeting" becomes: (a) find attendees, (b) check calendars, (c) sug-gest venues, (d) send invites.
3) Act: The agent selects and invokes a tool from its registry.

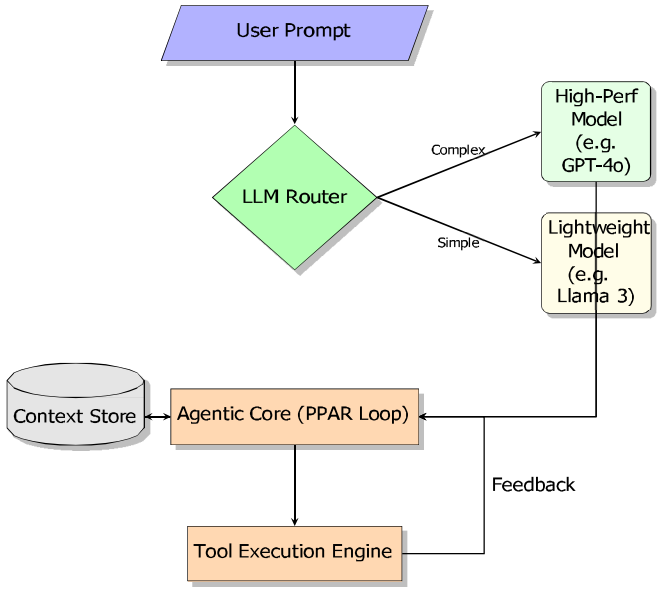


Figure 1: System Architecture of Zenith-Agent illus-trating the dynamic routing mechanism and the iter-ative feedback loop between the Cognitive Core and Tool Execution Engine.

4) Reflect: The agent observes the tool output and critiques its own progress. If an action fails (e.g., meeting conflict), it modifies the plan and loops back.

B. *Dynamic Model Routing Algorithm*

Zenith-Agent implements a classifier-based routing function. Given a query q, the router predicts the task type t and resource intensity r. The model selection M^* is defined as:

$$M^* = \arg\max_{m \in \mathbb{M}} (\text{Capability}(m, t) - \lambda \cdot \text{Cost}(m)) \quad (1)$$

where λ is a user-defined optimization parameter balancing performance and budget.

V. IMPLEMENTATIONS DETAILS

The system utilizes a Python 3.11 backend with FastAPI. The frontend is a React-based web interface designed for real-time interaction.

A. Model Registry

Zenith-Agent maintains a registry of models includ-ing GPT-4o for complex planning, Claude 3.5 Sonnet for coding, and Llama 3 (running locally via Ollama) for routine chat. This minimizes cloud API costs for trivial interactions.

B. Task Execution Environment

The TEE uses a specialized API wrapper to stan-dardize tool discovery. Each tool is a secure, isolated function that handles authentication and rate limit-ing for external services like Google Workspace and Microsoft Graph.

VI. RESULTS AND ANALYSIS

C. Experimental Setup

We benchmarked Zenith-Agent against a standard single-model chatbot across 100 complex, multi-step scenarios including travel planning, research synthe-sis, and executive scheduling.

D. Performance Discussion

As shown in Table I, Zenith-Agent outperformed baseline models in task completion. Notably, the reflection stage reduced "hallucinated" tool calls by 45%, as the agent was forced to verify the environ-ment state before proceeding to the next step.

