---

title: Smart-Query Platform — County Intelligence Reasoning Engine date: 2026-04-17 status: active area: red-cross priority: high url: https://explorer.jbf.com/api/smart-query repo: https://github.com/franzenjb/vulnerability-explorer tags: [smart-query, lightrag, claude, tool-use, rag, county-intelligence, strategic]

Smart-Query Platform — Strategic Plan

> Operationalizing a four-tool Claude router (SQL + knowledge graph + full-text + AGOL) across the Red Cross app ecosystem.

Context

Built over the session of 2026-04-17: a county intelligence reasoning engine at `explorer.jbf.com/api/smart-query`. Claude-routed tool-use endpoint with four retrieval modes:

- **`supabase_sql`** — read-only SQL over `county_rankings` (3,144 counties × ~50 fields), `fema_declarations`, `alice`
- **`lightrag_query`** — semantic RAG over 20,686 processed knowledge-graph documents (ALICE, NRI, SVI, FEMA declarations, Wikipedia)
- **`text_search`** — Postgres tsvector+GIN full-text search over Wikipedia articles for every US county
- **`agol_query`** — live public AGOL feature services (FLARE Fire Incidents CY2024 — 103,400 real incidents with RC response categorization)

Plus automatic Red Cross chapter/region/division enrichment (names + codes) on any county-specific answer.

The strategic question: Move this from Dragons personal tool to something that measurably changes how Red Cross chapter/region/division leadership operates — and produce demos that land with executives.

Strategic Framing (the big idea)

Most AI-at-Red-Cross efforts wrap ChatGPT around policy docs. What we have is categorically different:

1. **Grounded computation.** Ranking/counting/aggregation answers are exact SQL — not approximations from retrieved chunks. "Top 5 counties by ALICE households" returns the real top 5, not whichever 5 got retrieved.
2. **Federal data assembled.** FEMA NRI, CDC SVI, ALICE, Census ACS, FEMA disaster declarations, Wikipedia — joined at the county level through one interface. Assembly is the hard part; data is public.
3. **Red Cross-native scoping.** Every county carries its chapter/region/division. No other RAG deployment at Red Cross knows this hierarchy.
4. **Auditable.** Every answer includes a `trace` showing which tools ran, what SQL was executed, what LightRAG returned. Not a black box.

The shift this enables: A chapter director, region VP, or division EVP can ask a question of their data without waiting for GIS or National. Analyst work drops from days to seconds.

Current Integration Status (Tier 1 — complete 2026-04-17)

All four apps share the same brain. Fix a bug or add a capability at `explorer.jbf.com/api/smart-query` and all four inherit it.

App	URL	Integration	Status
Vulnerability Explorer	explorer.jbf.com	Hosts `/api/smart-query`. Red Cross persona sample questions.	Live
Spatial Ops	ops.jbf.com	QUERY tab proxies to smart-query with state context. Right panel resizable.	Live

App	URL	Integration	Status
County Intel	county.jbf.com	Narrative panel rewritten. Follow-up chat input. Free-form questions routed.	Live
Situational Awareness	sitaware.jbf.com	Floating Ask drawer, gated behind `?ask=1` / Cmd-K / `askEnable()`. DOC-officer tactical prompt.	Live

Tier 2 — New surface: `intel.jbf.com` (1 week)

A chapter/region/division-scoped dashboard. First surface thats not Dragon-only.

User flow: 1. Land on `intel.jbf.com` → pick scope from dropdown (chapter / region / division) 2. Auto-load pre-built dashboard: vulnerability summary, hazard exposure, ALICE population, recent FEMA declarations, fire response stats 3. Chat box below: ask anything, scoped automatically to your counties

Architecture: thin frontend calling `/api/smart-query` with a scope-prefix injected into the system prompt.

Tier 3 — Proactive intelligence: briefing generator (1-2 weeks)

Push, not pull. The biggest leadership-impressing thing in the roadmap.

- **3A. Weekly division digest (cron-driven email):** FEMA declarations in the last 7 days, counties crossing hazard thresholds, ALICE trend shifts, smart-query-synthesized narrative.
- **3B. Event-triggered alerts:** New FEMA declaration → rapid exposure brief auto-emailed to chapter/region VP within seconds.

Powered by: cron + smart-query + `gws-gmail` CLI. The existing [[projects/disaster-sitrep-generator]] skill can be the synthesis engine.

