

Agentic GIS

Transforming GIS From a Map You Read Into Operational Intelligence You Question.

JEFF FRANZEN · JUNE 2026 · FOR DISCUSSION — INDEPENDENT RESEARCH

A Red Cross leader does not want a map. They want an answer: *how many homes were destroyed in this county, where are the open shelters near them, and how does that compare to the county next door?* Today that answer takes a trained analyst hopping between layers, widgets, and dashboards. The agentic-GIS pattern collapses the hop into a question.

The thesis is one sentence: **the LLM should not know the answer — it should know how to find the answer.** Spatial tools produce the facts; the language model is a router that picks the right tool and translates the result. This is not "ChatGPT over GIS." It is a disciplined inversion of where the intelligence lives.

The architecture in one line

Tools + Data = the intelligence. The LLM = the orchestrator. Read-only spatial functions return structured facts — dictionaries, not prose. The model never computes a number; it chooses which function to call, fills the arguments, and narrates what comes back. Every figure is traceable to a tool, so the agent can be wrong about *which* question to ask but never about the arithmetic.

The five primitives cover half the work

Nearly all operational spatial reasoning at the Red Cross reduces to five read-only functions. They use no AI. They return facts.

1 **LOCATE**
get_county_summary
FIPS code or fuzzy county name in; the full canonical row out — administration, vulnerability, current operations.

2 **NEAR**
find_shelters_near
A point and a radius in; resources within a true metric circle out (STRtree index, equal-area CRS — not degrees).

3 **COUNT**
count_operational_metrics
Damaged homes, volunteers, DAT incidents — aggregated cleanly inside a county, chapter, or division boundary.

Where the line is drawn — governance

The discipline is sharper than the architecture. There is a bright line between analytics and judgment, and the agent must never cross it.

"1,247 damaged homes in County A" is analytics — safe, factual, defensible. "County A is under-resourced" is a resource-allocation judgment — and that crosses the line. Build tools that count and rank. Keep the decision human-owned.

THE GOVERNANCE RULE — THE AGENT COUNTS, IT DOES NOT DECIDE

This is the difference between a tool leadership will trust and one they will quietly shelve. The agent refuses to recommend an allocation, carries data caveats verbatim, and never fabricates a figure. It makes the human faster, not absent.

Strategy met working code in the same 48 hours

The NotebookLM notebook reasoned *top-down* from a GeoAI tutorial and a three-model debate to a hand-it-to-Claude-Code spec (Project Keystone). The answer-engine repo built *bottom-up* to a running Phase-2 proof of that exact spec — on public RAPT, HIFLD, NRI, and emPOWER data. They meet precisely.

KEYSTONE SPEC (TOP-DOWN)	ANSWER-ENGINE, RUNNING (BOTTOM-UP)
Canonical data contract	catalog/*.json — 12 layers, fields + caveats as machine-readable contracts
Locate primitive	county_risk_profile recipe
Near primitive	wildfires_near_place, hospitals_near_place, exposure-ring intersects
Count primitive	count_features + high_risk_county_facilities
Summarize primitive	hurricane-scenario KPI sweep — population, vulnerable, healthcare, public safety, emPOWER
Compare primitive	compare_counties — added June 2026; the last of the five
LLM-as-router, never recall	backend/agent.py — Claude picks a recipe, never invents a number
Count, don't decide	non-fabrication rule + verbatim caveats; refuses allocation calls

The honest part — where the real work is

answer-engine was built fast **because the public layers were already API-shaped**. Keystone's hard part is the *internal* data — DDA, shelters, DAT, volunteers — un-joined and trapped inside Experience Builder widgets. As the debate put it: *the hard part isn't the loop, it's the data plumbing*. Keystone is a **data**

project wearing an AI costume. The agent is almost an accessory; the win is one FIPS-keyed county table that makes every system speak the same language.

The ask

Phase 0.5 — inventory the trapped tools. Experience Builder projects aren't maps; they are functions hiding inside widgets — `count_DDA`, `filter_by_county`, `compare_counties` already exist, disguised. Cataloguing them extracts the tool library and seeds the canonical table at once. Use `answer-engine` as the live proof-of-concept in any pitch: it makes the bet concrete instead of theoretical, before a dollar is spent on internal plumbing.

The one-sentence version: stop building dashboards people have to read, and start building tools that answer the question — where the model finds the answer and the human still owns the decision.