

Goal and framing

The project brief asked for a prototype AI-assisted system that automatically discovers clinical trials from a seed input and organizes detailed trial information from public sources. I considered two directions: target landscape discovery ("what trials exist for KRAS G12C?") and patient–trial matching ("which trials might this patient qualify for?"). I spent my first few hours on the target landscape tool; it worked, but was low-leverage because ClinicalTrials.gov already supports structured queries for that use case. The harder bottleneck is "what's plausible for this patient," which requires interpreting free-text eligibility criteria against clinical history. I pivoted to patient–trial matching because the potential for real clinical value was higher, and it allowed me to work with what makes agentic tools worthwhile in the first place, which is inference beyond straightforward data retrieval.

Seed input and outputs

Seed input: a structured patient profile (cancer type, biomarkers, age.sex, prior treatments, ECOG, comorbidities, metastases).

Outputs: patient_matches.md (human-readable ranked trials with reasoning), patient_matches.json (structured for integration), trials.json (full records with raw eligibility text), trials.csv (spreadsheet export), landscape.md (target-level summary).

System overview

Starting from a patient profile, the system generates search terms (cancer synonyms + biomarker variants), discovers enrolling trials on ClinicalTrials.gov, retrieves trial metadata and eligibility text, and produces ranked candidates with supporting factors, conflicts, and uncertainties. Trials are classified as HIGH (confidence ≥ 0.80), MEDIUM (0.50–0.79), or EXCLUDED (< 0.50).

Architecture

The core constraint was balancing cost, quality, and auditability. A pure LLM agent loop is expensive and unpredictable. A rules-only parser can't interpret broad language like "adequate organ function." Embedding-only matching struggles with negation ("prior platinum required" vs "prior platinum excluded" look similar in vector space). This led to a multi-stage design: deterministic filters first, then LLM only where deterministic code falls short.

Stage 0, Query-time filtering: Push coarse constraints upstream to ClinicalTrials.gov (age buckets, sex, recruiting-only status) to reduce retrieval volume. Local exact checks remain because API age buckets are inclusive: a trial accepting ages 18–75 appears in both "adult" and "older" searches.

Stage 1, Deterministic screening: Verify structured constraints (status/age/sex), validate NCT IDs, and apply a relevance pre-filter (biomarker in title +0.5, cancer type +0.3; threshold 0.2) to avoid scoring irrelevant trials.

Stage 2, LLM scoring: For viable candidates, the LLM evaluates raw eligibility text against the patient profile and returns supporting factors, conflicts (ECOG, prior therapy, brain mets, comorbidities), and uncertainties from missing data.

Data source and auditability

I used ClinicalTrials.gov only. It provides a stable API, canonical NCT IDs, and eligibility text. EU CTR was scoped out due to schema differences and time constraints. Auditability was a first-class requirement: eligibility criteria are stored verbatim, raw API responses are saved for replay, search terms are tracked with provenance ("manual" vs "LLM"), and all trials are tied to validated NCT IDs. If a match looks wrong, the full chain from query → trial → eligibility text → LLM reasoning is inspectable.

Example output

From a 65-year-old male with Stage IV NSCLC (KRAS G12C, post-chemo/IO):

NCT06881784 — Confidence 95% (HIGH)
Title: RASolve 301: Phase 3 Study of RMC-6236 vs Docetaxel in RAS-Mutant NSCLC
Sponsor: Revolution Medicines
Supporting: ECOG 1, pathologically confirmed NSCLC, prior platinum + anti-PD-1, documented KRAS G12C
Conflicts: none
Uncertainties: none

From an 82-year-old male with CLL (ECOG 3, CHF, CKD):

NCT03331198 — EXCLUDED
Reason: Patient has ECOG status of 3; trial requires ECOG ≤1.

NCT06136559 — EXCLUDED
Reason: Patient has previously treated CLL; trial requires treatment-naive individuals.

Validation

I validated the system using GPT-4o on 15 patient profiles spanning 12 tumor types: standard biomarker-driven cases (KRAS G12C, EGFR, ALK, BRAF, HER2, BRCA1/2) plus difficult edge cases (ECOG 3 with comorbidities, rare sarcoma, active CNS disease). Across these profiles, the system evaluated ~1,349 unique trials with zero hallucinated NCT IDs.

For a standard NSCLC KRAS G12C patient, the system returned 34 HIGH matches from 100 trials. For a post-osimertinib EGFR patient with resistance mutations, it found 14 HIGH matches from 60 trials, resulting in a 23% match rate, up from 11% after search term refinement. For a rare synovial sarcoma case, refined search terms expanded coverage by 61% (100 trials vs 62 baseline). Most importantly, an elderly CLL patient with ECOG 3 and multiple comorbidities correctly matched 0 of 64 trials, with explicit reasoning citing ECOG requirements and comorbidity exclusions. Over-matching is a common failure mode; this negative control verified the system doesn't force matches when a patient is genuinely ineligible.

Prompt calibration

Early outputs failed in domain-specific ways: age 65 was flagged as "above typical range," prior platinum chemotherapy was marked as a conflict even when the trial required it, the model hallucinated facts from missing data, and confidence scores collapsed to a narrow range. Fixes required explicit guardrails: "do not assume facts not in the profile," "age 65 is not atypical for oncology trials," and calibrated confidence rubrics grounded in observed failure modes rather than generic prompts.

Use of AI tools

I used Cursor throughout, bouncing around between Opus 4.5 and Composer 1 for planning and development. I also borrowed ChatGPT initial brainstorming and refining documents, but found that the codebase context, conversation history, and ability to generate artifacts alongside code made Cursor much more powerful for this project. AI helped with boilerplate and rapid prototyping, though it required significant insight, iteration, and deliberate scoping. I kept control over the architecture.

Conclusion

Starting from a patient profile, this prototype discovers relevant trials, structures detailed information including raw eligibility text, and produces ranked recommendations with explicit reasoning. LLMs are used where deterministic code falls short: expanding search terms into clinical synonyms and interpreting unstructured eligibility criteria against patient history. The output is designed for human review, surfacing conflicts and uncertainties instead of making shaky assumptions. The pivot from target landscape to patient matching paid off: the harder problem turned out to be the more interesting one.