GEO/AI Search Monitoring & Optimization Platform

Created by	Bibo Wang
Created time	@October 24, 2025 12:04 PM
Tags	

Cloud-Agnostic System Architecture (Canvas)

> Purpose: A vendor-neutral blueprint tying business capabilities to technical components. Replace terms like “object storage” or “stream bus” with your preferred tech later.

---

1) High-Level Architecture (Conceptual)

```
┌──────────────────────────────────────────────────────
──────────────────────────────┐
│                         Client Layer                        │
│  • Web App (Dashboards, Action Center)  • Admin Console  • Public API  │
└──────────────────────▲──────────────────────────────
──────────────────▲────────────┘
           │                                │
           │ GraphQL/REST                         │ Webhooks/SDKs
           │                                │
┌──────────────────────┴──────────────────────────────
────────────────┴──────────────┐
│                     Application Services                   │
│  Plan Service ──▶ Orchestrator ──▶ Queue ──▶ Collectors          │
│      │        │          │          │
│      │        └──▶ Schedules      └──▶ Raw Artifacts (answers)  │
│      │                           │   │
│      ▼                                  ▼    │
│  Parser/Normalizer ──▶ Brand Resolver ──▶ Scoring/Trends ──▶ Insights/
AC   │
│      │              │             │          │      │
│      ▼              ▼             ▼          ▼       │
│  GEO Audit Crawler ──▶ Issue DB ──▶ Recommender ──▶ LLM Emulator
─▶ Exper. │
│      │                                    │    │
│      └─────────────────▶ Alerts/Notifs ◀─────────────── Event Rules ◀
──────┘    │
└────────────────────────────────────────────────────────────────────
────────────────────────────────────┘
          │               │            │
          │               │            │
     ┌───────────┴───────────┐     ┌──────────┴─────────┐    ┌────────┴───
────────┐
     │  Data Layer   │        │ Observability │      │  Governance   │
     │ • Object Store│        │ • Metrics/Logs│      │ • Identity    │
     │ • Doc/Key DB  │        │ • Tracing     │      │ • Tenancy     │
     │ • Search Index│        │ • Dashboards  │      │ • Billing     │
     │ • Stream Bus  │        │          │      │ • Retention   │
     │ • Graph Store │        │          │      │ • Audit Logs  │
     └───────────────────────┘     └────────────────────┘    └───────────
─────────────┘
```

Legend: AC = Action Center; Exper. = Experiment Lab.

2) Component Catalog (Business ↔ Tech)

Client Layer

- **Web App (Dashboards & Workbench):** Explore visibility, evidence (answers + citations), competitor diffs; manage prompts, actions, experiments.
- **Admin Console:** Workspaces, roles, usage, plans, retention.

- **Public API/SDK:** Read scores/answers/audits; push prompts; receive webhooks.

Core Application Services

1. **Plan Service (Discover & Plan)**

   Business: Define topics, prompts, engines, locales, cadence, budget.

   Tech: REST/GraphQL for CRUD; validation; versioned plan snapshots.

2. **Orchestrator (Run Scheduling)**

   Business: Turns plans into executable jobs.

   Tech: Scheduler → Job generator → enqueues to **Queue** with back-pressure.

3. **Queue (Work Dispatch)**

   Business: Smooths spikes; isolates failures.

   Tech: Durable FIFO/standard messages, retry & dead-letter.

4. **Collectors (Acquisition)**

   Business: Ask AI engines; capture answers, citations, screenshots.

   Tech: Headless browser/HTTP clients, proxy rotation, human-like pacing; write **Raw Artifacts** to **Object Storage**.

5. **Parser & Normalizer**

   Business: Make results comparable across engines.

   Tech: Async workers transform HTML/JSON → `AnswerRecord` + `Citation[]`.

6. **Brand & Competitor Resolver**

   Business: Map mentions/links to brands & competitors.

   Tech: Alias dictionaries, fuzzy matching, ownership graph.

7. **Scoring & Trends**

   Business: KPIs: Answer Share, Citation Share, Prominence, Sentiment, Deltas.

   Tech: Batch jobs computing daily `ScoreSnapshot`; indexed in **Search**.

8. **Insights & Action Center**

Business: Prioritized fixes with expected impact (content outlines, schema tasks).

Tech: Heuristics + simple models; LLM assist for outlines; ticketing/webhook connectors.

9. **GEO Audit Crawler**

   Business: Site readiness check (FAQ/schema/metadata/speed/clarity).

   Tech: Rendered crawl; rules engine emits `AuditIssue` + fix recipes.

10. **LLM Emulator (Test Harness)**

    Business: Predict which variant is more cite-worthy before publishing.

    Tech: Deterministic LLM configs, judge prompts, reproducible seeds.

11. **Experiment Lab**

    Business: Validate lift with pre/post or A/B; publish proof.

    Tech: Experiment registry, windows, significance/uplift calc.

12. **Alerts & Notifications**

    Business: Digest + real-time alerts on drops/spikes & competitor surges.

    Tech: Rule engine on metric streams → email/Chat/webhooks.

Data Layer (Technology-Agnostic)

- **Object Storage:** Raw HTML/JSON/PNG artifacts; versioned; lifecycle policies.
- **Document/Key-Value DB:** Jobs, answers, citations, scores, audits, actions, experiments.
- **Search Index:** Fast filtering/aggregations (by engine/locale/topic/brand/date).
- **Stream Bus:** Event pipeline (ingest → parse → resolve → score → index).
- **Graph Store (optional):** Brand ↔ Domain ↔ Topic ↔ Citation relationships.
- **Cache:** Hot queries, pre-computed leaderboards.

