Case Studies

AI Engineering Case Studies

How Perry Systems Inc builds production-grade AI agents, retrieval-augmented generation (RAG) systems, and knowledge graphs that drive measurable outcomes for performance marketing, lead generation, regulated industries, and AI-driven content automation.

AI search now handles **22% of all queries** in 2026, Google AI Overviews appear in **99.9% of informational searches**, and AI-referred visitors convert at **14.2%** versus **2.8%** for traditional organic — roughly **5x more valuable per visit**. The companies that win the next decade are the ones shipping real AI systems, not AI marketing decks. The case studies below are a partial list of what we have shipped to production.

What We Build

Capability	What It Means in Practice	Where It Shows Up
AI Agents	Autonomous, tool-using systems built on Anthropic Claude, OpenAI GPT-4o, and Anthropic Managed Agents — orchestrated with LangGraph, custom agent chains, and the Model Context Protocol (MCP).	Social Media Consultant, GEO Audit Agent, Google Ads Auditor, Smart Strategy Generator
Retrieval-Augmented Generation (RAG)	Hybrid semantic and keyword search over private document corpora, with citation-grade provenance on every response.	Healthcare Policy Knowledge Graph, AI Newsletter Engine
Knowledge Graphs	Neo4j-modeled entity-relationship graphs that answer multi-hop questions vector search alone cannot reach — fused with vector retrieval for content depth.	Healthcare Policy Knowledge Graph
Agent Chains and Agent Graphs	LangGraph-orchestrated multi-agent workflows with stateful checkpoints, parallel sub-agents, and self-correcting retrieval loops.	Healthcare compliance dual-agent system, social-media intelligence pipeline
Video Search and Analysis	FFmpeg scene detection through the Rendi FFmpeg API, OpenAI Whisper transcription, and frontier-model visual analysis — with searchable content summaries indexed for SQL queries.	Social Media Consultant Agent

Capability	What It Means in Practice	Where It Shows Up
Agentic Web Scraping	Apify-driven, terms-of-service-aware data acquisition across LinkedIn, Instagram, Google Ads Transparency, Meta Ad Library, and SEC EDGAR — with caching layers that cut external API spend significantly.	Lead Generation Pipeline, Smart Strategy Generator, Social Media Consultant
Frontier-Model Integration	Production deployments using OpenAI, Anthropic Claude, and Perplexity — routed by task to balance capability and cost.	Every case study below
Model Context Protocol (MCP)	We design, build, and operate MCP servers — exposing AI capabilities to Claude Desktop, internal client platforms, and any MCP-compliant surface without per-feature redeploys.	Internal MCP gateway infrastructure

---

AI Agents in Production

Social Media Consultant Agent

A full-stack Instagram intelligence agent built for a performance marketing client in the **Advertising & Marketing** industry. The agent scrapes, processes, and analyzes competitor and brand accounts at scale, surfacing the kind of pattern-level insights account teams used to take days to assemble manually. Dedicated Apify actors handle posts, reels, comments, follower counts, and profile metadata through a single orchestration layer; media is routed automatically through a parallelized video and image processing pipeline.

For video posts, the agent calls the **Rendi FFmpeg API** for scene detection and frame extraction, pulls audio for **OpenAI Whisper** transcription, and runs frame-by-frame visual analysis with a frontier vision model for content review, pacing assessment, and engagement prediction. A cost-efficient summarization model then generates a short content summary indexed for SQL search, so every video is discoverable by what is actually inside it — not just its caption. Multi-layer caching (account freshness, video processing, AI analysis) cuts repeated runs to a fraction of fresh-processing time.

The agent persists everything to a Databricks-backed data warehouse with automated daily pipelines that keep tracked accounts current with zero manual intervention. The result is an agent that can answer "show me every showroom-shot reel from these dealerships in the last 30 days, ranked by engagement velocity" in a single chat turn.

Stack: Apify (multiple dedicated Instagram actors), Rendi FFmpeg API, OpenAI Whisper, Anthropic Claude (vision and summarization), Databricks (Delta Lake, Unity Catalog Volumes), Next.js, Postgres, signed-URL media proxy.

