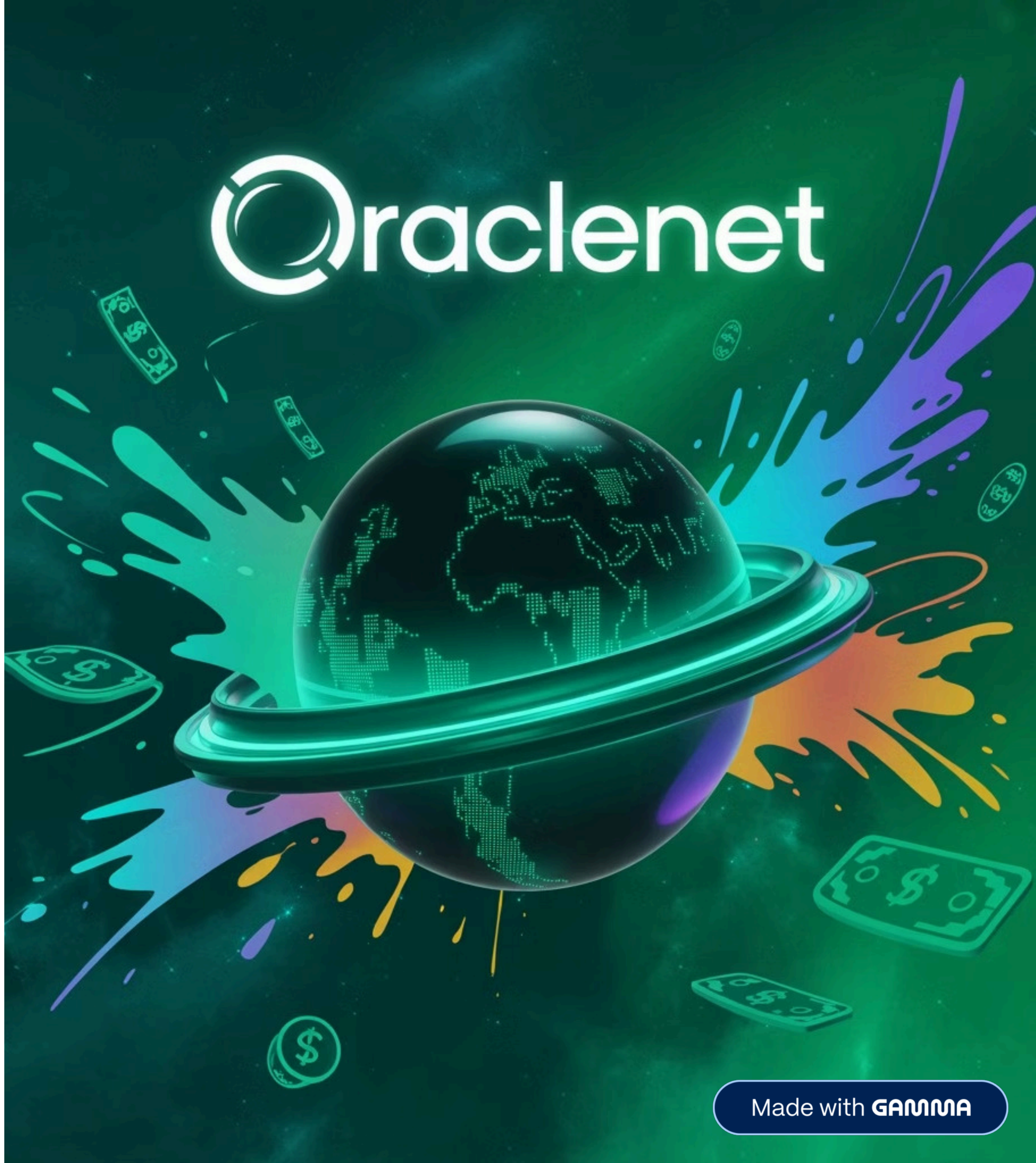
OracleNet - Dynamic Pricing Assistant

Agentic AI for Real-Time Price Optimization

By P Sai Srujan Reddy and Swayam Srujan Tripathi



Presentation Agenda

1. Problem Statement
2. Solution Overview
3. Functional Requirements
4. High Level Design (HLD)
5. Low Level Design (LLD)
6. Technical Implementation
7. Non-Functional Requirements
8. AI-Driven Pricing Strategies
9. Benefits & Impact
10. Future Enhancement
11. Demo & Questions

Problem Statement

Current E-commerce Pricing Challenges

Manual Monitoring

Time-consuming competitor analysis.

Static Models

Inability to respond to market changes.

Data Fragmentation

Scattered competitor pricing information.

Reactive Strategies

Adjustments after market shifts.

Dynamic Pricing Assistant Solution

Competitor Scraper

Automated data collection from Flipkart.

Price Optimizer

AI-driven pricing strategy recommendations.

Impact Simulator

Predictive analysis of pricing decisions.

Vector Database

Scalable data storage with Pinecone.

