Gregory Burns

burnsgregm@gmail.com | 831-226-6907 | San Francisco Bay Area
linkedin.com/in/gregburns | github.com/burnsgregm | burnsgregm.netlify.app/portfolio.html

AI Empower Enterprise RAG — One-Page Summary

Author: Greg Burns

Role: Machine Learning Engineer • Systems Architect

Tech Stack: Python · GCP (Cloud Run, Pub/Sub, Eventarc, GCS, Firestore Native + Vector) · Vertex AI (Gemini 2.5 Pro, text-embedding-004) · Streamlit · Terraform · Pandas/PyArrow

Overview

Demo RAG V5 is the live, demo-facing implementation of the AI Empower Enterprise RAG Service, built for a medical education platform. It ingests large PDF/PPTX documents (e.g., textbooks, lecture decks), indexes them with a Parent–Child vector strategy in Firestore Vector Search, and serves tenant-aware, conversational answers powered by Gemini 2.5 Pro.

V5 focuses on storage decoupling: the ingestion worker now performs a dual-write to Firestore (for real-time retrieval) and Parquet in Cloud Storage (for future migration to a dedicated vector engine such as Vertex AI Vector Search), enabling library-scale growth without re-ingesting the corpus.

Key Capabilities

Enterprise RAG Backend

- Asynchronous ingestion pipeline for large, multi-hundred-page documents.
- Multi-tenant design using client_id filters for strict data isolation.
- Parent–Child chunking:
  - Parent chunks (~2,000 chars) for rich LLM context.
  - Child chunks (~400 chars) for precise vector similarity search.

Storage Decoupling (V5 Feature)

- Dual-write ingestion worker:
  - Writes embeddings and metadata to Firestore Vector Search for low-latency retrieval.
  - Writes the same records to Parquet in GCS for offline indexing and analytics.
- Enables a clean V6 path to swap or augment the vector engine without changing client behavior.

Retrieval & Generation

- Vector search over child chunks, then resolution to parent chunks for context assembly.
- RAG pipeline incorporates:
  - Retrieved passages
  - Tenant context
  - Recent chat history (last few turns per session_id)
- Uses Gemini 2.5 Pro for grounded, conversational answers with source snippets.

Business Impact

Demo RAG V5 shows how to take an RAG prototype and evolve it into a platform-ready service with: Scalability: Fan-out ingestion enables reliable processing of large documents without timeouts.

- Cost & Future-Proofing: Dual-write storage decouples real-time search from long-term vector storage and analytics.
- Tenant Isolation & Governance: Per-tenant filtering and session-based memory align with enterprise data boundaries.
- Stakeholder Trust: A clean demo UI on top of the real backend makes the system understandable and testable by non-engineers.

This project demonstrates my ability to architect and implement end-to-end, cloud-native RAG systems on GCP, balancing real-time retrieval, scalability, and long-term platform evolution.