Google Ads Auditor

A productized AI agent that delivers a complete Google Ads diagnostic — account structure, performance trends, optimization-score breakdown, conversion-tracking integrity, budget pacing, and a remediation roadmap — as an inline PDF and Excel deliverable in roughly **67 seconds end-to-end**. Built for an **Advertising & Marketing** client, the agent is available standalone through their web app and chat-accessible through our MCP gateway, so account managers can fire an audit from inside Claude Desktop or the client's internal AI surface without leaving the conversation.

The agent ingests Google Ads API data, runs a series of structural and policy checks against the client's proprietary auditing rubric, and uses Claude to synthesize findings into prioritized client-facing language. Authentication uses a signed service-session pattern that lets external MCP clients trigger audits without browser-based OAuth flows. Result: prospects get a real, branded audit deliverable in their inbox during the discovery call — not a week later.

Stack: Google Ads API, Anthropic Claude, Python, FastAPI, PDF and Excel rendering pipeline, signed service-session auth, MCP-accessible.

Generative Engine Optimization (GEO) Audit Agent

A production Anthropic Managed Agent that assesses a website's readiness to be cited by AI-powered search engines — ChatGPT, Perplexity, Google AI Overviews, Gemini, and Claude — across the signals that actually drive AI citation: brand-entity recognition, schema markup quality, robots.txt configuration for AI crawlers (OAI-SearchBot, PerplexityBot, Claude-SearchBot, GPTBot), Wikidata and Knowledge Graph presence, NAP consistency, content structure for AI extraction, and third-party citation footprint.

The agent runs an entity audit, a technical-readiness scan, and a baseline citation report measuring how often the brand surfaces across **50–100 target queries** on each AI platform. Output is a prioritized roadmap mapped to a four-pillar GEO framework. Brand search volume — the **single strongest predictor of AI citation at 0.334 correlation** — is benchmarked against the top three competitors. Built for an **Advertising & Marketing** client; exposed through MCP for chat-trigger access from any compliant surface.

Stack: Anthropic Managed Agents, Anthropic Claude, web search and crawling, schema validation, JSON-LD analysis, MCP integration.

Smart Strategy Generator

An AI-powered competitive intelligence engine that ingests a target company plus three competitors and produces a polished, branded strategy PDF in roughly **15 minutes** — work that previously consumed a full day of analyst time. The agent orchestrates a multi-stage data pipeline using web scraping for site structure and on-page messaging, **SerpAPI** for Google Ads Transparency Center data, **Apify** for Meta Ad Library creative, and **OCR** for image-based ad copy.

A frontier-model reasoning step reads the consolidated dataset against a curated industry-vertical strategy knowledge base to surface positioning gaps, channel mix, audience signals, and messaging opportunities. The interface guides users through competitor setup, runs the scan, and exports a publication-ready report. Live ad-spend signals (which network competitors favor, posting frequency, audience targeting clues) surface in the deliverable — context manual research consistently misses. Built for an **Advertising & Marketing** client.

Stack: Python, Streamlit, OpenAI GPT-4, Selenium, SerpAPI, Apify, Google Ads API, Tesseract OCR, ReportLab, Docker.

Workflow Automation for Google Ads

Ad Disapproval Monitor

A daily Google Ads policy-compliance pipeline running across **100+ client accounts** for an **Advertising & Marketing** client. Every morning the system pulls all disapproved ads through the Google Ads API, extracts the full violation context — campaign, ad group, ad copy, policy topics — and routes the structured payload through **OpenAI GPT-4o** with an ads-policy-tuned prompt. The model returns a violation summary, a detailed root-cause description, and an ordered list of corrective actions per ad.

Each finding becomes a Jira ticket on the client's project board: auto-labeled by violation type, due-dated one business day out, populated with a formatted inventory table, and assigned to the responsible PM. Account managers work remediation, not data gathering. The end-to-end loop runs daily without intervention and has eliminated the manual triage cycle that previously consumed several hours of PM time per week.

Stack: Python, FastAPI, Google Ads API, OpenAI GPT-4o, Jira REST API, LangChain, scheduled cron.

Performance Alerting System

A Google Ads optimization-score watcher that turns a lagging KPI into a leading trigger for intervention. The system queries the Google Ads API for optimization scores at the campaign

level across every active client account, then computes a **cost-weighted account average** that mirrors Google's own internal methodology more closely than naive averaging — a multi-million-dollar campaign at a depressed score impacts portfolio health differently than a small-budget one at the same score, and the math reflects that.

