Agent Architecture (Cyclic LangGraph, cost-aware Tavily use)

Goal

Return **Top-10 recommended events** per query with: **title, time, location, organizer, url, recommendation, rationale** (one sentence ≤350 chars, hidden/clickable).

Scope & Tech

- **LangGraph** orchestration with a **cyclic loop** (GPT-Newspaper style).
- **Tavily** for web search + extract (cost-aware).
- **LlamaIndex** for lightweight parsing/field normalization.
- UI in scope; cloud deploy out of scope.

Inputs, Defaults, Guardrails

- **Inputs (min):** free-text + date range + city (default city=NYC; date window=next 14 days).
- **Cache first:** key = hash(query+filters+model+version).
- **Budget:** soft cap **$0.10/query**. If projected cost exceeds cap → **block**, show estimate, let user explicitly re-run.

Loop strategy with Tavily (how the cycle stays cheap)

- **One baseline `/search`** (depth=`basic`, `include_domains` from profile, date filter=14 days, `max_results`≈15–20). Operate the **cycle on this candidate set**.

- Inside the loop, **re-rank/filter locally**; call **`/extract` only for the shortlisted URLs** (e.g., top 8–12) to normalize fields + produce rationales.

- **Only re-search** if the Gate says coverage/quality is insufficient (e.g., <10 viable candidates or recall target missed). On re-search, tweak allowlist/keywords/date window and optionally bump depth=`advanced` once.

LangGraph State (minimal)

- `QuerySpec`: {text, city, date_from, date_to, model, version}

- `UserProfile`: {allowlist_domains[], keywords[], prior_feedback}

- `Candidates`: [{url, title?, snippet, score}]

- `Extracted`: [{title, time, location, organizer, url}]

- `Top10`: [{…fields…, recommendation, rationale}]

- `Decision`: {"revise" | "accept", notes}

Agents (product-oriented objectives)

1. **Profile & Planner**
   Builds/refreshes **search profile + plan** from (a) onboarding/pre-approved events (seed **allowlist** of domains that surfaced them **before** they happened), and (b) user feedback. Emits `UserProfile` + tuned `QuerySpec`. On feedback, revises domains/keywords/date window.

2. **Retriever (Tavily Search)**
   Runs **one** cost-effective `/search` (depth basic, include_domains, date filter) to get candidates (URLs + snippets + scores). On Gate-requested re-search, updates parameters (domains/keywords/window, optional depth=advanced).

3. **Extractor / Normalizer**
   Performs **selective `/extract`** on shortlisted URLs to pull text and normalize **title, time, location, organizer, url** (uses LlamaIndex parsing where helpful).

4. **Recommender / Gate**
   Scores normalized items against the profile; marks **approved/not-approved**; generates the **≤350-char rationale**; selects **Top-10**.
   If coverage/quality insufficient → **Decision=revise** with concise feedback (e.g., “add domain X”, “widen to 21 days”), looping back to **Profile & Planner**. Otherwise **accept**.

Functions (supporting, not agents)

- **InputGuard** (validate/complete `QuerySpec`)
- **CacheCheck** (short-circuit on hit)
- **BudgetGate** (cost projection; enforce $0.10 cap)
- **CanonicalizeMerge** (dedupe/upsert events; soft-delete handling)
- **UIFormatter** (shape final Top-10 payload; rationale hidden/click)
- **TelemetryLogger** (agent_runs/query_runs with tokens/costs)

Control Flow (with cycle)

InputGuard → CacheCheck → BudgetGate →
Profile & Planner → **Retriever** → **Extractor** → **Recommender/Gate** →
if **revise** → back to **Profile & Planner** (cycle)
else **accept** → CanonicalizeMerge → UIFormatter → TelemetryLogger.

Behavior & Policies

- **Allowlist seeding:** mine pre-approved events to identify reliable domains; evolve via feedback.
- **Rationale:** sentence-only, ≤350 chars, no URLs (URLs shown separately).
- **Cost hygiene:** default search depth **basic**; **extract only top K**; re-search **only** on Gate failure.
- **Benchmark (later): Recall@10 ≥ 50%** against CSV of pre-approved events.