Emilio Gagliardi

RÉSUMÉ · AUG 2026

emilio.gagliardi@gmail.com | 780.885.9850 | emilio-gagliardi.ca

AI engineer and platform architect who takes retrieval-augmented AI systems from first commit to paying customers. Most recently led technical development at Policy Solver: a multi-tenant graph RAG platform through which municipal governments query their bylaws, policies, and procedures and receive cited, auditable answers. Sole architect of the whole system - ingestion pipeline, relational, vector and graph databases, the FastAPI retrieval server, and the Azure ecosystem and environments that ran it. Builds with modern agentic coding tooling under spec-driven discipline. Grounded in experimental science: a research master's built a lasting practice of hypothesis testing, calibrated thresholds, and evidence over intuition.

01 EXPERIENCE

Policy Solver · Lead AI Engineer

Jun 2025 – Jul 2026

Contract Jun 2025; full-time Jan 2026. Led technical development of a multi-tenant, retrieval-augmented AI platform for municipal governance · **five production repositories, ~540K lines of code**.

- **Retrieval engine · FastAPI RAG server:** Built the production retrieval service end to end - a three-lane architecture running document-summary, section-chunk, and Neo4j graph lanes concurrently under asyncio, merged through weighted reciprocal-rank fusion and Cohere cross-encoder reranking, with an intent classifier that routes each query to a retrieval policy and short-circuits conversational turns at zero retrieval cost. Wired the service into Langfuse to track token consumption per lane, run prompt tournaments against held-out query sets, and monitor retrieval performance in production.
- **Graph extraction · custom classifiers:** Researched, trained, and shipped the two classifiers the extraction pipeline depends on: a SetFit/ONNX section classifier and a novel clause classifier. Both replaced brittle hand-tuned heuristics and cut extraction cost by keeping routine decisions off the LLM, and both can be fine-tuned per customer and per corpus, so extraction adapts to each municipality's drafting conventions instead of being fixed at design time.
- **Data platform · three database classes, one retrieval system:** **PostgreSQL/Supabase** with row-level-security tenant isolation and RPC-driven task queues; **Zilliz/Milvus and Qdrant** vector stores running hybrid dense + BM25 sparse retrieval, including provisioning and serving of the dense and sparse embedding models; and a **Neo4j knowledge graph** synchronized from Postgres through typed column contracts that make schema drift a build failure rather than a production incident. Graph retrieval ran through a rule-based Cypher query planner with tenant guards, circuit breakers, and a structured error taxonomy.
- **Ingestion pipeline:** Built the Dagster Cloud document-intelligence pipeline end to end - SharePoint ingestion over Microsoft Graph combining delta-based change detection with scheduled full-catalog polling, Azure Document Intelligence parsing, a purpose-built normalized semantic chunker, summarization, entity extraction, and embedding - sensor-driven, with lazy task fan-out, adaptive concurrency, and automated replica scaling.
- **Agentic search surface:** Built an MCP server that exposes the corpus as agentic search tools, letting assistants and coding agents query provincial and municipal documents through the same tenant-guarded retrieval stack behind the product.
- **Observability:** Built the Langfuse observability layer with deep tracing from document ingestion to cited answer, plus long-lived file-based prompt and response assets that keep every generation reproducible and inspectable long after the trace window closes - the layer that made retrieval-quality regressions discoverable instead of anecdotal.
- **Environments and cloud platform:** Stood up and operated isolated test and production environments across the entire stack, and built the original Azure application ecosystem: Container Apps, Blob Storage, Key Vault, and Document Intelligence. Containerized each application and shipped it through Azure Container Registry and GitHub Actions CI/CD. Also delivered the original SharePoint implementation, covering content-type schemas, metadata governance, Graph API provisioning, and tenant onboarding tooling.
- **Engineering standards and governance:** Authored the engineering system other developers (human and agent) work within: 13 enforced coding standards, cross-repository error-code taxonomies, and the operational runbooks that made the platform runnable by people other than its author.

PortalFuse · Data & AI Consultant

2023 – 2025

Built the automated security-intelligence reporting product for a Windows patch-management SaaS.

