A RAG-based Enterprise Knowledge Management System for Urban Sanitation Planning

Francesco Cosimo Mazzitelli
Dept. of Engineering
University of Sannio
Benevento, Italy
f.mazzitelli@studenti.unisannio.it

Eugenio Zimeo
Dept. of Engineering
University of Sannio
CINI Smart Cities & Communities Lab
Benevento, Italy
zimeo@unisannio.it

***Abstract*—Urban sanitation services are critical to smart city ecosystems, as they ensure the liveability of urban environments and require timely access to complex and distributed organizational knowledge. Retrieval Augmented Generation (RAG) enhances document analysis by combining relevant text chunks with user prompts for Large Language Models (LLMs). This paper introduces a blackboard-based modular architecture for enterprise knowledge management in urban sanitation planning. Unlike traditional Retrieval-Augmented Generation (RAG) systems that rely on static pipelines, our approach enables flexible coordination between specialized agents through a shared blackboard.**

***Index Terms*—Knowledge retrieval, Large Language Model, Retrieval Augmented Generation**

I. Introduction

Smart city services such as urban sanitation, mobility, and infrastructure management require timely access to structured, distributed, and often domain-specific knowledge. LLMs, enhanced through Retrieval-Augmented Generation, have proven effective in supporting such knowledge-intensive tasks. However, most existing RAG implementations follow static pipelines that limit flexibility and adaptability across diverse service requirements. This paper proposes a novel architecture that builds upon modular RAG principles by adopting a blackboard-based coordination model. Instead of statically composing processing pipelines, tasks are now published on a shared blackboard where specialized agents can autonomously evaluate whether and how to contribute based on their capabilities and the current state of the task. This design supports more flexible and asynchronous task resolution, allowing agents to collaborate dynamically in response to complex or evolving queries. For instance, when a user submits a query, an initial attempt to answer may be performed using local knowledge. If unsuccessful, the task is posted on the blackboard, prompting relevant agents to retrieve additional context or restructure the prompt. This shifts the architectural focus from workflow orchestration to collaborative reasoning among modular components. The remainder of this paper is structured as follows: Section II overviews existing RAG architectures and related works. Section III introduces the proposed blackboard-based architecture and explains its components and agent collaboration model. Finally, Section V summarizes our contributions and outlines directions for future research.

II. Related Works

Large Language Models (LLMs) are widely used for knowledge-intensive tasks, but domain-specific fine-tuning remains costly and rigid. Retrieval-Augmented Generation (RAG) addresses this by enriching LLMs with external content, improving accuracy and interpretability [1]. A typical RAG pipeline includes a *Retriever* for selecting top-k relevant chunks from a knowledge base and a *Generator* (the LLM) that produces the final response. RAG systems are generally categorized as **Naive**, **Advanced**, or **Modular** [2]. Naive RAG relies on fixed-size chunking and simple retrieval. Advanced RAG improves relevance via query reformulation and re-ranking. Modular RAG [3] enables dynamic composition of pipelines based on the prompt, with strategies such as *Rewrite-Retrieve-Read*, *Generate-Read*, and *Recite-Read.* While state-of-the-art Modular RAG approaches focus on enhancing traditional question answering pipelines primarily by improving individual components, such as retrieval or prompt construction through query rewriting or reranking operators, they remain constrained to a single task type. By contrast, our architecture introduces a dynamic orchestration mechanism based on a blackboard model, which leverages modularity not only to optimize, but also to diversify functionality. Agents autonomously contribute to the processing pipeline by evaluating the input prompt and determining their relevance to the task. This enables the composition of tailored workflows for a wide range of tasks, such as document autocompletion, summarization, inter-document comparison, and question answering, without requiring predefined execution paths. The result is a flexible and extensible framework capable of supporting heterogeneous knowledge-intensive services in dynamic smart city contexts.

III. Task-Oriented Modular RAG

Figure 1 illustrates the interaction flow of the revised blackboard architecture designed for the integration and coordination of AI agents. The system is visually segmented using color coding to distinguish the nature of each component: blue elements represent data structures and content, including user input, intermediate artifacts, and shared state on the blackboard; green elements correspond to the autonomous agents that perform specific tasks; while orange elements denote control and coordination components.