Business Use Cases:

E-commerce Optimization

Dynamic price adjustment for electronics.

Market Competitiveness

Strategic undercutting or price matching.

Profit Maximization

Balance customer influx with profit margins.

Tech Stack

Frontend

Streamlit UI

Backend

Python with LangChain

LLM

Ollama with Deepseek R1 model

Database

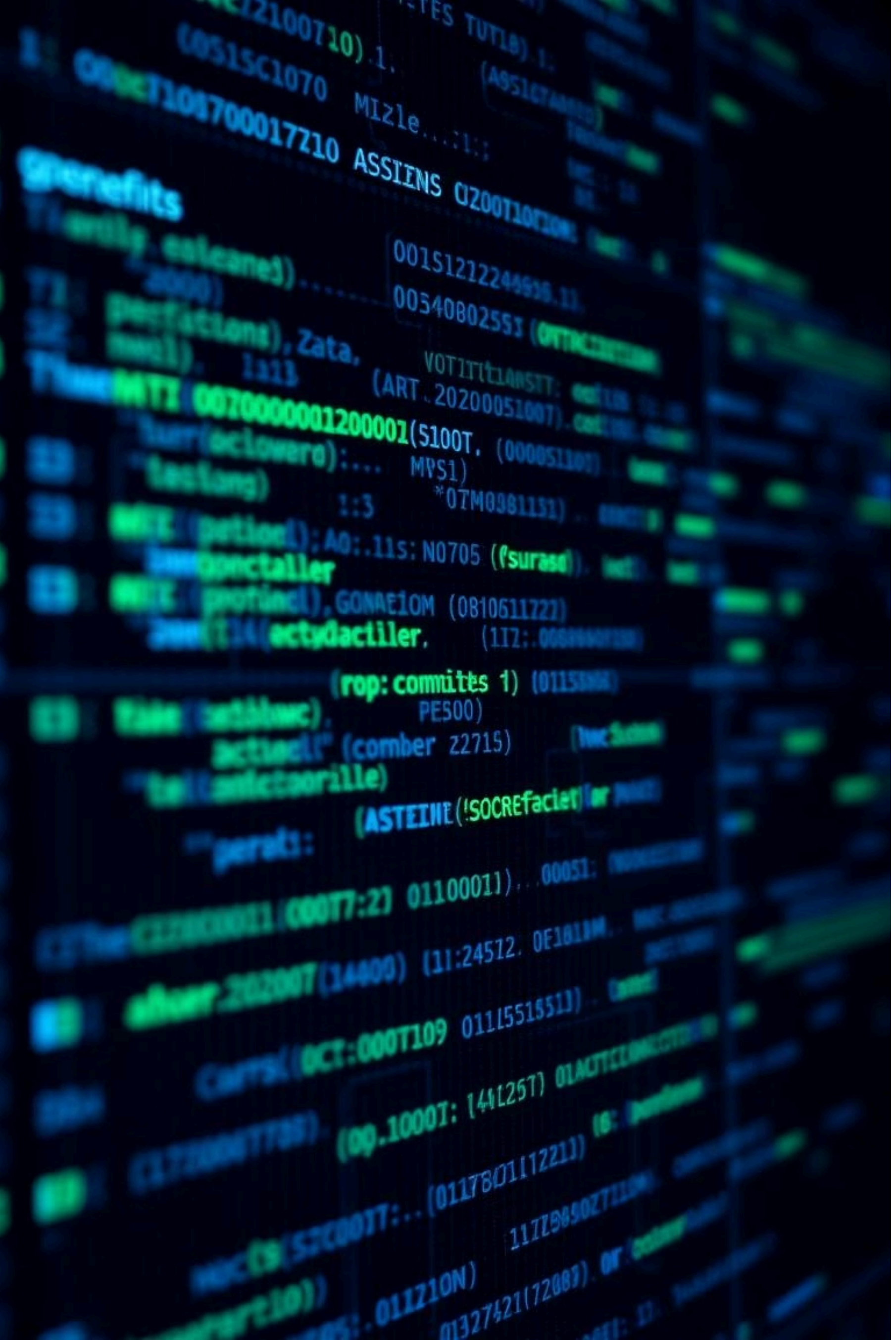
Pinecone Vector DB

Web Scraping

BeautifulSoup + Requests

Data Processing

Pandas, NumPy



Functional Requirements: (Core System Capabilities)

Web Scraping

Extracts Amazon product prices (e.g., Samsung Galaxy S22 Ultra, ₹24,990).

Data Management

Pinecone vector database storage with local JSON backup.