When a cost-weighted score drops below a configurable threshold, the system creates a high-priority Jira ticket on the client's board with the score, a due date, and a clear call to action. Every run logs to a structured audit trail with customer ID, score, ticket reference, and run status — feeding operational dashboards that surface which accounts drift, which fixes hold, and which require escalation. Threshold and cadence are configurable per client tier. Built for an **Advertising & Marketing** client.

Stack: Python, Google Ads API, Jira REST API, structured audit logging, scheduled cron.

Knowledge Systems: RAG, Knowledge Graphs, and Agent Graphs

Healthcare Policy Knowledge Graph

A dual-agent retrieval system built for a **Healthcare** compliance technology client serving payors and policy teams. The platform answers complex regulatory questions across thousands of payor policy documents — questions where naive document chat fails because they require both **semantic understanding of policy text** and **structured reasoning across metadata** (disease, medical field, policy type, payor, line of business).

The architecture splits the work between two specialized agents orchestrated through **LangGraph** with stateful checkpoints:

- A **Graph Agent** models documents, diseases, medical fields, policy types, procedure codes, and payors as a **Neo4j knowledge graph** and answers metadata-driven questions through generated Cypher queries — for example, "which policies cover orthopedic procedures for Medicare Advantage and have changed in the last 90 days?"
- A **RAG Agent** runs hybrid semantic retrieval inside document content using a healthcare-tuned embedding stack. The agent retrieves top-K chunks, evaluates them iteratively, and stops only when grounded confidence is high.

Every response carries source citations back to the originating document — non-negotiable for medical-legal compliance. An automated ingestion pipeline keeps the graph and vector index fresh as new policy versions land. **FastAPI** serves the programmatic surface; **Streamlit** provides the end-user interface; WebSocket streaming delivers real-time answer generation.

Stack: LangGraph, OpenAI, Anthropic Claude, Neo4j, Chroma, MongoDB Atlas, healthcare-tuned embeddings, FastAPI, Streamlit, Postgres.

AI Newsletter Intelligence Engine

An AI-driven editorial automation platform built for a **Financial Services** client whose weekly credit-risk newsletter previously consumed hours of manual research, sourcing, and writing per issue. The platform replaces the manual editorial cycle with a configurable AI pipeline.

Editors define companies and risk factors of interest — distress signals, revenue declines, supply-chain disruption, regulatory action, bankruptcy indicators. **Perplexity API** handles news discovery and cited research across the open web. **OpenAI** synthesizes findings into structured outputs: priority-ranked distress signals, two-paragraph evidence chains, and short headlines that match the publication's voice. **SEC EDGAR** integration validates public-company references. A templated rendering layer produces Outlook-safe HTML email ready to send through any platform.

The system enforces editorial rigor through structured outputs and template-driven rendering, so the AI does not freelance tone or layout. Generation can run on-demand through the web interface or on a fixed cadence.

Stack: Python, FastAPI, React, Streamlit, Perplexity API, OpenAI, structured-output validation, SEC EDGAR API, Docker.

AI-Driven Lead Generation

LinkedIn Demand-Signal Pipeline

A daily AI-driven lead-generation pipeline built in-house at Perry Systems to support our shift from hourly development services to AI-driven outcomes tied to client ROI. The system pulls LinkedIn job postings every morning through **Apify** using cookieless public-page scraping — a deliberately terms-of-service-conservative posture, in contrast to vendors that rotate burner accounts against private LinkedIn APIs and carry the resulting service-continuity and brand-fit risk.

Postings are filtered against a tight ICP (vertical, headcount band, role pattern) and routed by keyword tightness against three service offerings: AI Knowledge Cores, Workflow Automation Agents, and Decision Systems. Empirical validation across **400 real postings** showed tight-keyword searches yielded approximately **21% ICP-qualifying** rates, while generic terms pulled mostly hyperscalers at roughly 4%. Volume modeling: 5–10 tight-keyword URLs running daily produces 50–150 qualifying leads per day.

