Multi-Agent Sports Intelligence System

Developed a production-grade agent framework using Google ADK for real-time sports knowledge retrieval and validation

Developed By: **Tausif Khanooni**

Project Objectives & Scope

Multi-Agent Design

Production-grade system built with Google Agent Development Kit for intelligent orchestration

Knowledge Management

RAG integration for deep retrieval from internal sports manuals and documentation

Real-Time Intelligence

Tavily Web Search implementation for live sports updates and current events

Production Ready

Custom observability and deployment to Google Cloud Platform Cloud Run

Multi-Agent Intelligence Architecture

A sophisticated **Tri-Agent Architecture** ensures accuracy and boundary control through specialized roles:

Orchestrator Agent

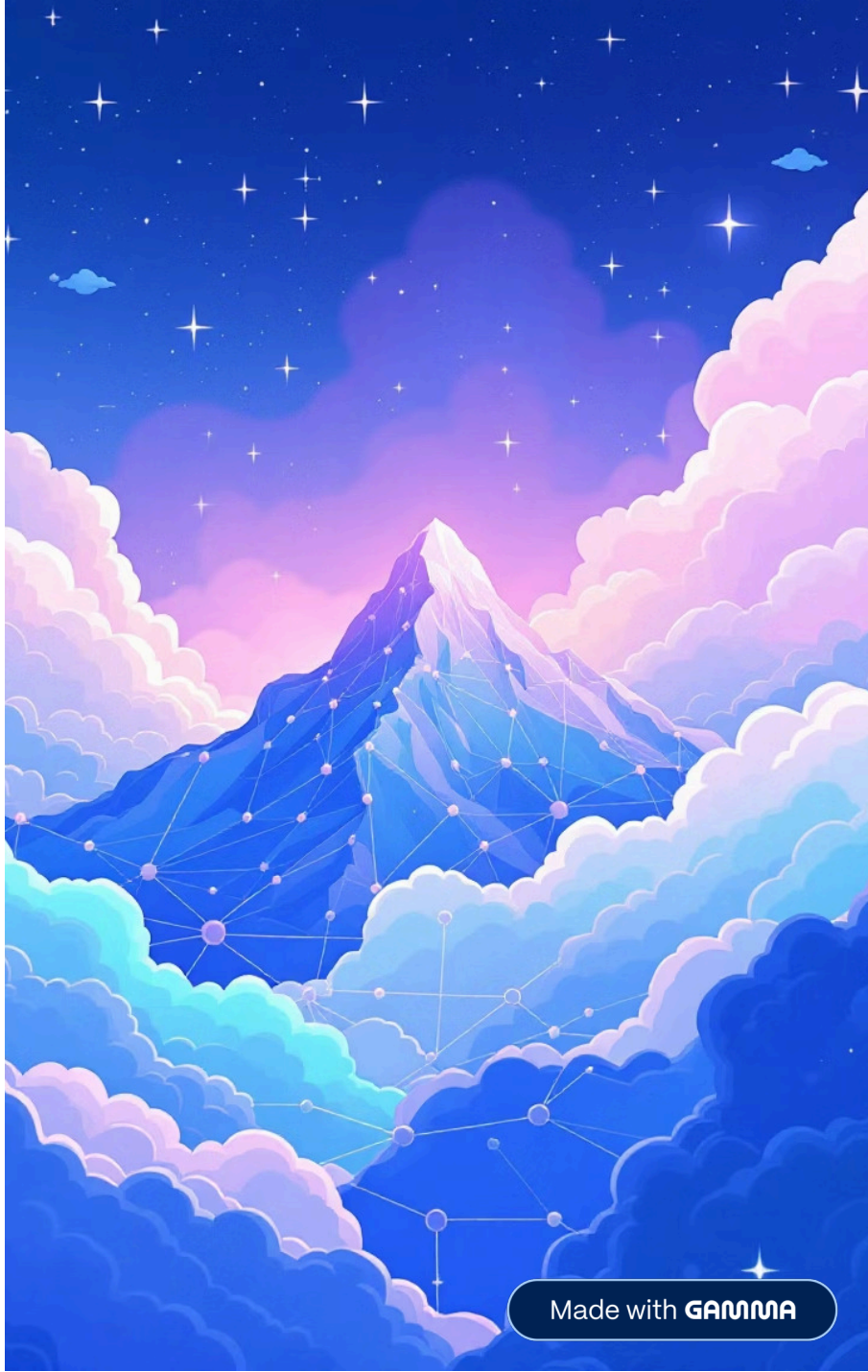
The "Brain" that routes queries to appropriate agents and manages session memory