LLM Integration

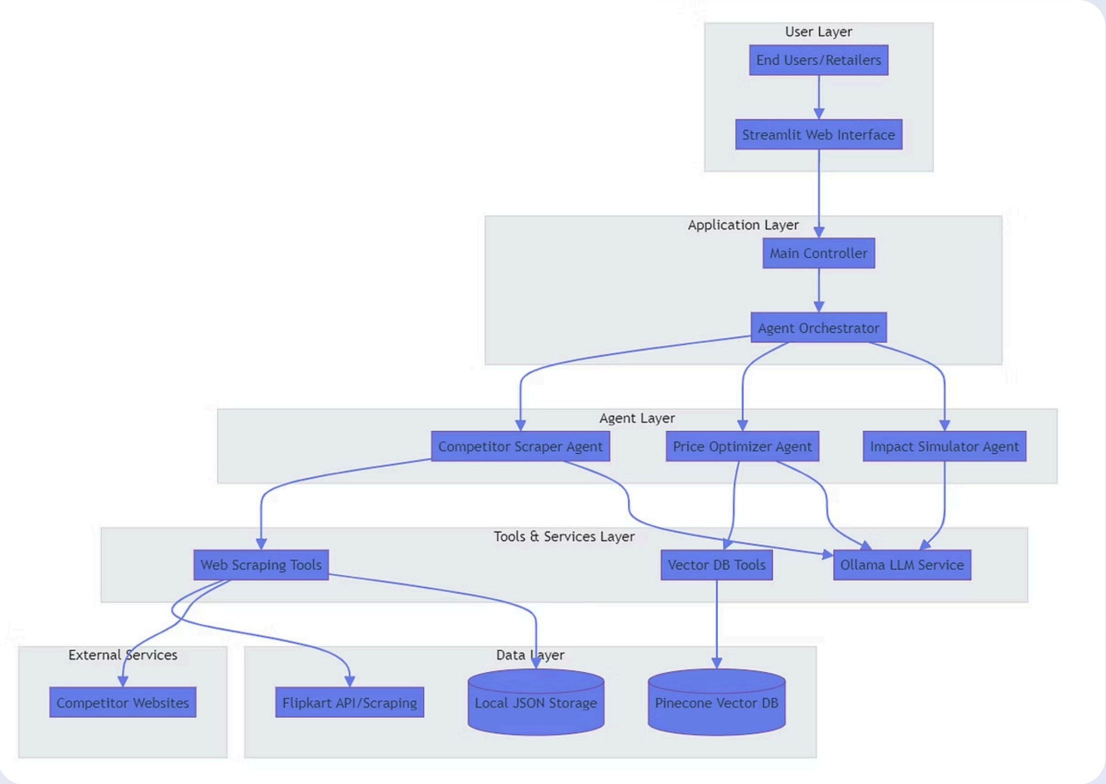
Ollama with Deepseek R1 for agent reasoning and error handling.

User Interface

Streamlit-based input/output with product specification fields.

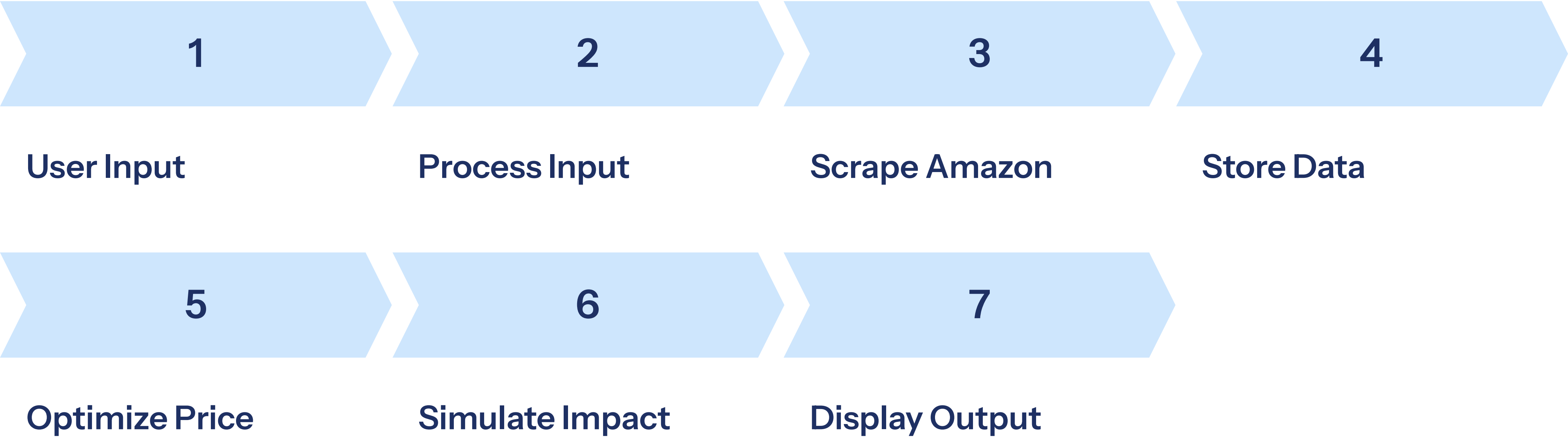
High Level Design

This diagram illustrates the key components and data flow of our dynamic pricing assistant system.



System Architecture Overview

Multi-Agent Architecture Flow:



1 User Input

2 Process Input

3 Scrape Amazon

4 Store Data

5 Optimize Price

6 Simulate Impact

7 Display Output

Three Specialized AI Agents

CompetitorScraperAgent

Role: Web scraping and data collection

Tools: BeautifulSoup, web_scraper, store_in_pinecone

Output: Structured pricing data from Flipkart

PriceOptimizerAgent

Role: Strategic pricing analysis

Tools: query_pinecone for competitor data, pricing algorithms

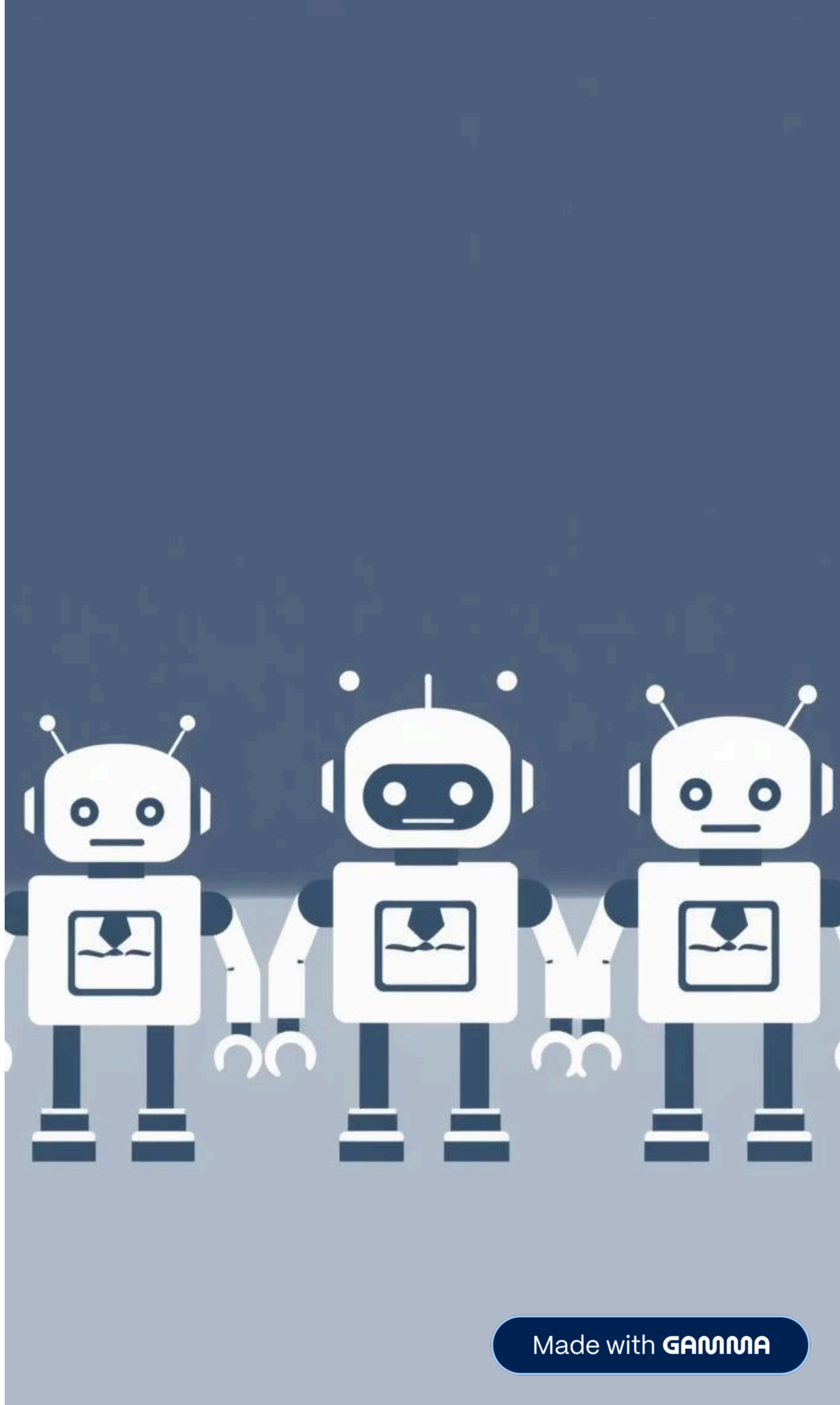
Output: Pricing recommendations to maximize profit and competitiveness

