



LLM Integration Developer

Remote (Canada) | ~\$65,000-90,000/year

AI-powered agronomic interpretation · RAG pipelines · Agent design · Automated reporting

Overview

Vintality is a precision viticulture platform serving vineyard managers across the Okanagan Valley, BC. The platform combines LoRaWAN IoT sensor networks (soil moisture, temperature/humidity, weather stations), agronomic models running as AWS Lambda functions, GIS-based block management, and a React dashboard.

We are hiring an LLM Integration Developer for a one-year contract to build an AI layer across the platform — starting with a conversational interface for querying farm data, expanding into automated reporting, intelligent alerting, and a sensor support knowledge base.

This is a technical build role, not a research role. You will be shipping production features on a live platform used by farm supervisors managing real vineyards.

The problem we are solving

Farm supervisors currently navigate multiple data streams — soil moisture per block, canopy temperature and humidity, disease risk scores, ETref calculations, irrigation deficits, spray windows, frost forecasts — across a structured dashboard. The data is there. What is missing is interpretation: contextual, agronomic language that connects the numbers to decisions.

A supervisor checking Block 5 at 6am should be able to ask: "Is it safe to irrigate today?" or "Why is PM risk spiking on my Cab Franc blocks?" and get a grounded answer that references their actual sensor readings, block history, and current model outputs — not generic advice.

Beyond the conversational interface, the same underlying capability — LLM reasoning over structured farm data — enables automated weekly reports, contextualised alert notifications, onboarding guidance for new farms, and a self-service support layer for sensor troubleshooting.



Scope of work — full year

Phase 1: Conversational farm query interface (Months 1–4)

The core deliverable. A chat interface embedded in the farm dashboard that allows supervisors to ask natural language questions about their farm data.

Query types to support:

- Soil and irrigation queries — current VWC by block, deficit, ETref, last irrigation date, Lumo delivery history
- Canopy environment queries — temperature, humidity, VPD per block, multi-sensor averaging and divergence
- Disease risk queries — PM and Botrytis risk scores per block, driving conditions, spray window status
- Weather and forecast queries — wind, solar radiation, frost probability, GDD season progress
- Cross-block comparison — which blocks are most at risk, which need attention first
- Historical context — how does today compare to last week, last season

Technical approach:

- RAG pipeline over structured sensor data — PostGIS / RDS as the data source, not document retrieval
- Tool-use / function calling pattern — LLM calls structured data functions rather than having raw data injected into context
- Agronomic interpretation layer — prompt design that grounds responses in viticulture domain knowledge
- Context window management — farm profile, block configuration, varietal data, current season stage injected as system context
- AWS Lambda integration — query functions that retrieve current and historical sensor readings, model outputs, and irrigation events

Phase 2: Automated farm reporting (Months 3–6)

Weekly and monthly farm summary reports generated automatically from sensor and model data. Delivered in-app and by email.

- Weekly summary — irrigation actions taken vs recommended, disease risk trajectory per block, notable weather events, sensor health
- Monthly report — ETref vs actual irrigation delivered (Lumo integration), GDD accumulation vs 5-year average, spray program compliance, block-level performance summary



- Season-end report – full agronomic summary per block, yield correlation data where available, sensor placement recommendations for next season
- Report generation pipeline – scheduled Lambda trigger → data aggregation → LLM narrative generation → PDF/email delivery

This reuses the same data retrieval and prompt infrastructure as the chat interface – it is essentially the conversational interface running on a schedule rather than on demand.

Phase 3: Contextualised alert interpretation (Months 4–7)

When the platform fires an alert – frost risk, PM spike, irrigation deficit, sensor offline – the LLM adds a narrative layer that contextualises it against farm history and suggests a response.

- Frost alert enrichment – "Block 3b lower slope sensor is reading 1.8°C below block average. Based on tonight's forecast, this block has the highest frost exposure on the farm."
- Disease risk narrative – "PM risk on Block 5 Cab Franc has increased 28 points in 48 hours. Driving conditions: RH sustained above 72%, canopy temp 20°C. Last spray was 9 days ago."
- Irrigation urgency context – "Block 3 deficit has reached 38mm. ETref has averaged 3.4mm/day this week. Without irrigation in the next 24 hours, you will be below 45% field capacity."
- Sensor anomaly flag – "T/H-B3b has read 3°C above block average for 6 consecutive days. This may indicate sensor drift, a mounting position issue, or a genuine microclimate. Recommend field inspection."

Phase 4: Sensor support and troubleshooting knowledge base (Months 5–8)

A RAG-powered support layer that allows supervisors and field technicians to ask questions about sensor installation, calibration, connectivity, and troubleshooting without needing to contact support.

- Knowledge base sources – Node documentation, LoRaWAN/TTN troubleshooting guides, AWS IoT Core diagnostics, Lumo integration guides, internal installation SOPs
- Sensor placement advisor – given a block's polygon geometry, varietal, row orientation, and topography, recommend optimal sensor placement for T/H, soil, and rain gauge nodes
- Connectivity diagnostics – query TTN gateway coverage, last seen timestamps, RSSI/SNR values, suggest remediation
- Calibration guidance – soil sensor calibration workflows by soil type, T/H sensor verification against reference readings

Phase 5: Onboarding assistant (Months 7–10)

New farm onboarding is currently a manual process. An LLM-guided onboarding flow reduces setup time and ensures farms are configured correctly from day one.