Researcher Agent

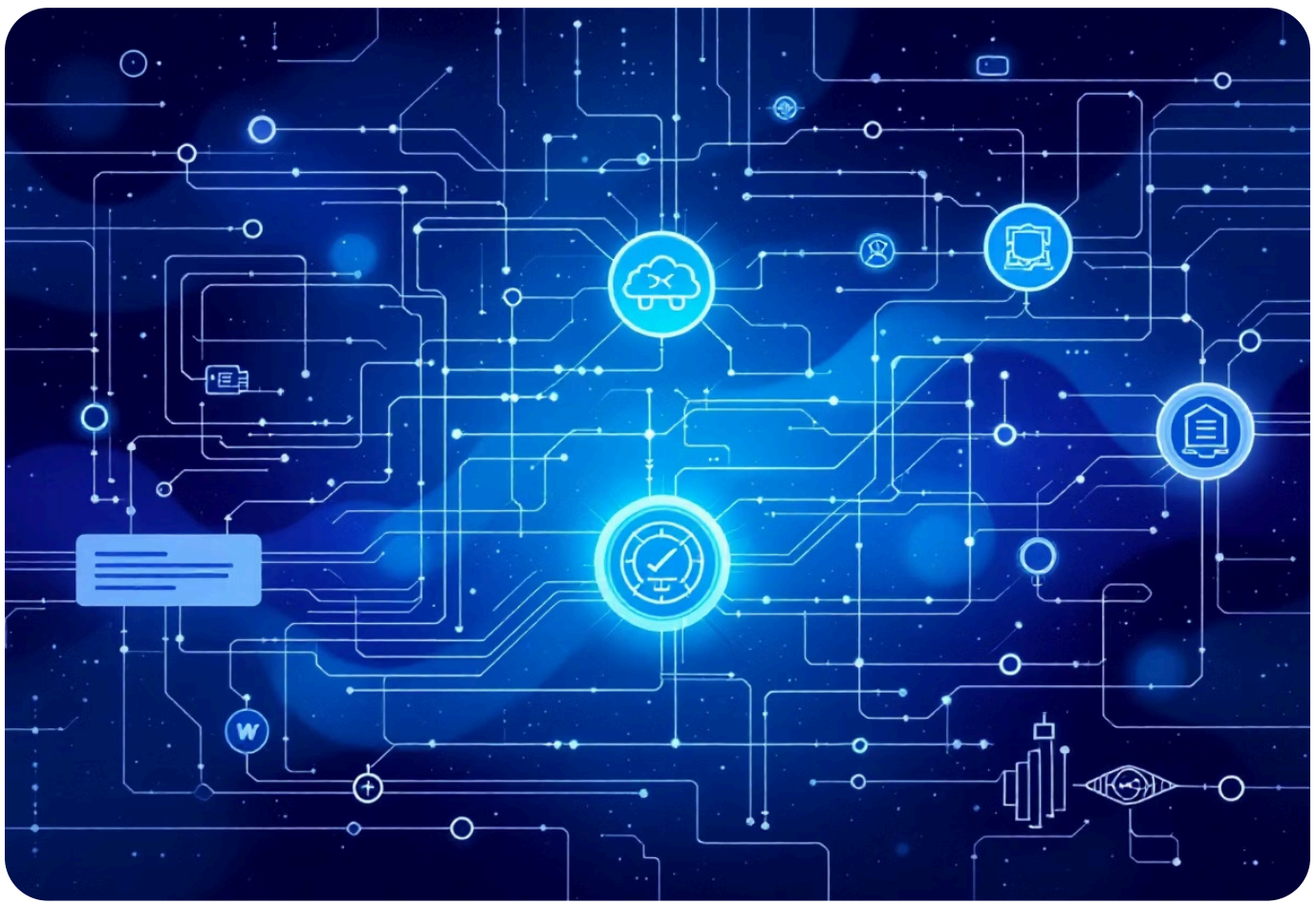
The "Specialist" that intelligently decides between RAG and web search based on query intent

Reviewer Agent

The "Judge" that validates responses for quality and ensures source citations



Communication Patterns & State Management



Sequential Flow

High-reliability handoff from **Orchestrator → Researcher → Reviewer** ensures quality control at each stage.

Feedback Loop

The Reviewer can reject low-quality Researcher outputs, triggering automatic re-analysis and refinement.