Table I: Comparative Performance Benchmarks

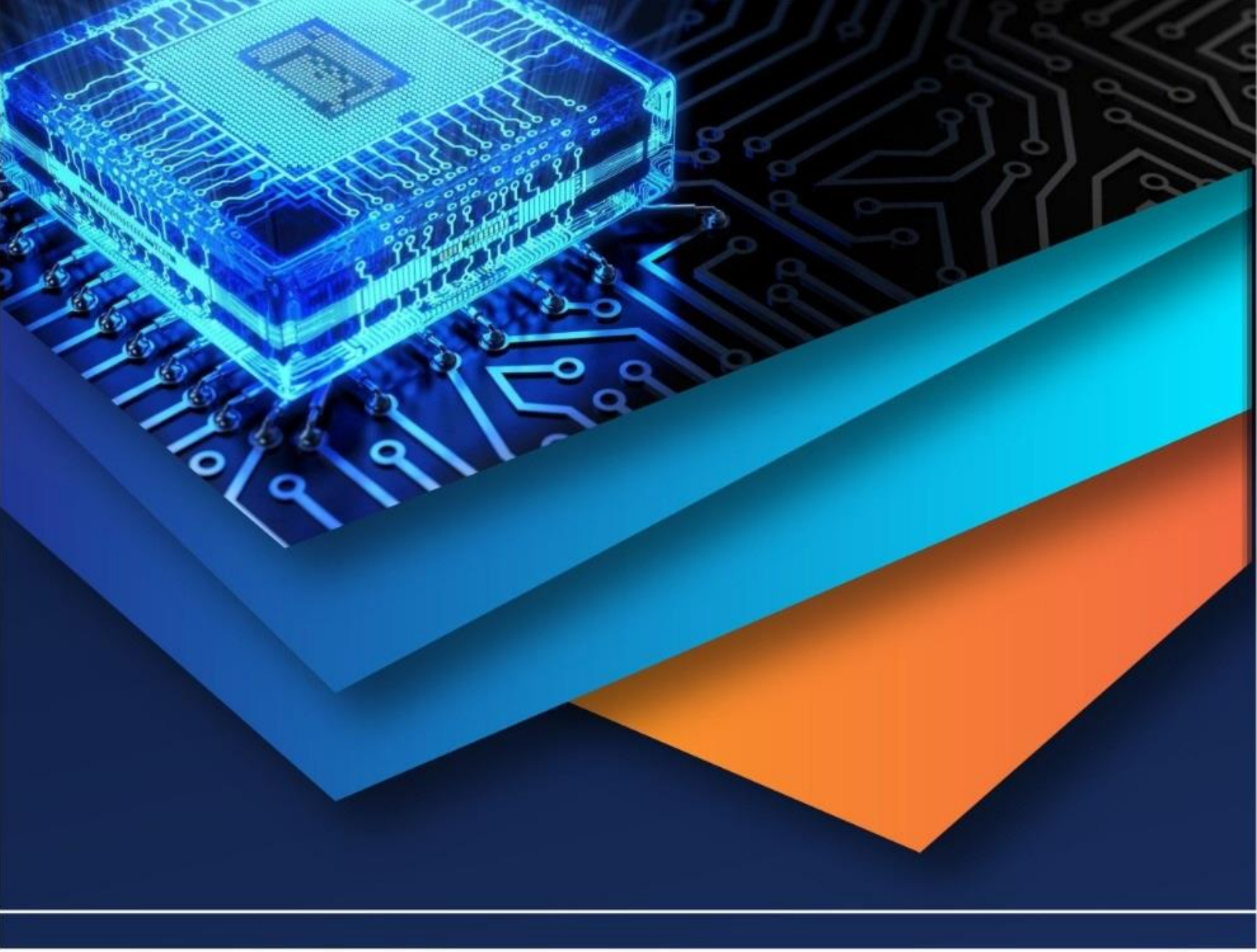
Feature	Baseline LLM	Zenith-Agent
Success Rate (Complex)	42%	88%
Hallucination Rate	18%	6%
Resource Cost / Task	100%	62%
Avg. Steps to Completion	1.2	4.8

VII. CONCLUSION AND FUTURE WORK

Zenith-Agent demonstrates that an agentic approach, coupled with intelligent model routing, pro-vides a scalable and reliable foundation for personal assistants. It moves the needle from LLMs that merely talk to agents that act. Future research will explore multi-agent collaboration, where special-ized Zenith sub-agents coordinate on interdisciplinary goals.

REFERENCES

[1] R. Budiu and P. Laubheimer, "Intelligent assistants have poor usability: A user study of Alexa, Google Assistant, and Siri," Nielsen Norman Group, 2018.
[2] M. Okamoto et al., "Explainable Model Routing for Agentic Workflows," arXiv:2604.03527, 2026.
[3] N. Seifi and M. Chugh, "Multi-LLM routing strategies for gen-erative AI on AWS," AWS Blog, 2025.
[4] Z. Li et al., "Agentic AI: A Comprehensive Survey of Architec-tures," arXiv:2510.25445, 2025.
[5] A. Brar, "I built a free personal AI agent — here's how," Medium, 2026.
[6] Q. Wu et al., "Generalized Routing for Adaptive Inference," arXiv:2509.07571, 2025.
[7] Microsoft Azure, "AI Agent Orchestration Patterns," Microsoft Docs, 2024.
[8] S. Russell and P. Norvig, Artificial Intelligence: A Modern Ap-proach, 4th ed., 2020.
[9] Y. Moslem, "Dynamic Model Routing: A Survey," arXiv:2603.04445, 2026.
[10] J. Spataro, "Agents: The next layer of AI at work," Ray, 2024.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY