Marketplace Optimisation Agents

# Abstract

BlueMart's online marketplace ingests millions of seller‑supplied SKUs every month. Images are often low‑resolution, copy is incomplete or non‑compliant, and search‑critical keywords are missing. Manual QA cannot keep pace, resulting in long approval queues, lower search ranking, and lost GMV. Content‑Perfect proposes a Google Cloud‑native, LangGraph‑orchestrated agent suite that enriches content, enforces policy compliance, and exposes a high‑quality catalogue through an agent‑to‑agent (A2A) OpenAPI---reducing **time‑to‑publish from 24 h to < 5 min** and lifting SKU conversion by ≥ 10 %.

# Background

* **Company**: BlueMart – retail giant with chains in 30+ cities plus an e‑commerce platform.
* **Scale**: Thousands of third‑party sellers, millions of SKUs, high daily churn.
* **Pain Points**:  
  • Manual listing review delays go‑live.  
  • Vendors complain that “the algorithm hides my product.”  
  • Brand‑consistency & legal exposure from unvetted copy.

# Objective & Success Metrics

Design and implement an automated pipeline that:

1. Crafts SEO‑optimised titles, rich descriptions, and hero images.
2. Scores each submission against a policy knowledge graph and offers fix‑tips.
3. Publishes or rejects listings in < 5 min with minimal human touch.
4. Serves a curated catalogue to shoppers and external agents.

**Target KPIs**

* *Time‑to‑publish*: 24 h → **< 5 min**
* *SKU conversion*: **+10 %** uplift
* *Vendor rejection rate*: 18 % → **< 3 %**
* *AnswerBot top‑3 accuracy*: **≥ 95 %**

# Data Assets

|  |  |  |
| --- | --- | --- |
| Asset | File | Purpose |
| Compliance Rules | compliance\_rules.txt | Knowledge base for Inspector Agent. |
| Seller Submission Samples | seller\_submission\_samples.csv | Ten draft listings used by Phase 1 for ingestion tests. |
| Content Crafter Samples | content\_crafter\_samples.csv | Ten partially filled listings the Crafter Agent enriches. |
| Product Catalogue | catalog\_answer\_dataset.sqlite (table **catalog**) | Master dataset storing **both sample SKUs and all subsequent vendor uploads** after compliance validation. |
| Category Taxonomy | Google Retail Taxonomy (taxonomy.en‑US.txt) <https://www.google.com/basepages/producttype/taxonomy.en-US.txt> | Reference list for category & attribute validation. |
| Sample Images | Any image files | Used to test vision quality checks; linked URLs in datasets are dummy placeholders. |

# Solution Design & Detailed Phases

**Phase 1 – Seller Submission Intake & Pre‑Processing**

* Marketplace exposes a **REST endpoint /v1/sku/submit** for vendors, mirroring the schema in vendor\_submission\_api\_sample.csv.
* Incoming JSON/CSV payloads are staged in **Cloud Storage** and inserted into **SQLite (catalog, status = "draft").**
* Metadata extraction, language detection and HTML sanitisation run inline; optional raw archives go to **Firestore.**

**Phase 2 – Content Crafter Agent**

* LangGraph node calls **Vertex AI Gemini** for SEO titles, keyword‑rich descriptions, and attribute extraction.
* Generates hero images via **Vertex AI Imagen**; dummy URLs are stored for the capstone.
* Outputs persist to Cloud Storage and update the same SQLite row.
* Agent snapshot: The Crafter is an autonomous "**copy‑and‑creative**" **specialist**: it transforms sparse vendor data into search‑optimised, policy‑ready content while logging its prompt chain for audit and later RLHF.

**Phase 3 – Compliance Inspector Agent**

* Runs rule checks from compliance\_rules.txt and validates category alignment via **Google Retail Taxonomy (Vertex AI Vector Search).**
* Scores each listing 0--100, attaches inline fix‑tips, and updates status → "approved" / "needs\_fix" / "rejected" in SQLite.
* Fail low‑scoring items or route them to vendor UI for self‑service correction.
* Agent snapshot: The Inspector is the marketplace's **governance gatekeeper**---it enforces legal, brand, and safety policies, providing granular remediation guidance. It consumes Crafter output and emits a structured verdict for the Orchestrator.