Tier 4 — Platform layer: `dragon-brain-api` (refactor)

Move `/api/smart-query` out of explorer.jbf.com into a standalone hosted service. Every Red Cross-adjacent app hits the same endpoint. One place to add tools, tune prompts, log usage, version the API. Least flashy, most durable.

Extension Tools to Add

The four current tools cover county-level structured + narrative + exact-match + fire ops. Discipline: each new source must earn its place by answering a *class of question* no existing tool can.

Tool	Data source	Unique question class
`shelter_query`	Real shelter data (AGOL, not the fictitious Supabase table)	Operational shelter queries — where needs sourcing
`nws_alerts`	NWS API	"What active alerts intersect my chapter right now?"
`realie_parcels`	Real parcels (scaling beyond Pinellas)	Shelter-candidate property searches
`disaster_detail`	FEMA OpenFEMA API	Deep-dive on specific declaration numbers

Leadership Demo Scripts

Three 90-second demos, each answering a question a real VP has asked.

Demo 1: The Pre-Season Hurricane Brief

"*Rank counties in the Southeast and Caribbean Division by hurricane risk and ALICE household count. Show chapters for the top 10.*" → Markdown table with exact numbers + chapter coverage. 18-second answer vs. three-day GIS request.

Demo 2: The After-the-Declaration Brief

"*FEMA just declared a disaster in Iberia Parish, LA. Give me a rapid exposure brief.*" → Full county profile + prior disasters + chapter + geographic context. 20 seconds vs. week-long request from National.

Demo 3: The Defensible Ranking

"*Five most populous counties with the highest ALICE household counts.*" → LA 1.53M, Harris 780K, Cook 707K, Maricopa 588K, Miami-Dade 502K. → Contrast with ChatGPTs answer (fabricated Polk County IA at 50K). Side-by-side.

Bonus: The Service-Gap Demo (added 2026-04-17)

"*Top 10 chapters by fires without RC notification in 2024.*" → ARC of Greater Chicago leads with 1,154 real fires. Service delivery failure map no one else at Red Cross has exposed this plainly. → Powered by FLARE + `agol_query` tool.

Operationalization

1. **Cost:** ~$0.03-0.05/query Opus. 1,000/day ≈ $1-1.5K/month. Fine; plan for $5-10K/mo at chapter-wide scale. Redis cache + Haiku for simple lookups when needed.
2. **Auth:** Open now. Tier 2 (`intel.jbf.com`) launches with Vercel middleware email allowlist or AGOL OAuth tie-in.
3. **Observability:** Log every query + trace. Learn what people actually ask. Prioritize new tools from real usage.

Privacy Guardrails

- FLARE AGOL layer tested anonymously — publicly accessible, no AGOL login required.
- `agol_query` tool strips Latitude/Longitude from all responses. Point-level data goes through [[projects/flare-v2-red]] app where 500m client-side jitter is applied.
- Belt-and-suspenders: both LLM-layer (system prompt) and server-layer (row stripping) privacy.

Recommended Sequence

1. **Week 1 (complete):** Tier 1 deployed across four apps.
2. **Week 2:** Record the three demo videos. Show leadership.

3. **Week 3-4:** Build `intel.jbf.com` (Tier 2). Alpha with 3-5 chapter directors.
4. **Month 2:** Weekly division digest (Tier 3A) + platform refactor (Tier 4).
5. **Month 3+:** Event-triggered alerts (Tier 3B) + extension tools driven by real usage.

Critical Files (reference)

- `~/dev/vulnerability-explorer-deploy/api/smart-query.js` — the engine (canonical)
- `~/dev/vulnerability-explorer-deploy/index.html` lines 1470-1620 — frontend UI pattern
- `~/dev/spatial-ops-jbf/app.js` — ops QUERY tab integration
- `~/dev/county-intel-v2/index.html` — county narrative panel rewrite
- `~/dev/sitaware/index.html` — Ask drawer (gated)
- `~/dev/dragons-brain/populate_wiki_supabase.py` — template for bulk Supabase population

Related Notes

- [[projects/dragons-brain-lightrag]] — the knowledge graph backing lightrag_query
- [[projects/arc-real-estate]] — chapter realignment engine, candidate for Tier 2-ish embed
- [[sessions/2026-04-16]] — LightRAG data ingestion + Supabase table discovery
- [[sessions/2026-04-17]] — smart-query build session (this work)