Each qualified lead is enriched with **Exa.ai** for the job poster's LinkedIn profile. **Anthropic Claude** with prompt caching generates a personalized connection-request note tailored to the prospect's company size, vertical, and matching service offering. Output is an Expandi-ready CSV with a human-review gate. All-in monthly cost: roughly **$150** including all third-party services.

Stack: Apify (cookieless public-page scraping), Exa.ai neural search, Anthropic Claude with prompt caching, Expandi (outreach delivery), Python orchestration.

Productized Services

Generative Engine Optimization (GEO) Service

We productized a complete GEO service offering for an **Advertising & Marketing** client — three tiers (Audit + Quick Wins, Foundation Retainer, Full GEO Management) built on a four-pillar framework: **Brand Entity Building**, **Third-Party Presence**, **Original Data and Statistics**, and **YouTube and Long-Form Video**. Strategy is anchored in academic and industry research: the Princeton/Georgia Tech GEO paper (KDD 2024), SE Ranking's 2.3M-page citation study, OtterlyAI's 100M+ YouTube citation study, and Semrush's 248K-Reddit-post analysis.

The service tracks citation rate, Share of Voice, and AI-referred conversion rate across **ChatGPT, Perplexity, Google AI Overviews, Gemini, and Claude**. AI-referred visitors convert at **14.2%** versus **2.8%** for traditional Google organic — a difference reporting puts in front of every retainer client every month.

Most agencies selling "AI SEO" are doing rebranded SEO. This program builds across the full citation surface: owned-site optimization, Reddit and forum strategy, review-platform management, press and earned media, original data programs, and long-form video. **68% of ChatGPT citations come from third-party sources**, and brands are **6.5x more likely to be cited through third-party domains** than their own — which is why a website-only strategy fails on AI search.

Internal AI Infrastructure

Autonomous Research Pipeline

Perry Systems' internal AI engineering knowledge base grows itself. We run **four scheduled remote agents** every weekday morning, each researching a slice of the AI landscape: Anthropic platform updates, OpenAI product releases, AI industry news, and trending GitHub

repositories. Each agent reads the prior week of archived digests for deduplication, runs web research, writes a source-cited digest, and auto-merges a pull request into the knowledge-base repository.

A local processor then consolidates daily digests into weekly rollups, surgically integrates new findings into existing reference notes (append-not-replace, so prior thinking is preserved), and archives processed material. A weekly script generates a one-page executive summary of the week's developments. The pipeline runs against the existing AI subscription, so incremental cost is minimal.

The result: our AI engineering team — and our client-facing recommendations — stay current without anyone reading papers manually. It is also a working reference architecture for clients who want autonomous research, monitoring, or knowledge-management systems of their own.

Stack: Scheduled remote AI agents, local orchestration layer, git-versioned knowledge base, optional local LLM for triage.

Model Context Protocol (MCP) Engineering

We design, build, and operate **MCP servers** — the Anthropic-led open standard for connecting AI clients to tools and data. Our internal MCP gateway exposes Anthropic Managed Agents, the Google Ads Auditor, the GEO Audit Agent, and other capabilities to Claude Desktop, internal client platforms, and any MCP-compliant surface through a single transport. Async trigger/poll patterns match long-running AI workflows. The result: every new AI capability ships in days, not sprint cycles, without ever redeploying the consumer application.

Technologies We Work With

Category	Tools and Models
Frontier LLMs	OpenAI (GPT-4o family), Anthropic Claude (Opus, Sonnet, Haiku), Perplexity API for cited web research
Embeddings	OpenAI text-embedding family, healthcare- and domain-tuned open-source embedding models
Vector Databases	Chroma, MongoDB Atlas Vector Search, Pinecone, pgvector
Knowledge Graphs	Neo4j, Cypher query generation, GraphRAG patterns
Agent Frameworks	LangGraph, LangChain, custom agent chains, Anthropic Managed Agents, FastMCP
Speech and Video	OpenAI Whisper, Rendi FFmpeg API, FFmpeg scene detection, OCR pipelines

Category	Tools and Models
Web and Data Acquisition	Apify, Selenium, Playwright, BeautifulSoup, SerpAPI, Exa.ai, SEC EDGAR
Marketing and Ads APIs	Google Ads API, Meta Ad Library, Facebook Marketing API, Google Analytics 4, Google Search Console
Data and Compute	Databricks (Delta Lake, Unity Catalog Volumes, Jobs API), Postgres, MongoDB, SQL Server, Redis
Application Stack	Next.js, React, FastAPI, Streamlit, Node.js, Python, TypeScript
DevOps and Hosting	Vercel, Docker, AWS, Azure, GitHub Actions
Project and Delivery	Jira REST API, Expandi, MCP for cross-tool orchestration

---

Frequently Asked Questions

What is an AI agent and how is it different from a chatbot?