Observability & Governance

- **Metrics/Logging/Tracing:** p95 latencies, run success, parse errors, cost/run.

- **Identity & Tenancy:** Workspaces, roles (Viewer/Editor/Admin), API tokens.
- **Billing/Usage:** Runs, seats, emulator hours; plan limits, overage.
- **Data Retention & Audit:** Configurable TTLs; immutable audit logs.

3) Key Data Contracts (neutral JSON)

QueryPlan

```
{
  "plan_id": "qp_2025_10_24",
  "workspace_id": "ws_...",
  "engines": ["chatgpt","perplexity","gemini"],
  "locales": ["en-US"],
  "prompt_ids": ["p1","p2"],
  "cadence": "daily",
  "budget_cap": {"runs_per_day": 800}
}
```

AnswerRecord

```
{
  "answer_id": "ans_...",
  "job_id": "job_...",
  "engine": "chatgpt",
  "prompt_id": "p1",
  "locale": "en-US",
  "answer_text": "markdown...",
  "answer_type": "summary",
  "citations": [
    {"url":"https://brand.com/x","domain":"brand.com","position":1,"first_party":true}
  ],
  "evidence": [{"type":"screenshot","uri":"obj://raw/.../page.png"}],
  "created_at": "2025-10-24T09:00:00Z"
}
```

ScoreSnapshot (daily)

```
{
  "score_id": "sc_...",
  "workspace_id": "ws_...",
  "brand_id": "br_...",
  "topic_id": "t_...",
  "engine": "chatgpt",
  "locale": "en-US",
  "window_start": "2025-10-23",
  "answer_share": 0.42,
  "citation_share": 0.35,
  "prominence": 0.61,
  "sentiment_avg": 0.18,
  "delta_answer_share": 0.07
}
```

AuditIssue

```
{
  "audit_id": "au_...",
  "brand_id": "br_...",
  "url": "https://brand.com/x",
  "issue_code": "FAQ_SCHEMA_MISSING",
  "severity": "high",
  "rationale": "FAQ block absent; competitors cited for Q&A queries",
  "fix_recipe": {"type":"faq_block","schema":"FAQPage","example":"..."},
  "impacted_prompts": ["p1","p7"],
  "status": "open"
}
```

4) Core Workflows (Step-by-Step, Plain Language)

A) Daily Monitoring Loop

1. **Plans** create jobs → **Orchestrator** schedules them.
2. **Collectors** ask AI engines; store raw pages & screenshots.
3. **Parser** extracts answer text + links; **Resolver** tags brands.
4. **Scoring** summarizes KPIs; **Index** makes them searchable.
5. **Dashboard** shows trends; **Alerts** flag anomalies.

B) GEO Audit → Action Center

1. **Crawler** scans selected site paths; runs checks.
2. Emits **AuditIssues** with severity + how-to fix.
3. **Recommender** ties issues to lost prompts & expected lift.
4. One-click create tasks or CMS drafts via **Integrations**.

C) LLM Emulator (Pre-Publish Test)

1. Choose a losing prompt + page; load Variant A/B.
2. **Emulator** asks a judge model which is more cite-worthy.
3. Store result; ship the predicted winner.

D) Experiment Lab (Post-Publish Validation)

1. Define window & pages; run pre/post or A/B.
2. Collect metrics; compute **uplift** with significance.
3. Publish report with evidence.

E) Communications

- **Digest:** weekly summary of wins/losses, actions completed.
- **Alerts:** immediate notifications for drops/surges.
- **Exports/API:** feed BI or CRM with scores & issues.

5) Non-Functional Requirements (Targets)

- **Reliability:** ≥99.9% dashboard availability; ≥97% successful runs/day.
- **Performance:** p95 query < 1.5 s; ingestion to dashboard < 60 min.
- **Cost Controls:** per-workspace budgets; back-pressure on queue.
- **Security:** org/workspace isolation; encrypted data at rest & transit; signed evidence URLs.
- **Privacy/Robots:** configurable modes (strict/standard/lab) with customer consent.

6) Deployment Topology (Technology-Neutral)

- **Stateless Services:** Orchestrator, Parsers, Scorers, Recommender, API.
- **Worker Pools:** Collectors, Crawler, Emulator jobs (container or function runtime).
- **Data Stores:** Object storage, document DB, search index, stream bus, optional graph store.
- **Gateways:** API (REST/GraphQL), Webhook receiver, Auth server.
- **Pipelines:** CI/CD with automated tests & canary releases.

7) Roadmap (Business + Build)

- **Phase 1 (MVP):** Plans, Collectors (2 engines), Parser, Dashboard v1, GEO Audit v1, Alerts.
- **Phase 2:** Brand Resolver, Scoring/Trends, Action Center, API/Exports, Dashboard v2.
- **Phase 3:** LLM Emulator, Experiment Lab, Graph Explorer.
- **Phase 4:** Automation rules, Billing/Tenancy, Compliance toolkit.

8) Glossary

- **Answer Share:** % prompts where your brand appears in AI answers.
- **Citation Share:** % of links in AI answers pointing to your domains.
- **Prominence:** Weighting of mention order/position in the answer.
- **GEO:** Generative Engine Optimization — optimizing to be cited by AI.

Notes

- Replace each generic component with your chosen technology (e.g., “object storage” → your blob store).
- This canvas is intentionally implementation-agnostic so it fits any cloud/on-prem stack.