An Architectural Analysis of an AI-Infused Mobile Application

Nate Seon, Sam Chung
School of Technology & Computing, City University of Seattle

I. SUMMARY

This research presents and analyzes a deployment architecture for a Generative AI-powered mobile app in the cloud using Retrieval Augmented Generation (RAG). We implement a React Native mobile app that utilizes the Gemma 2 Large Language Model (LLM) and LangChain. We document a visual representation of the app's deployment architecture using Unified Modeling Language (UML) and analyze the deployment diagram to discuss its architectural styles. This architectural analysis aims to enhance collaboration within the development team by providing a clear understanding of the system architecture.

II. SYSTEM ARCHITECTURE

Frontend: React Native Mobile/Web App
Backend/API: Express.js (Node.js)
LLM Server: Ollama + Gemma 2
Vector DB: ChromaDB
Metadata DB: MongoDB Atlas
MLOps: MLflow, FastAPI for model tracking

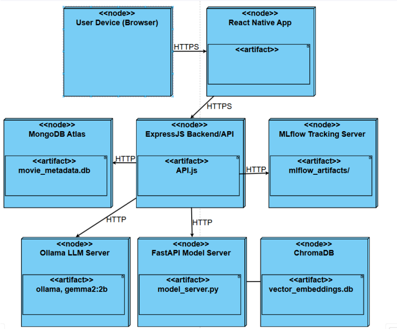

Figure 1. Deployment Architecture of a RAG-based AI Mobile Application.

III. IMPLEMENTATION

The RAG flow of the system is realized by the integration of these components. The React Native frontend interacts with the Express.js backend. Express.js orchestrates data retrieval from MongoDB Atlas and semantic search from ChromaDB. The resulting context is then sent to Gemma 2, served by Ollama, for on-device LLM inference. Optional MLflow and FastAPI serve experiment tracking and model serving.

IV. RESULT & VALIDATION

The system's core functionality was validated by end-to-end test scenarios and component-level testing.

Component	Verification Output
Ollama LLM Server	Gemma2 running locally
ChromaDB	Embeddings indexed & searchable
MongoDB Atlas	Metadata stored & retrieved
Express.js	API/chat functioning

Test Cases:

- Registered movie ("Panic"): Returns proper information.
- Unregistered movie ("Harry Pottle"): Responds with a "No info" response, demonstrating good data handling.

V. CONCLUSION

Our proposed architecture offers a scalable, cost-effective, and privacy-improving solution to AI-driven recommendations and chat in mobile/web applications. By leveraging RAG, updating in real-time is achievable without prolonged retraining of LLM weights, which offers efficiency. Moreover, by utilizing a local LLM and encrypted cloud storage, core data processing largely remains on-premises or within controlled environments, which enhances privacy.

REFERENCES

Ngo, C., & Chung, S. (2024). CS628 Full-Stack AI Development. City University of Seattle.

Smith, B. (2025, May 8). Winning the AI race: Strengthening U.S. capabilities in computing and innovation. Microsoft on the Issues. https://blogs.microsoft.com/on-the-issues/2025/05/08/winning-the-ai-race