An AI agent is an autonomous, tool-using system that can plan multi-step work, call external APIs, and act on outcomes — not just generate text. Our GEO Audit Agent crawls a website, validates schema, runs entity recognition, and produces a roadmap. A chatbot just answers questions. We build agents on Anthropic Managed Agents, LangGraph, and the Model Context Protocol.

What is retrieval-augmented generation (RAG) and when do you use it?

Retrieval-augmented generation (RAG) lets a language model answer questions using a private document corpus while citing sources. We use RAG when answers must be grounded in a client's own documents — payor policies, internal SOPs, regulatory filings, product documentation — rather than the model's training data. Our healthcare compliance platform combines hybrid semantic search with healthcare-tuned embeddings and citation tracking on every response.

When do you choose a knowledge graph over a vector database?

We use a knowledge graph when answers depend on **relationships between entities**, not just text similarity. Vector search alone cannot answer "which suppliers for critical components have quality issues in the last 18 months and also serve our top-three customers" — that is a multi-hop graph query. We model entities and relationships in **Neo4j** and fuse Cypher queries with vector retrieval to get the best of both.

How does Perry Systems approach video search and analysis?

We extract semantically meaningful frames using FFmpeg scene detection through the **Rendi FFmpeg API**, transcribe audio with **OpenAI Whisper**, run frame-by-frame visual analysis with a frontier vision model, and generate searchable content summaries with a cost-efficient summarization model. The result: every processed video is searchable by its visual content, not just its caption.

Which AI models does Perry Systems use in production?

We use all top frontier models, routed by task: **OpenAI** for general-purpose reasoning and speech-to-text; **Anthropic Claude** (Opus, Sonnet, Haiku) for structured reasoning, agent orchestration, and cost-efficient summarization; and **Perplexity API** for cited web research. Model routing is a meaningful share of the unit economics — using a small model for content summaries and a frontier model only for the work that needs it cuts cost without cutting capability.

What is agentic web scraping and how is it different from traditional scraping?

Agentic web scraping uses AI agents to plan, execute, and recover from data-acquisition tasks across the public web — adapting to layout changes, rotating dedicated actors per task, and respecting platform terms of service. We use **Apify** with dedicated actors for Instagram, LinkedIn, Google Ads Transparency, Meta Ad Library, and SEC EDGAR — each chosen for legitimacy and reliability rather than running burner-account workflows that get nuked when platforms tighten detection.

Do you build Model Context Protocol (MCP) servers?

Yes. We design, build, and operate MCP servers in production. Our internal MCP gateway is a Python server (FastMCP plus the Anthropic SDK) that exposes Anthropic Managed Agents, productized auditing tools, and other AI capabilities to Claude Desktop, client applications, and any MCP-compliant surface through a single transport. Async trigger/poll patterns match long-running AI workflows.

Can Perry Systems help my company build internal AI tooling?

Yes. The autonomous research pipeline we run internally — scheduled remote agents plus a local processor that consolidates and integrates findings into a knowledge base — is a working reference architecture we adapt for client use cases: competitive intelligence monitoring, regulatory change tracking, market-signal aggregation, and internal documentation Q&A.

How does Perry Systems measure AI work?

Every shipped AI initiative gets a documented business case before it starts and an impact retro after it ships. We measure cost per run, hallucination rate against a golden eval set, time saved versus the manual baseline, and where applicable, revenue or pipeline attribution. We do not ship AI without evals.

Work With Us

If your team is evaluating AI agents, RAG, knowledge graphs, video intelligence, agentic web scraping, or Generative Engine Optimization, we should talk. Perry Systems Inc combines deep AI engineering depth with the delivery discipline that gets these systems into production — and keeps them there.