UCP // FLUENT

ARCHITECTURAL SPECIFICATION

Middleware Environment: Google Cloud Platform (GCP)

Scope: High-Availability Catalog Mapping & UCP Protocol Integration

1. Core Architectural Pillars

UCP Fluent is an enterprise-grade middleware stack bridging legacy enterprise OMS backends with the Universal Commerce Protocol (UCP). The infrastructure is designed to process massive catalog payloads into machine-callable datasets for AI agents.

Semantic Constraint-Logic

Replacing unstructured text with high-density semantic attributes. By mapping 100,000+ variants to exact UCP standards, the system enables constraint-based AI filtering (e.g., medical-grade ISO standards, exact color hexes).

State Integrity (ATS Optimization)

Maintaining strict data parity and sub-200ms latency to prevent agentic hallucination risks. The system utilizes continuous synthetic transaction monitoring to protect the merchant's global Agentic Trust Score (ATS).

Native Interoperability

Utilizing the Agent Payments Protocol (AP2) to process transactions natively within AI interfaces, bypassing legacy redirect flows.

2. Infrastructure Stack Overview

Control Plane	Management Hub & Cache	Cloud Run (Node/Remix) + Cloud Memorystore (Redis)	Orchestrates the secure Native OMS → GCP handshake, utilizing Redis for sub-second session state caching and high-speed data retrieval.
Discovery	MCP Server	Model Context Protocol	Operates as the low-latency endpoint serving UCP manifests to AI agents.
Identity	Registrar Engine	Custom Provisioning & Import Engine	Flexible programmatic generation of GTINs (by prefix or single format) and seamless ingestion of legacy third-party codes for unique variant identity.

Intelligence	Semantic Engine	Vertex AI (Gemini 2.5)	Advanced multimodal vision processing for automated, high-density attribute extraction.
Data	Lakehouse	BigQuery + Vertex AI	Semantic vector storage for high-speed catalog query execution.

Action	Payment Routing	AP2 Protocol	Native encrypted payment mandates via supported agentic vaults.

3. Zero-Friction Ingestion Vectors

To ensure rapid enterprise deployment without custom engineering overhead, UCP Fluent utilizes an asynchronous pull architecture. The middleware extracts catalog payloads via three standardized vectors:

- **Vector 1 — OAuth Native App:** Secure, read-only catalog extraction via platform-agnostic API handshakes.
- **Vector 2 — API Key Synchronization:** Direct polling of existing OMS GraphQL/REST endpoints using client-generated, read-only API keys.
- **Vector 3 — Automated Feed Ingestion:** Scheduled, high-frequency ingestion of existing XML/CSV data lakes (e.g., Google Shopping feeds).

4. Data Flow & Machine Learning Pipelines

Loop 1: The Identity Pipeline (Automated Provisioning)

- **Trigger:** Merchant initialization and catalog sync.
- **Action:** System engages the Registrar Engine to evaluate existing catalog identity markers. • **Outcome:** Up to 100,000 variants are assigned unique identifiers (dynamic prefix generation, single format creation, or legacy import mapping), staged in BigQuery, and written back to the merchant's variant records.

Loop 2: The Vision Pipeline (Multimodal Enrichment)

- **Trigger:** Detection of incomplete physical attribute data within the catalog.
- **Action:** Vertex AI (utilizing Gemini 2.5 via Cloud Dataflow) analyzes variant imagery.
- **Outcome:** Accurate hex codes, material compositions, and structural patterns are mapped and pushed to Google Merchant Center (GMC) Supplemental Feeds.

Loop 3: The Validation Pipeline (Synthetic Monitoring)

- **Trigger:** High-frequency scheduled intervals.
- **Action:** Google Cloud Functions simulate agentic queries against the merchant endpoint (e.g., verifying price and shipping calculation latency).
- **Enforcement:** If response exceeds latency thresholds or logic fails, the system temporarily toggles protocol readiness state to protect the merchant's ATS.

5. Edge Hosting & Protocol Deployment

Translated UCP JSON-LD payloads are not injected back into the legacy client codebase. UCP Fluent hosts the deterministic schema on Google Cloud CDN. Client deployment requires zero code — only the configuration of a single CNAME DNS record:

```
ucp.merchantdomain.com → edge.ucpfluent.com
```

This architecture isolates compute load from the client's core storefront, eliminating performance risk and reducing enterprise deployment friction to a single DNS configuration.

6. Operational Scaling Strategy & GCP Utilization

API Resilience

The system utilizes Platform-Agnostic GraphQL Bulk APIs alongside Google Cloud Storage to bypass standard REST throttling during 100,000+ variant syncs.

Real-Time Fidelity

Google Cloud Pub/Sub manages continuous inventory webhooks, updating the UCP layer in sub-second timeframes to prevent overselling anomalies.

Compute Efficiency

Vertex AI Batch Prediction is heavily utilized to optimize initial multimodal vision processing for massive catalog ingestions, prior to real-time agentic exposure.