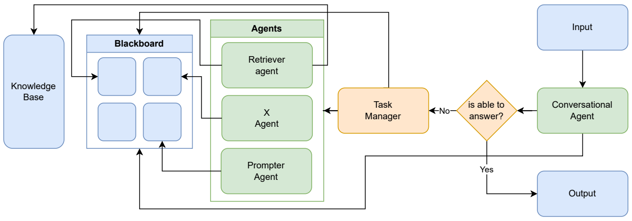

Fig. 1. Blackboard-based modular architecture

At the core of the architecture is the *Blackboard*, a shared data structure that acts as the central communication hub among all components. Each agent monitors the blackboard and autonomously decides whether to act based on the evolving state of a task. This decentralized model supports a more flexible and adaptive form of task resolution, allowing for parallel and asynchronous contributions from multiple agents. In the current implementation, agent coordination and state transitions are managed using **Node-RED**, a flow-based development environment that facilitates rapid prototyping. As shown in Figure 2, Node-RED nodes represent both agents and blackboard events, enabling visual traceability and easy reconfiguration of the collaboration logic.

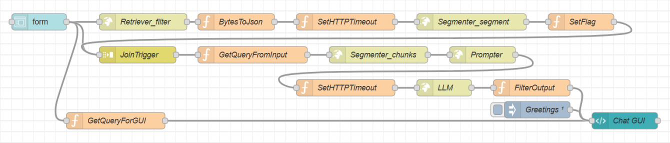

Fig. 2. Node-RED flow implementing blackboard coordination

The *Input* component ingests user queries or external triggers and publishes corresponding tasks on the blackboard. Agents independently evaluate the shared context and autonomously decide whether to contribute based on their capabilities. The resulting prompt is then consolidated and made available to the conversational agent for response generation. This blackboard-based coordination, combined with RESTful microservices, promotes modularity, parallelism, and reusability, while enabling flexible adaptation to diverse knowledge tasks typical of smart city environments. Node-RED remains useful for prototyping and visualizing agent interactions during system development.

IV. Implemented Agents

The implemented agents operate as independent RESTful microservices within the blackboard-based architecture, where each agent monitors the shared blackboard for relevant data or conditions and contributes for the creation of the final result.

The *Retriever* agent performs semantic document retrieval based on vector similarity. Upon detecting a query-related entry on the blackboard, it retrieves all filenames associated with the given category and computes an embedding of the user's query. It then compares this embedding with those of stored documents using a similarity metric, identifying the most relevant file. The result, including metadata and file reference, is written back to the blackboard for use by subsequent agents.

The *Chunker* agent is designed to segment documents in a way that preserves semantic coherence, improving retrieval performance. It detects paragraph boundaries by analyzing font size variations, treating larger fonts as titles and smaller ones as paragraph bodies. To address excessive token lengths, a parent-child structure is used: parent paragraphs are subdivided into shorter child chunks. During retrieval, matches on child chunks allow the reconstruction of semantically complete parent sections, which are then posted to the blackboard.

The *Prompter* agent generates structured prompts tailored to the user's query and task context. Once sufficient context is available on the blackboard, it composes a prompt suitable for tasks such as question answering or document completion. The generated prompt includes a system message instructing the LLM to rely exclusively on the provided context and to return a fallback message if the answer is not derivable from the available information. The prompt is then published on the blackboard for the conversational agent to consume.

V. Conclusion

This work presents a modular, blackboard-based RAG architecture designed to support flexible and scalable deployment across several knowledge-intensive tasks in smart city domains. By leveraging RESTful microservices and a shared blackboard as the coordination substrate, the system enables opportunistic collaboration among autonomous agents that react dynamically to the evolving state of the task. This agent-based design, combined with blackboard-mediated interaction, promotes extensibility, maintainability, and runtime adaptability without requiring predefined workflows. In the current prototype, orchestration logic is prototyped via Node-RED to simulate dynamic behavior; however, this is a transitional solution. Future work will focus on replacing manual orchestration with fully autonomous agent triggering, based on prompt embeddings, content tagging, and blackboard state inference.

References

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.

[2] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang, "Retrieval-augmented generation for large language models: A survey," 2024. [Online]. Available: https://arxiv.org/abs/2312.10997

[3] Y. Gao, Y. Xiong, M. Wang, and H. Wang, "Modular rag: Transforming rag systems into lego-like reconfigurable frameworks," 2024. [Online]. Available: https://arxiv.org/abs/2407.21059