LLM-Powered Astrologer Recommendation Engine

— By Devansh Singh

This document outlines a scalable, privacy-conscious architecture for deploying a Large Language Model-powered astrologer recommendation system at Vedaz, optimized for 50,000 monthly active users.

I. LLM Stack Recommendation and Justification

Selecting the optimal LLM stack is paramount for Vedaz, balancing stringent data privacy, deep customization, and scalable cost-efficiency. Our recommendation is a fine-tuned open-source LLM (Mistral 7B or LLaMA 3 8B) deployed via Hugging Face Inference Endpoints.

- **OpenAI (Proprietary APIs):** Offers state-of-the-art performance and rapid prototyping. However, it presents significant data privacy concerns due to default data retention policies, limited model control, and a pay-per-token cost model that escalates with high usage.
- **Hugging Face (Managed Open-Source):** Provides a secure, managed service for open-source models. It is SOC2 Type 2 certified and GDPR compliant, with explicit 30-day log retention policies. This option balances control with operational simplicity, offering autoscaling and "scale-to-zero" features for cost optimization.
- **Open-Source Models (LLaMA, Mistral):** Deliver unparalleled data sovereignty and full fine-tuning capabilities, crucial for domain-specific applications. While requiring higher technical expertise and upfront investment, they become more cost-effective for high-volume usage.

Recommendation Justification: The hybrid approach of fine-tuned open-source models on Hugging Face Inference Endpoints is ideal. It ensures complete control over sensitive user data, allows for deep customization essential for precise astrologer recommendations, and offers long-term cost efficiency for a production system serving 50,000 monthly active users.

Feature	OpenAI (Proprietary)	Hugging Face (Managed Open-Source)
Data Privacy	Default 30-day retention/ZDR option	30-day log retention/private endpoints
Customization	Limited API Fine-tuning	LoRA/QLoRA
Cost Model	Pay-per-token	Hourly/Pay-as-you-go
Setup Time	Minutes	Hours

II. Hosting and Scaling Strategy

Robust hosting and dynamic scaling are critical for delivering high performance and managing costs for 50,000 monthly active users. Our strategy leverages cloud-native solutions with optimized inference techniques.

- **Deployment Setup:**
  - **Containerization:** Models will be containerized using Docker for portability and consistent environments.
  - **API Serving:** FastAPI will expose the LLM inference as a high-performance RESTful API.
  - **Orchestration:** Kubernetes (e.g., AWS EKS, Azure AKS, GCP GKE) will manage container deployment, resource allocation, and scaling.
  - **Optimized Inference:** Integration with vLLM and Triton Inference Server will maximize throughput and minimize latency through techniques like continuous batching and PagedAttention.
- **Cloud Provider Options:**
  - Major providers like **AWS, Azure, or Google Cloud Platform (GCP)** offer the necessary GPU instances (e.g., NVIDIA A10G, L4).
  - Leverage multi-model endpoints (e.g., AWS SageMaker Inference Components) to share GPU resources and reduce idle costs.
- Architecture Diagram (1-line):
  User → API Gateway → Load Balancer → Kubernetes Cluster (LLM Inference Pods) → Vector DB/Data Store
- **Scaling Mechanisms:**
  - **Autoscaling:** Horizontal Pod Autoscalers (HPAs) will dynamically adjust the number of LLM pods based on real-time metrics like GPU utilization and queue size.
  - **Continuous Batching:** vLLM's continuous batching will process requests as they arrive, maximizing GPU utilization and improving throughput for varying input/output lengths.
  - **Cold Start Optimization:** Strategies like streaming model weights directly to GPU memory will minimize latency during scale-up events.

III. Monthly Cost Estimation for 50,000 Monthly Active Users

Cost-efficiency is paramount; our estimates for 50,000 MAU highlight significant differences across deployment strategies. These figures are based on conservative usage assumptions.

- **Assumptions for Usage and Token Counts:**
  - **Monthly Active Users (MAU):** 50,000
  - **Average Daily Interactions per User:** 1 interaction/day (1.5M interactions/month)
  - **Average Tokens per Interaction:** 500 input + 100 output = 600 total tokens
  - **Total Monthly Tokens:** 900 Million (900,000,000) tokens

LLM Stack/Hosting Strategy	Core Cost Driver	Estimated Monthly Cost

OpenAI GPT-4o Mini (API)	Token-based pricing	~$2,700.00
Hugging Face Inference Endpoints (Mistral 7B)	GPU-hours (managed service)	~$8,760.00 (before scale-to-zero optimization)
Self-Hosted AWS EC2 (LLaMA 3 8B)	GPU-hours + Operational Overhead	~$7,300 - $8,000+

Note: These costs are highly sensitive to actual usage patterns and token consumption. Implementing token optimization techniques (e.g., concise prompts, context trimming) is crucial for sustainable cost management.

IV. Privacy and Safety Concerns

Ensuring robust privacy and safety is non-negotiable for building user trust and ethical AI deployment, especially when handling sensitive user data.

- **Data Privacy & PII Handling:**
    - **Concern:** Exposure of Personally Identifiable Information (PII) from user chat history and profiles to third-party services or inadvertent model memorization. Non-compliance with data protection laws like India's DPDP Act.
    - **Mitigation:** Implement robust PII masking and anonymization techniques (e.g., regex, Named Entity Recognition - NER) to prevent sensitive data from leaving Vedaz's control. Ensure explicit user consent and lawful purpose for all data processing, adhering strictly to DPDP Act requirements.
- **Bias & Fairness:**
    - **Concern:** LLMs, trained on vast internet data, can learn and propagate societal biases, leading to unfair or discriminatory astrologer recommendations.
    - **Mitigation:** Employ unbiased prompting strategies and ensure training data is representative of diverse user demographics. Continuously monitor model outputs for bias and implement fairness metrics with human-in-the-loop review processes.
- **Hallucinations:**
    - **Concern:** LLMs may generate factually incorrect, nonsensical, or fabricated astrologer recommendations or descriptions.
    - **Mitigation:** Utilize Retrieval Augmented Generation (RAG) architectures to ground LLM responses in Vedaz's trusted astrologer database, significantly reducing factual errors. Implement fact-checking mechanisms and output filtering guardrails.
- **Prompt Injection & Security:**
    - **Concern:** Malicious user inputs (prompt injection) can manipulate LLM behavior, leading to unintended actions or data leakage.
    - **Mitigation:** Enforce robust input validation and sanitize user queries. Implement secure API key management and encrypt all data at rest and in transit. Conduct regular abuse monitoring and red-teaming exercises to identify and patch vulnerabilities.