**“Local Activity Finder”**

**Research Document**

**Version 1.0**

**10/18/2025**

### **Version 1.0** - *10/18/2025* – M. Rado – Initial document creation and base research added.

## 1) Provider choices (why these?)

### Activities / Places — Geoapify Places

* **Why**: more reliable responses and broad category coverage; generous free tier; straightforward category filters we can map to indoor/outdoor.
* **Tradeoffs**: some categories are noisy or generic; requires API key in production.

### Weather — Open-Meteo

* **Why**: free, no-key API; stable uptime; hourly data supports our indoor/outdoor recommendation rule; fast and simple.
* **Tradeoffs**: interpolation quirks in mountainous/coastal regions; we normalize to { tempC, windKph, precipMm, precipProb, hourly }.

### Geocoding — Nominatim (OpenStreetMap)

* **Why**: free with global coverage and solid reverse geocoding; aligns with OSM ecosystem used by activities.
* **Tradeoffs**: strict usage policy (identify user agent, avoid heavy/bulk usage); quality varies by region.

Note: We trialed OpenTripMap for activities first; reliability issues (timeouts/empty payloads) led us to prefer Geoapify. We maintain a category mapping to set the indoor flag.

## 2) Rate-limit policies & client behavior

### General approach

* **Exponential backoff** on upstream 5xx/429 with a bounded retry count.
* **Short-TTL caching** on our side to reduce calls and latency.
* **Rounded coordinate caching** (e.g., weather keyed by lat/lon rounded to ~2–3 decimals) to boost cache hits without harming UX.

### Suggested TTLs (server-side)

* **Weather**: 5–10 minutes per rounded (lat,lon).
* **Geocode**: 10–15 minutes per (query, limit).
* **Activities**: 2–5 minutes per (lat,lon,radius,type).

### Mobile guidance (client-side)

* **Geocode**: in-memory results cache 5–15 minutes for suggestions.
* **Weather/Activities**: cache 2–10 minutes; allow pull-to-refresh to bypass.

CI & tests use **mock mode** with fixtures under storage/app/fixtures/<set>; .env.testing and default fixtures are seeded so tests don’t hit providers.

## 3) Data licensing & usage notes

Always verify current provider terms before release.

* **Nominatim / OSM**: Usage policy expects an identifiable User-Agent and rate-friendly access; ODbL applies to database-derived data. For simple geocoding results, attribution “© OpenStreetMap contributors” is typically expected. Cache; do not bulk-harvest. Local database should be used in production.
* **Open-Meteo**: Free to use; attribution recommended; no key required. Respect fair-use via caching and backoff.
* **Geoapify**: API access under plan quotas; API key required; attribution (“Powered by Geoapify”) typically required. Check plan-specific caching/redistribution rules.
* **OpenTripMap (if retained)**: free tier, attribution; reliability concerns suggest keeping it behind a feature flag.

**Attribution in UI (lightweight):**

* Weather: “Weather data by Open-Meteo”.
* Activities: “Places data by Geoapify”.
* Geocoding: “Geocoding by Nominatim / © OpenStreetMap contributors”.

## 4) Risks & mitigations

### Upstream reliability / spikes

* **Risk**: 429s/timeouts → blank screens or slow UX.
* **Mitigation**: exponential backoff + bounded retries; short-TTL cache; user-visible retry affordance.

### Data sparseness / quality variance

* **Risk**: Some regions lack rich POIs or names; geocodes may be imprecise.
* **Mitigation**: category fallbacks; name fallbacks (use category when name is null); avoid hard UI dependencies on rich metadata.

### Rate-limit or policy violations

* **Risk**: excessive traffic or missing attribution can trigger blocks.
* **Mitigation**: cache and back off; identify UA; ensure attributions are visible; document quotas.

### Fixture drift (mock vs live)

* **Risk**: Tests/UX diverge from production if fixtures go stale.
* **Mitigation**: keep fixtures small but realistic; add a regeneration checklist; ensure identical normalization in live vs mock paths.

### Privacy & key handling

* **Risk**: leaking provider keys or sensitive coordinates.
* **Mitigation**: server-side secrets; env-based keys; do not log high-precision coordinates; rotate keys.

## 5) What to revisit later

* Stronger ranking features for recommendations (popularity, hours, price tier).
* Multi-provider fallback (e.g., Mapbox/HERE) if Geoapify coverage becomes a constraint.
* API versioning (/v1) once contracts stabilize.
* Offline “fixture packs” for mobile end-to-end demos.