**Phase 4 – Orchestrator Workflow**

* Compose a LangGraph workflow that sequences Crafter → Inspector → Decision.
* Edge‑case events push to **Pub/Sub** for human review. Encrypt secrets (API keys, connection strings) with Secret Manager.
* Flow manager: The Orchestrator is a lightweight **state and error‑handling brain**; it ensures idempotent updates to SQLite, orchestrates retries, and surfaces audit events to monitoring.

**Phase 5 – Catalog Answer Agent & A2A OpenAPI**

* Build a read‑only agent that surfaces curated catalogue answers via an **OpenAPI 3.1** spec. (it servers the products where status="approved")
* Support agent‑to‑agent (A2A) queries for product recommendations, stock availability, and price.
* Cache popular queries in **Memorystore for Redis** for sub‑100 ms latency.
* Agent snapshot: The Answer Agent is a **conversational catalogue concierge**---it turns SKU rows into coherent answers, recommendations, or JSON payloads for external bots, always reflecting the latest approved data in SQLite.

**Phase 6 – Observability & Guardrails**

* Instrument every LangGraph node for prompt‑level tracing, token usage, and latency using Cloud Logging.
* Stream system metrics (CPU, memory, latency, error rate) to **Cloud Monitoring**; create dashboards for visualization.
* Enable Vertex AI safety APIs to block harmful or policy‑violating outputs.

**Phase 7 – Containerisation & Deployment**

* Package each agent and the orchestrator as Docker images; push to **Artifact Registry.**
* Deploy to **Cloud Run** with auto‑scaling rules.
* Configure CI/CD via **Cloud Build** triggering on main‑branch merges.

**Phase 8 – Pilot Roll‑out & Continuous Improvement** (optional)

* Activate the pipeline on 5 % of daily submissions; compare KPIs against control cohort.
* Fine‑tune prompts, policy thresholds, and image model parameters based on Cloud Logging analytics.
* Gradually increase traffic to 100 % after KPI targets are met.

# Architecture Overview

The processing pipeline is orchestrated by **LangGraph**, managing the flow through distinct phases:

* **Content Crafter Agent** enriches the incoming data using Vertex AI Gemini model, creating SEO-optimized product titles, detailed descriptions, and extracting relevant product attributes. Additionally, the Crafter generates hero images using Vertex AI Imagen. Enriched content and metadata are logged via Cloud Logging for observability and persisted back into Cloud Storage, updating the relevant SQLite entry.
* **Compliance Inspector Agent** evaluates enriched content using predefined compliance rules (compliance\_rules.txt) and category alignment checks with the Google Retail Taxonomy indexed in Vertex AI Vector Search. The agent assigns each listing a score, offers inline fix tips for minor issues, and updates listing statuses in SQLite ("approved," "needs\_fix," or "rejected"). Significant compliance issues trigger automatic routing to vendor self-service UIs for corrections.
* **LangGraph Orchestrator** manages the overall workflow, sequencing the agents, handling edge-case escalations via Pub/Sub (for human intervention), and securely managing system secrets through Secret Manager.
* **Catalog Answer Agent** serves as the final interaction point, exposing approved product listings through a read-only OpenAPI 3.1 compliant API. It efficiently handles agent-to-agent (A2A) queries, caching frequent requests using Memorystore for Redis to maintain sub-100 ms response times.

# Expected Deliverables

1. **Source Code**
   * LangGraph workflow definitions and node handlers.
   * Prompt templates and image‑generation scripts.
2. **Google Cloud Configuration**
   * Cloud Deployment Manager templates for all GCP resources.
   * Container image URIs in Artifact Registry.
   * Cloud Run service configurations.
   * Secret Manager and IAM policy configurations**.**
3. **OpenAPI Specifications**
   * Seller Submission API
   * A2A Catalog Answer API.
4. **Observability Artifacts**
   * Cloud Monitoring dashboard configurations.
   * Cloud Logging query templates and alert policies.
5. **Operational Runbook**
   * Deployment guide, scaling procedures, rollback steps, incident playbook.
6. **Security & Compliance Docs**
   * Threat model, data‑flow diagrams, PII handling statement.