- Engineered **Kedro** ingestion-transformation-publication pipelines aggregating CVE and KB-article data from the National Vulnerability Database and the Microsoft Security Response Center across Windows and browser products.

- Designed and automated a weekly, monthly, and quarterly reporting cadence. The quarterly edition analyzes ~250 CVEs across CVSS v3.1 scoring, CWE categorization, and severity and exploitability distributions in 19 visualizations, then closes with strategic guidance addressed to CISOs.
- Delivered dual-perspective analysis - CVE-centric risk and KB-centric remediation - so administrators move from threat to patch inside a single document.

Wild Rose College · Technical Lead

2021 – 2023

- Directed technology across ecommerce, LMS, and marketing systems; rebuilt the site theme spanning WooCommerce and LearnDash; advised on IT budget and vendor selection.
- Designed the ActiveCampaign automation integrating WooCommerce and LearnDash that tracked and communicated with thousands of customers and drove **$3.5M in first-year sales**.

Earlier career · Founder & Owner

2011 – 2022

Founded and ran **Uplyft** (2015–2022), a web-development consultancy modernizing retail clients' web properties, and **Eco Evolver** (2011–2015), a controlled-environment agriculture design and education company - including a fully automated 25′ × 7′ living-wall installation for an elementary school.

02 AGENTIC ENGINEERING PRACTICE

- Run a supervised, spec-driven agentic delivery system at the IDE level across **Claude Code, OpenAI Codex, and Hermes**: 450+ OpenSpec architecture proposals implemented across five repositories, each through explicit plan → agent implementation → multi-model code review → feature-series review. Roughly a third of commits in the flagship repository record review-driven corrections - measured evidence that frontier-model output requires structural verification, not trust.
- Built the guardrails coding agents operate within: 20 purpose-built agent skills, 9 agent personas, and 21 blocking rules across the platform, each typically distilled from a caught defect. Example: an agent-authored change passed every unit test and failed in production against the live schema; the fix became a generated column-contract layer with live-schema CI guards - an entire failure class converted into a build error.
- Hold a clear division of labor: agents accelerate implementation; humans own architecture, review, and the long-term health of the codebase. Work from agentic and graph-based views of the code rather than file-based ones, monitoring decision quality, duplication, and structural debt as the codebase accumulates.

03 EDUCATION & CREDENTIALS

MSc, Psycholinguistics · University of Alberta. Experimental design, statistical inference, and computational study of language - the research foundation of the data science practice.

BA (Hons), Psychology · University of Alberta

Microsoft Professional Program in Data Science (2019) · **DataCamp Data Scientist Associate** (2023)

04 TECHNICAL SKILLS

AI / ML	RAG and graph RAG architecture, hybrid retrieval (dense + sparse, weighted RRF), Cohere and cross-encoder reranking, intent classification and query routing, knowledge-graph extraction, classifier training and fine-tuning, Haystack, LiteLLM, SetFit, ONNX, spaCy, BGE-M3 embeddings, semantic chunking
AGENTIC ENG.	Claude Code, OpenAI Codex, Hermes and coding-agent harnesses; spec-driven development (OpenSpec); agent skill and rule authoring; multi-model code review orchestration; MCP server development; graph-based code analysis
DATA PLATFORMS	PostgreSQL/Supabase (RLS multi-tenancy, RPC design, migrations), Neo4j/Cypher, Milvus/Zilliz, Qdrant, Redis, Azure Blob Storage
ORCHESTRATION	Dagster (Cloud Serverless), Kedro, Airflow, Azure Data Factory
CLOUD & DEVOPS	Azure (Container Apps, Container Registry, Document Intelligence, Key Vault), Docker, GitHub Actions CI/CD, pytest, mypy, ruff
LANGUAGES / APIS	Python, SQL, JavaScript/TypeScript; FastAPI, Microsoft Graph, SharePoint provisioning, REST
OBSERVABILITY	Langfuse tracing, prompt management and cost tracking; structlog; error-code taxonomy design; runbooks