ImpactSimulatorAgent

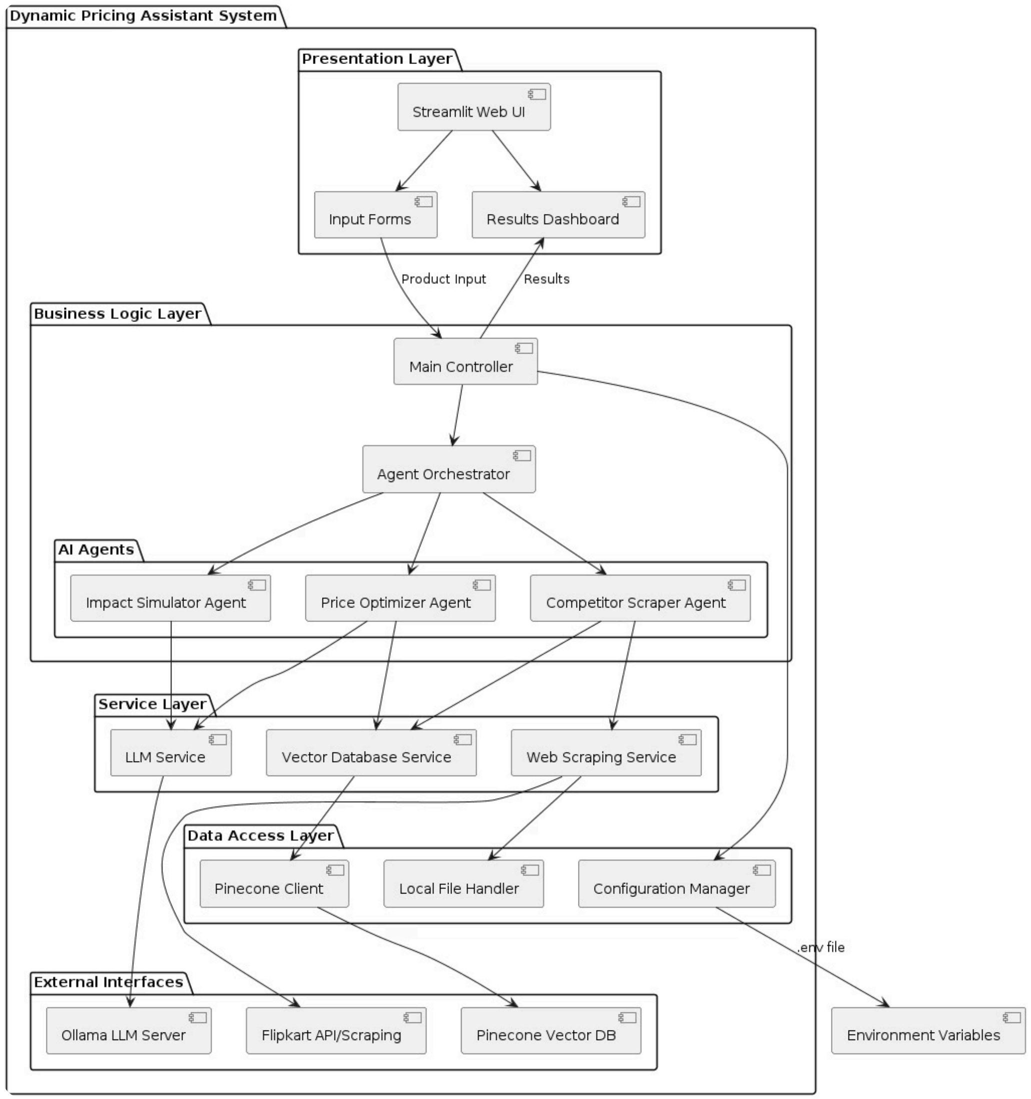
Role: Predicts customer influx and profit margins

Tools: predictive modeling, scenario simulation

Output: Analysis reports on pricing impact and sales forecasts



Low-Level Design: Component Architecture



Presentation Layer

- **Streamlit Web UI:** Main user interface for input/output
  - **Input Forms:** Collects product details and specifications
  - **Results Dashboard:** Displays pricing recommendations and analysis
- **Data Flow:** Product Input → Results display back to user

Business Logic Layer

- **Main Controller:** Central orchestrator managing all system operations
- **Agent Orchestrator:** Coordinates the three specialized AI agents
  - **Impact Simulator Agent:** Predicts market impact and profit margins
  - **Price Optimizer Agent:** Generates strategic pricing recommendations
  - **Competitor Scraper Agent:** Handles web scraping and data collection

Service Layer (Core Processing Services)

- **LLM Service:** Integrates with Ollama for AI-powered decision making
- **Vector Database Service:** Manages Pinecone operations for similarity search
- **Web Scraping Service:** Handles data extraction from e-commerce sites

Data Access Layer (Data Management)

- **Pinecone Client:** Direct interface to vector database
- **Local File Handler:** Manages JSON backup storage
- **Configuration Manager:** Handles environment variables and settings

External Interfaces (Third-party Integrations)

- **Ollama LLM Server:** Local language model for agent reasoning
- **Flipkart API/Scraping:** Source for competitor pricing data
- **Pinecone Vector DB:** Cloud-based vector storage and retrieval
- **Environment Variables:** Secure configuration management

Technical Implementation

Code Structure & Components

```
├── agents/       # AI agent implementations
├── tools/        # Web scraping and database tools
├── config/       # Environment and settings
├── models/       # Data model definitions
├── utils/        # Helper functions
├── main.py       # Main application entry point
└── requirements.txt   # Dependencies
```

Key Technologies

- LangChain for agent orchestration
- Pinecone for vector storage
- Ollama for local LLM inference

Data Flow & Storage

Information Architecture

Input Data:

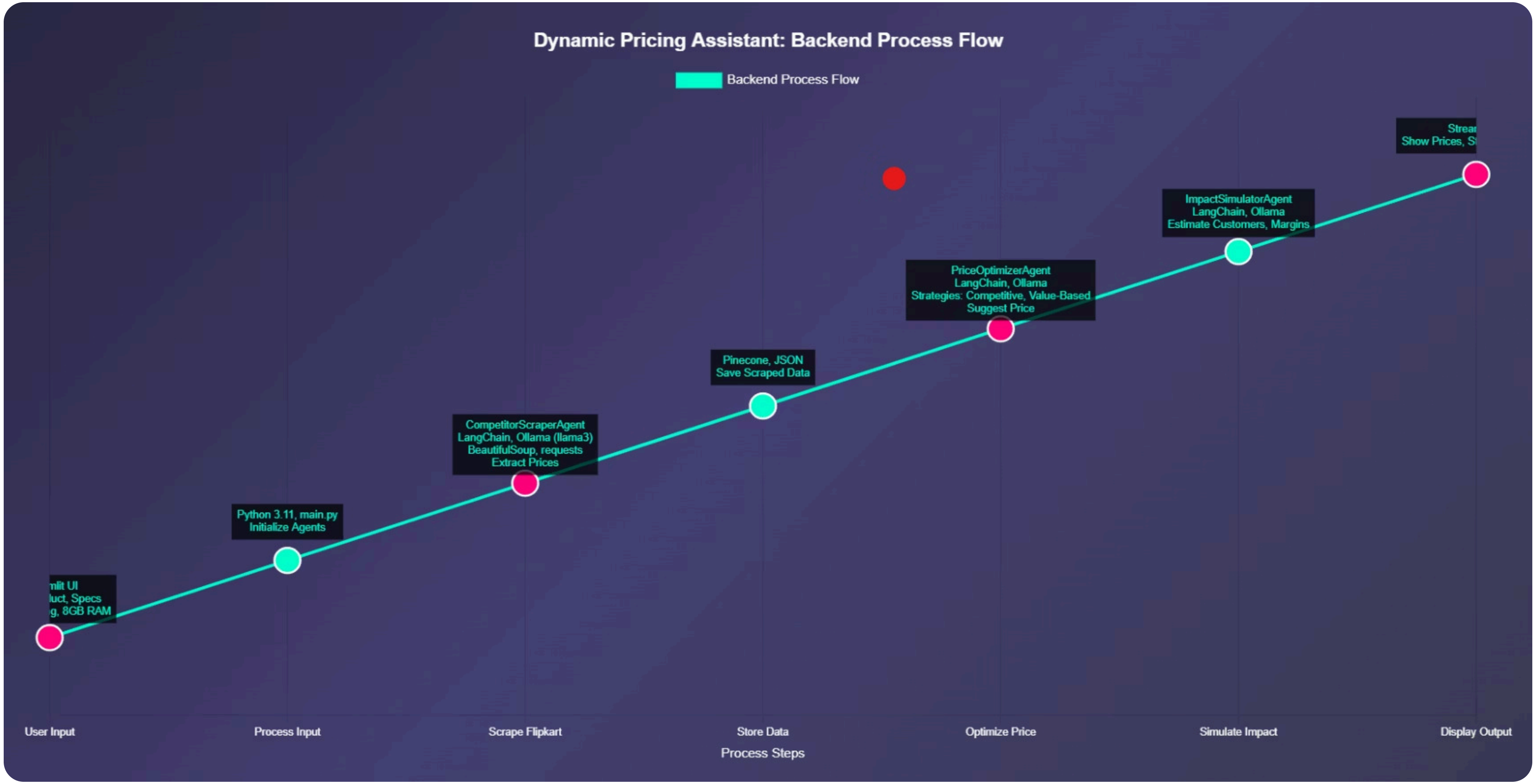
- Product name (e.g., "Smartphone")
- Specifications (e.g., "8GB RAM, 128GB storage, 5G")

Scraped Data Structure:

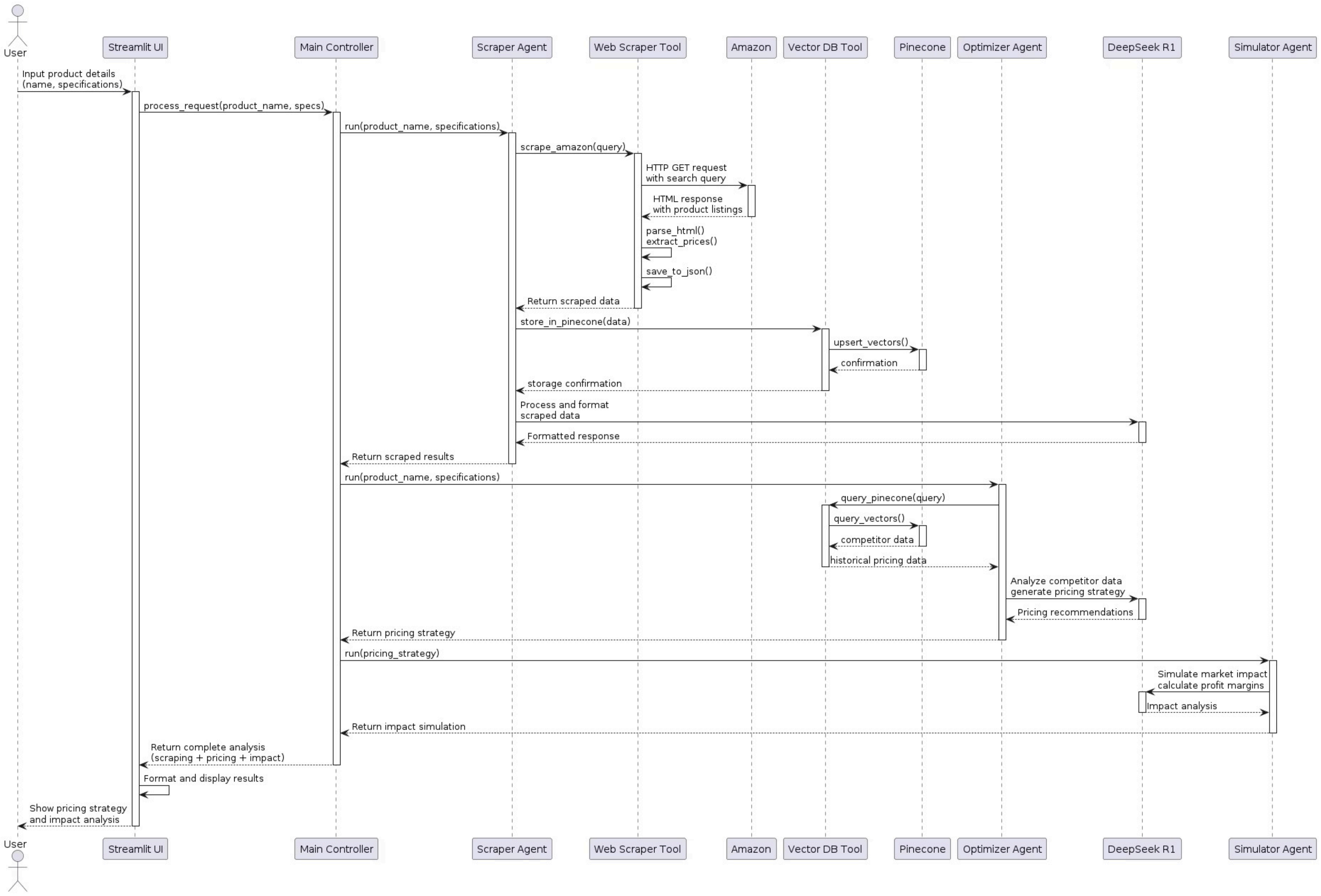
```
json{
  "source": {"source": "Amazon", "url": "..."},
  "product": {
    "name": "Samsung Galaxy S22",
    "specs": {"price": "₹24,990"},
    "category": {"name": "Smartphone"}
  }
}
```

Vector Database: Pinecone embeddings for similarity search

Backend Process Flow



Low-Level Design: Sequence Flow & Interactions



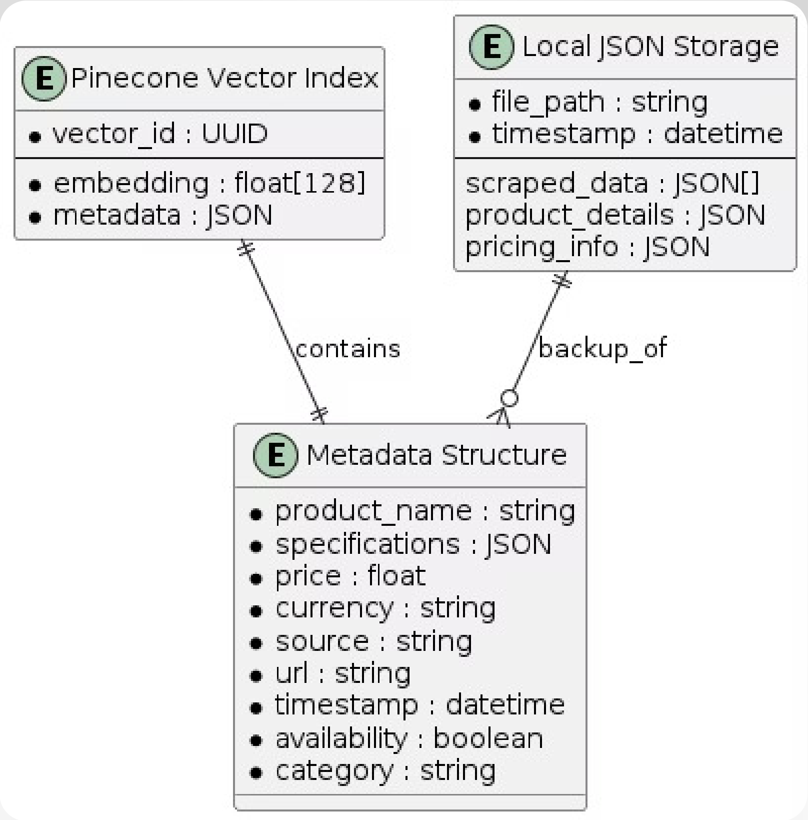
Backend Database Schema

Data Relationships

- Products ↔️ Categories: Many-to-One relationship
- Products ↔️ Competitor Data: One-to-Many relationship
- Products ↔️ Pricing Strategies: One-to-Many relationship
- Pricing Strategies ↔️ Impact