State Management

- Flask-level session handling for concurrent users
- Shared context objects for collaborative agent building
- Persistent memory across multi-turn conversations

RAG Support & Knowledge Pipeline

High-performance retrieval pipeline for unstructured sports data:

01

PDF Extraction

PyPDF parsing of Tennis and Cricket manuals

02

Semantic Chunking

Intelligent splitting to preserve context and meaning

03

Embedding Generation

Vertex AI text-embedding-004 for semantic search

04

Vector Storage

FAISS index for billion-scale similarity search

Web Search Capabilities

Tavily API Integration

Intelligent Switching

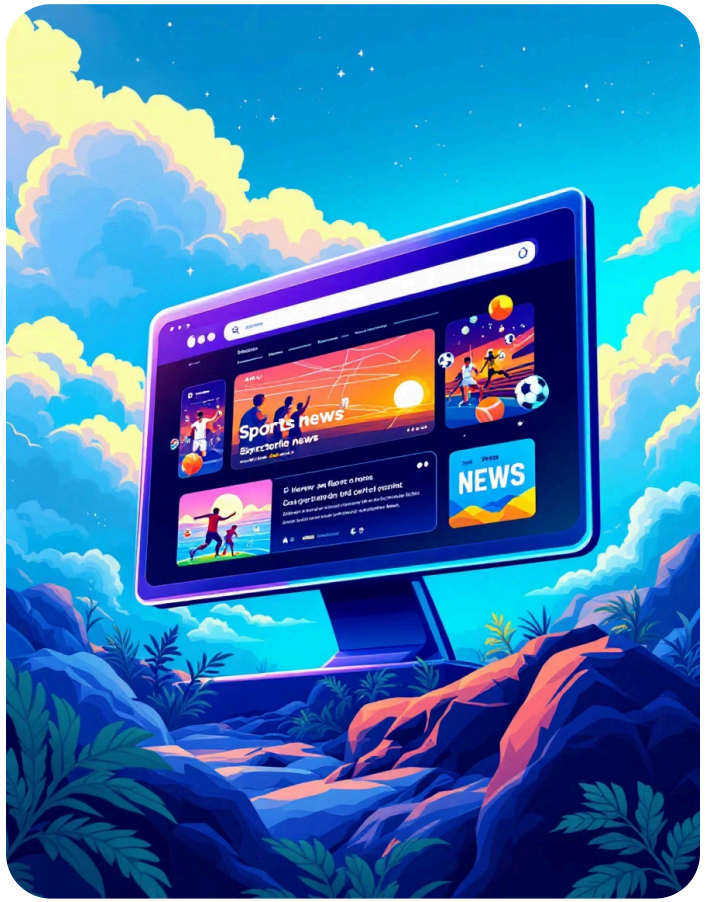
Researcher evaluates query intent and activates Tavily Tool for out-of-scope topics like Soccer World Cup

Agent-Optimized Results

Cleaned, high-relevance snippets designed specifically for LLM consumption

Source Citation

Every search result includes original URL for Reviewer Agent citation



Custom Tools & Agent Composition

System built on **FunctionTool** composition for flexible, maintainable architecture:

query_tennis_cricket_rag_tool

Custom logic to search FAISS index and return metadata with source references

web_sports_search_tool

Python requests implementation to integrate with Tavily Search API

Decoupled Architecture

Tools are decoupled from agents, enabling flexible sharing and role-based restriction

Observability & Monitoring

Production-Grade Monitoring

- **Cloud Run Logging**

  Integrated logs for monitoring agent execution times and token usage

- **Custom Decorators**

  Centralized logging middleware in src/observability/ to track every agent thought and action

- **Error Handling**

  Resilient loops that capture tool failures and provide helpful fallbacks

Deployment to Google Cloud Platform

1

Containerization

High-efficiency Docker image using python:3.10-slim for optimal resource usage

2

Backend Service

Deployed as sport-expert-chatbot on Google Cloud Run for auto-scaling

3

UI Layer

Separate static-serving container ensures high availability and fast response times

4

Secure Configuration

Environment variables for TAVILY_API_KEY and GOOGLE_CLOUD_PROJECT management

Project Structure & Results

Verified Capabilities

Tennis/Cricket Retrieval

Successfully answers questions from local PDF manuals

Soccer & General Sports

Retrieves current information via Tavily web search

Boundary Enforcement

Respects scope by refusing non-sports queries

Production Quality

Context-aware, source-cited, and ready for deployment

Directory Structure

```
MultiAgentAssignment/
├── src/
│   ├── main.py
│   ├── agents/
│   ├── tools/
│   ├── rag/
│   ├── observability/
│   └── ui/
├── knowledge_base/
├── faiss_index/
├── Dockerfile
└── requirements.txt
```

Conclusion & Demo

Verified Tennis/Cricket Retrieval

Successfully answers questions from local PDF manuals.

Verified Soccer & General Sports

Successfully retrieves current information via Tavily web search.

Verified Boundary Enforcement

Respects scope by refusing non-sports queries.

Production Quality

Context-aware, source-cited, and ready for deployment.