- Block configuration wizard — guides supervisors through uploading block polygons, assigning varieties, phenological stage, and sensor assignments
- First-season calibration guidance — soil sensor baseline period, ETref model warm-up, disease model activation thresholds
- Sensor placement recommendations — based on block area, variety, and topography inputs
- Integration setup support — Lumo valve configuration, TTN gateway registration, Visual Crossing API connection

Phase 6: Refinement, evaluation, and documentation (Months 9–12)

- Prompt evaluation framework — systematic testing of agronomic response accuracy against known scenarios
- Latency and cost optimisation — caching strategies, model selection tuning, context window efficiency
- Hallucination guardrails — validation layer that checks LLM responses against source data before surfacing to users
- Full technical documentation — agent architecture, prompt library, RAG pipeline design, evaluation methodology
- Handoff to core engineering team for ongoing maintenance

Delivery roadmap

| Phase | Key deliverables |
|-------------------------------|---|
| Phase 1 Months 1–4 | Conversational query interface · RAG pipeline · Tool-use function library · Chat UI component |
| Phase 2 Months 3–6 | Automated weekly/monthly report pipeline · PDF generation · Email delivery |
| Phase 3 Months 4–7 | Alert enrichment layer · Contextualised frost/disease/irrigation/sensor narratives |
| Phase 4 Months 5–8 | Support knowledge base · Sensor placement advisor · Connectivity diagnostics |
| Phase 5 Months 7–10 | Onboarding assistant · Block configuration wizard · First-season calibration guidance |

Phase 6

Months 9–12

Evaluation framework · Cost/latency optimisation · Hallucination guardrails
· Full documentation

Technical stack

Must have

- LLM API experience — Anthropic Claude or OpenAI GPT class models, tool use / function calling patterns
- RAG pipeline design — retrieval from structured data sources (not just document stores), context assembly, prompt construction
- Python — Lambda functions, data retrieval, prompt orchestration
- AWS — Lambda, API Gateway, RDS (PostgreSQL), S3, CloudWatch
- Prompt engineering — system prompt design, few-shot examples, chain-of-thought reasoning, output validation
- REST API design — endpoints for the React frontend to call the agent layer

Strong advantage

- LangChain, LlamaIndex, or similar orchestration framework experience
- Agricultural, environmental, or scientific domain data experience — you understand what ETref, VPD, and field capacity mean, or are willing to learn quickly
- React — enough to integrate a chat component and report viewer into an existing dashboard
- Streaming LLM responses — SSE or WebSocket patterns for real-time chat feel
- Evaluation frameworks — LLM output testing, automated accuracy benchmarking
- Vector database experience — pgvector (already available in PostGIS stack) or Pinecone/Weaviate for knowledge base retrieval

Stack you will integrate with

- AWS IoT Core → Lambda → RDS (PostgreSQL/PostGIS) — sensor data pipeline
 - GeoServer / PostGIS — spatial farm data, block geometries
 - Lumo API — irrigation event data
 - Visual Crossing — weather data
 - React frontend — existing dashboard where the AI features will surface
 - 12 agronomic Lambda functions — PM risk, Botrytis, ETref, frost, GDD, spray window, and others
-



What this role is not

This is not an ML engineering role – you will not be training or fine-tuning models. You will be building production integrations with frontier LLM APIs.

This is not a data science role – the agronomic models already exist as Lambda functions. Your job is to make their outputs accessible and interpretable through a conversational interface.

This is not a frontend-only role – the most important work is backend: the RAG pipeline, the tool-use function library, the prompt architecture, and the data retrieval layer.

Logistics

CONTRACT TYPE

One-year fixed-term contract with room for extension.

LOCATION

Remote. Any. BC is a bonus but not required.

COMPENSATION

CAD \$65,000 – \$90,000 for the year depending on experience, structured as monthly payments against milestones. Open to hourly rate equivalent for the right candidate.

REPORTING

Reports directly to the platform lead. Weekly check-ins via video. Async collaboration via Slack and GitHub.

START DATE

Flexible. Targeting early Q2 2025.

How to Apply

Send your resume and a short cover letter to:

hiring@vintality.com

Please include any work samples or portfolio links, and answer the screening questions (below).



Screening questions

- 1.** Describe a project where you built a RAG pipeline or LLM integration that queried structured data (database, API, or time-series) rather than documents. What was the retrieval strategy?
 - 2.** Have you used tool use / function calling with Claude or GPT-4? Describe how you structured the tools and handled multi-step reasoning.
 - 3.** How do you approach preventing hallucinations in a domain where accuracy matters – for example, if an LLM confidently states the wrong irrigation recommendation?
 - 4.** What is your experience with AWS Lambda and PostgreSQL? Have you built data retrieval functions that an LLM calls via tool use?
 - 5.** Share a GitHub repo, project writeup, or demo that shows LLM integration work you have built.
-