Integration Points

- Pinecone Vector DB: Stores embeddings, metadata in PostgreSQL
- Local JSON Storage: Backup for scraped data and analysis results
- Agent State Management: Tracked through agent_logs table



E Pinecone Vector Index

- vector_id : UUID

- embedding : float[128]
- metadata : JSON

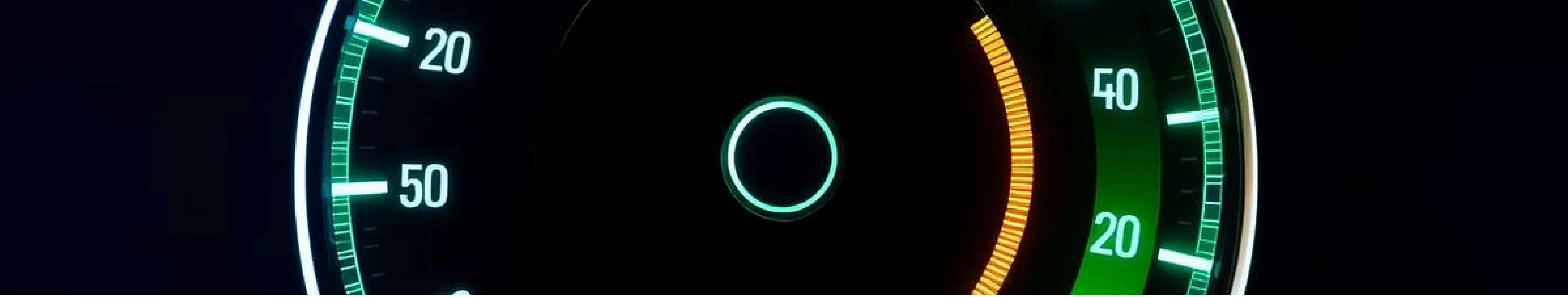
E Local JSON Storage

- file_path : string
- timestamp : datetime

scraped_data : JSON[]
product_details : JSON
pricing_info : JSON

E Metadata Structure

- product_name : string
- specifications : JSON
- price : float
- currency : string
- source : string
- url : string
- timestamp : datetime
- availability : boolean
- category : string



Non Functional Requirements: (Performance & Quality Standards)

Requirement	Target	Implementation
Performance	<5 seconds	Optimized workflows
Scalability	100 concurrent queries	Pinecone DB
Reliability	99.9% uptime	Robust error handling
Security	API key protection	.env configuration

AI-Driven Pricing Strategies

Competitive Pricing

5% below average competitor price.

Premium Positioning

10% below highest market price.

Value-Based Pricing

Feature-driven price optimization.

Penetration Pricing

5% below lowest competitor price.

Benefits & Impact

Business Value Proposition

For Retailers:

- Automated Monitoring: Eliminate manual price tracking
- Real-time Adaptation: Respond to market changes instantly
- Profit Optimization: Balance competition with margins
- Scalable Solution: Handle multiple product categories

For Customers:

- Competitive Prices: Benefit from optimized pricing
- Market Awareness: Transparent pricing strategies

ROI Metrics:

- Reduced manual effort by 80%
- Improved price competitiveness by 15%
- Enhanced profit margins through strategic positioning

Future Enhancements

Roadmap & Scalability

Phase 2 Developments:

- Multi-platform scraping (Amazon, Best Buy, Newegg)
- Advanced ML models for demand forecasting
- Real-time price alerts and notifications
- Integration with e-commerce platforms

Scalability Improvements:

- Microservices architecture
- Cloud deployment (AWS/Azure)
- Advanced caching mechanisms
- API rate limiting optimization

Demo & Questions

Live Demonstration

Demo Flow:

- Input product: "Smartphone with 8GB RAM"
- Show scraping results from Amazon
- Display pricing optimization recommendations
- Present impact simulation results

Key Metrics to Highlight:

- Response time < 5 seconds
- Accurate competitor price extraction
- Strategic pricing recommendations
- Business impact predictions

Q&A Session

Thank You

Contact & Resources

Project Repository: GitHub Link

Documentation: Technical specifications and setup guide in github.

Key Takeaways:

- Multi-agent AI system for dynamic pricing
- Real-time competitor analysis
- Strategic business impact optimization
- Scalable and maintainable architecture