Product Strategy Agent

# Abstract

**BlueCola,** a global beverage leader, is seeking to accelerate product innovation cycles without relying on expensive proprietary research or delayed insights from traditional panels. To achieve this, the company aims to harness publicly available data---news articles, social chatter, and search trends---through intelligent LLM-driven agents.

The **Public-Data Product Strategy Agent** addresses this by orchestrating four autonomous LangGraph agents:

* **TrendSpotterAgent** scrapes and clusters headlines into emerging consumer themes.
* **SocialSentimentAgent** mines Reddit posts to extract sentiment, pain points, and enthusiasm markers.
* **ConceptGenAgent** translates these insights into fresh product concept pitches.
* **RoadmapPlannerAgent** outlines a high-level launch plan with key deliverables and estimated resources.

All components are modular and integrated into a FastAPI backend, with endpoints for each insight stream and a React/Next.js UI for interactive exploration. Orchestration is driven by LangGraph and Azure Functions, and monitoring is handled via Azure Monitor and Azure Log Analytics.

This low-cost, real-time strategy agent empowers BlueCola's innovation teams to explore and validate ideas faster---resulting in faster go-to-market, better alignment with market needs, and a leaner R&D investment.

# Background

In the hyper-competitive Consumer Packaged Goods (CPG) industry, beverage companies like **BlueCola** must continuously introduce new variants and flavors to retain market relevance. Traditional innovation cycles are hindered by:

* **High cost of proprietary research**, focus groups, and trend agencies.
* **Long lead times** for insight generation and validation.
* **Reactive decision-making**, often missing emerging micro-trends.

To stay ahead, BlueCola seeks an **LLM-driven agentic platform** that can:

* Monitor public signals continuously.
* Generate concept ideas autonomously.
* Deliver go-to-market plans rapidly and with minimal manual effort.

# Objectives & Success Metrics

Design and implement a fully agentic, publicly-driven product strategy engine using Azure tools, enabling CPG product managers to:

1. Discover emerging trends from public data.
2. Assess market sentiment from social media.
3. Autonomously ideate new beverage product concepts.
4. Translate concepts into actionable product roadmaps.

**Success Metrics**

|  |  |
| --- | --- |
| Goal | Target Outcome |
| Trend Extraction Latency | < 30 seconds per batch |
| Sentiment Clustering Accuracy | ≥ 90% relevance in manual cross-check |
| Concept Generation Diversity | ≥ 3 distinct variants per trend |
| Roadmap Generation Completion | ≤ 10 seconds per request |
| System Stability (Uptime) | ≥ 99% with alerting |
| API Response Latency (avg) | < 1 second per endpoint |

# Data Assets

The following public-data APIs and integrations are available to the capstone team:

**News & Search APIs**

* **NewsAPI.org** – Real-time and historical articles from global and niche publications.
* **Tavily Search API** – Curated search insights for trend summarization and clustering.

**Social Media Mining**

* **Reddit API (via PRAW)** – Posts and comments from target subreddits like r/beverages, r/CPG, r/healthydrinks.

**Sample Tests**

* Simulated inputs for trend clustering, sentiment score validation, prompt tests for concept generation, and roadmap templates.

# Solution Design & Phases

The **Public-Data Product Strategy** Agent is composed of four specialized agents, each focused on a specific sub-task in the product strategy lifecycle. These agents operate independently but are orchestrated using a LangGraph-based flow, which enables intelligent sequencing and conditional routing. The overall solution is implemented in modular phases:

**Phase 1: TrendSpotterAgent (Trend Identification)**

* **Task:** Identify high-velocity emerging trends in the beverage and wellness domains.
* **Tools:**
  + Calls NewsAPI.org and Tavily Search API for curated headline ingestion.
  + Azure AI Gemini model is prompted to:
    - Extract top topics from text corpus.
    - Cluster these into consumer themes (e.g., "zero-sugar drinks", "adaptogenic herbs").
* **Output:** JSON clusters with metadata (theme label, confidence, supporting headlines).

**Phase 2: SocialSentimentAgent (Consumer Insight Extraction)**

* **Task:** Analyze social conversations to detect emotional drivers and objections.
* **Tools:**
  + Uses **PRAW (Python Reddit API Wrapper)** to scrape posts/comments from r/beverages, r/nutrition, and r/CPG.
  + Azure AI Gemini model scores:
    - Sentiment (positive, neutral, negative)
    - Themes (taste, packaging, health benefits, nostalgia, etc.)
    - Keywords and pain points
* **Output:** Summary cards of key consumer sentiments and topic-specific insights.

**Phase 3: ConceptGenAgent (Product Ideation)**

* **Task:** Convert trend and sentiment clusters into product concepts.
* **Tools:**
  + Azure AI Gemini is prompted with:
    - Top 3 trends
    - Corresponding sentiment clusters
  + Model outputs 3–5 innovative product pitches:
    - Flavor profile
    - Packaging idea
    - Tagline / market positioning
* **Output:** JSON list of product concept cards.

**Phase 4: RoadmapPlannerAgent (Launch Planning)**

* **Task:** Generate a high-level roadmap for taking selected concepts to market.
* **Tools:**
  + Prompt includes selected concept and planning parameters (timeline range, resource constraints).
  + Azure AI Gemini generates:
    - Milestones (e.g., R&D, regulatory, packaging, launch)
    - Tentative timelines
    - Resource notes
* **Output:** Project roadmap JSON with Gantt-like structure and planning tips.

**Phase 5: Agent Orchestration & Scheduling**

* **LangGraph** connects all agents into a conditional pipeline.
* Agents are triggered either:
  + **On-demand** via API (e.g., /insights/trends)
  + **Scheduled** via Azure Scheduler and Azure Functions (e.g., daily 8 AM run)
* Failures, retries, and low-confidence fallbacks are handled in the graph logic.

**Phase 6: UI Integration & Observability**

* FastAPI endpoints exposed for:
  + /insights/trends, /insights/sentiment, /concepts/generate, /roadmap/plan
* **Frontend UI** (Azure App Service) displays:
  + Trend clusters, sentiment cards, concept mockups, and roadmap timelines
* Instrumented with **Azure Monitor** and **Azure Log Analytics** for:
  + API latency
  + Token usage
  + Error counts
  + Success rate of generation flows

# Expected Deliverables

**Code Assets**

* LangGraph workflow for orchestrating all four agents.
* Python-based FastAPI server exposing four REST endpoints.
* Prompt templates for each LLM call, abstracted for reuse.
* Dockerfile and deployment scripts (multi-stage builds)

**Azure Configuration**

* Azure Functions deployment configurations for scheduled triggers.
* Azure App Service application for frontend UI.
* Azure Scheduler jobs for automated agent execution.
* Azure Container Registry container configurations.

**Frontend & Demo UI**

* Azure App Service application featuring:
  + Trend clustering UI with tag clouds or keyword tiles.
  + Sentiment summary view (charts + commentary).
  + Concept gallery with flavor, packaging, and tagline.
  + Interactive roadmap viewer with Gantt-style layout.

**Observability Stack**

* Azure Monitor dashboard:
  + Latency, success rate, error distribution, fallback usage.
* Azure Monitor alerts:
  + Triggered on high error rates, low-confidence responses, or endpoint failures..

**Documentation**

* Setup Guide:
  + Azure configuration, API key management, deployment scripts.
* Agent Customization:
  + Instructions for modifying prompt logic and swapping data sources.
* Runbook:
  + Monitoring, scaling, and common failure modes.
* Google API Discovery Service for each endpoint (/insights/trends, etc.)