CASE STUDY

GenAI-Powered Destination Assistants for DMOs

athenaworks

Index

Client Background

A travel-tech platform that partners with destination marketing organizations (DMOs); such as cities, regions, or tourism boards; to enhance visitor engagement. Unlike online travel agencies that focus on bookings, it’s core value lies in delivering personalized and revenue-aligned destination guidance through AI.

Our client supports destinations like San Francisco, Atlanta, South Dakota, Puerto Rico, Mexico Beach, Las Vegas, and more; scaling a tailored experience across over 80 clients.

Project Overview

Team Makeup

1 GenAI Engineer

Type of Service

Managed Project

Tech Stack

- Cloud computing: AWS (Kubernertes / EC2)
- Orchestration: Github actions
- Vector Store: Custom PostgreSQL with PGvector extension
- Pipeline Control: DVC
- Language Model: GPT-4o Mini (fallback: GPT-4 for reasoning)
- Front-End Config: Internal “Model Builder” tool for Sales/Product
- Scraping Infra: Bypass tools for Cloudflare-protected sites
- Evaluation: LLM scoring + internal QA processes

Challenges

The client faced multiple technical and operational challenges:

- **Content overload:** DMO websites often contain thousands of URLs and resources (one client site alone has roughly 8,000), overwhelming visitors looking for timely, relevant information.

- **Generic recommendations:** Open models like ChatGPT may recommend unrelated places — for example, suggesting Seattle instead of San Francisco, or sending a traveler an hour outside the city they actually asked about.

- **Revenue leakage:** Without alignment to business goals, GenAI answers could steer users away from sponsored listings or partners — the restaurants and experiences that fund the platform.

- **Scalability and customization:** Serving dozens of clients meant adapting the same GenAI core to local content, tone, and user expectations — San Francisco's model has nothing in common with Atlanta's, yet both had to run on one shared architecture.

- **Latency and deployment bottlenecks:** Early iterations took up to 15 days to deploy a new model, limiting how quickly the team could onboard a new destination.

- **Content freshness:** Expired events, promotions, and other stale content needed to be filtered out automatically, both when a model is built and at inference time.

Solution

Athenaworks built a multi-tenant GenAI concierge solution using the following components:

1. Retrieval-Augmented Generation (RAG) with Custom Vector Stores

- Custom PostgreSQL-based vector databases power retrieval for every destination model.
- Data is scraped exclusively from public content on each DMO's own website — no proprietary or first-party client data is used. Clients can include or exclude specific pages, languages, or URLs (for example, a city can flag a route or district it doesn't want the assistant to reference).
- The pipeline runs on Kubernetes on AWS, with Dockerized jobs and DVC (Data Version Control) managing model artifacts too large for standard version control.

2. Agentic Architecture & Event Awareness

- Heuristic and LLM-based filtering excludes expired events (festivals, marathons, promotions) during both model construction and inference.
- Agent chains (chain-of-thought reasoning) handle time-sensitive queries and route requests dynamically.
- The architecture is built for future API integration with real-time event data feeds.

3. Dynamic Language Support

- Assistants respond in whatever language the visitor uses — Mandarin, Spanish, Thai, or otherwise — with no per-language configuration required.

4. Business-Aware Response Prioritization

- Built-in override logic lets a city prioritize the answers that matter to its business — for example, routing every relevant query toward a curated list of ten sponsored, Michelin-starred restaurants instead of whatever an open model would suggest — even when the raw query points elsewhere.
- The same override layer lets clients tune response length and tone per audience: one client asked for shorter, punchier answers, which the team exposed as a configurable “specificity” parameter rather than a one-off code change.

5. Model-Building Platform (Self-Serve Interface)

- A no-code model-configuration tool (“Model Builder”) lets the Sales/Product team spin up a new city model from a template — no engineering ticket required, saving weeks of development per client.
- The same interface exposes per-model parameters (tone, specificity, link overrides) so business changes don't require a code deploy.

6. Invisible Compliance & Page Scraping

- Most tourism sites sit behind CDN anti-bot protections (Cloudflare and similar); Athenaworks built internal tooling that bypasses these blockers without any technical lift from the client, reducing friction and speeding up go-live.

7. Automated Evaluation Framework

- Every model generates its own held-out test set as it's built: the system writes representative questions, retrieves the RAG context, and produces answers automatically — no manual test-writing per city.
- Each answer is scored on two axes using an LLM-as-judge process: a context-relevance score and a hallucination score, giving the team an objective signal before anything reaches production.
- The team treats this as a floor, not a finish line: an automated eval can mark an answer “correct” even when it doesn't serve the client's business intent (e.g., routing to the objectively best link instead of a sponsored one), so every model still goes through a manual QA pass tuned to what that specific client actually wants.

“We didn't just spin up a RAG in an afternoon and call it done. The first thing we do for every model, before it ever gets near production, is put an evaluation system in front of it.”

Results

Metric	Result
Time to Deploy	Cut from 15 days → 2 days
Platform Scale	Grew from 9 to 80+ cities
Customizability	All models personalized without retraining
Multilingual Support	Fully dynamic, no setup required
Model Ownership	Handed off to internal Sales/Product teams via no-code builder
Cost Efficiency	Runs on GPT-4o Mini for most use cases, resorting to GPT-4 only for reasoning-heavy inference (e.g. event filters), saving 10x in LLM costs

Most GenAI implementations struggle to scale or reach ROI. Tiki's deployment stands out because it's in production and financially sustainable; an outlier in a space where 80% of GenAI projects stall or burn budget without return.
Tiki was engineered for scale and flexibility, combining smart use of heuristics, low-cost models, and reusability. It puts customization and control in the hands of product and sales teams, removing engineering bottlenecks.
This is not just another chatbot; it's an enterprise-grade, configurable AI concierge designed for destination marketing at scale.

Contact Us

athenaworks.com

athenaworks