International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(6s) 2026 pp. 1404-1419
https://doi.org/10.70917/ijcisim-2026-6172
Article

GeoTileIQ: A Compliance-First Adaptive Framework for Regional Place-Data Acquisition, Integration and Quality Scoring under Official API Constraints

Miryala Rajesh Kumar[1], Dr. Kiran B M[2], Gattu Prasad[3]

[1,2,3]Sphoorthy Engineering College, Department of CSE AND Sphoorthy Engineering College with Nadargul village,Sagar Road,Saroornagaer Mandal (Balapur),RR Dist, Hyderabad

Abstract: Consumer mapping platforms are the richest and freshest source of location-based business data; however, they permit bulk collection only by browser automation, which is fragile and may violate platform terms, or via an official place-search interface that returns few results per query, charges by queried data field and imposes request-rate limits. This paper explores how an authorising access based complete, accurate and affordable regional place dataset can be constructed. We introduce GeoTileIQ, a compliance-first framework with five technical contributions: (i) an Adaptive Quadtree Tiling Strategy (AQTS) based on a full result page as an online saturation signal, which has a completeness guarantee and bounds on the number of requests per tile; (ii) a cost-aware field-mask planner based on a closed-form break-even policy; (iii) a policy governor comprising token-bucket pacing, jittered exponential back-off, retention timers, and an audit log; (iv) a multi-source entity-resolution method using spatial blocking, a composite name–distance–category score, and a one-to-one assignment function; (v) a Data Quality Index with Bayesian-smoothed ratings and an aspect-level review-sentiment analyser. AQTS achieved 100% coverage with 516 requests compared with 1,035 requests for the smallest complete uniform grid (47.3% fewer requests per month); a success rate of 100% with 516 requests compared with 67.1% with 1,035 requests; and an F1 of 96.2% versus 39.4% for distance-only matching. The results are simulated and validation on a live interface is marked as a future task.

Keywords: place intelligence; Places API; adaptive spatial tiling; quadtree; entity resolution; POI conflation; data quality; aspect-based sentiment; responsible data acquisition

1. Introduction

Location intelligence is now commonplace and has become an integral part of retail site selection, delivery routing, tourism analytics, public-health access studies and urban planning. A query like “where are under-served neighbourhoods for primary health care?” needs to access the names, coordinates, categories, opening status and public ratings of thousands of primary healthcare facilities. This data is gathered on an ongoing basis from consumer mapping platforms, which have a greater coverage area and are updated more frequently than any single organisation. While manual collection is slow and impractical for doing at “scale,” automation is appealing, but what constitutes an appealing automation is important. Accessing a rendered map page with a browser will render unstructured output, fail if the page's layout changes and will not fit with platform terms of use. A large-scale 2025 scraping report of self declared bots indicates that compliance with access directives drops with the increasing strictness of these rules, and that compliance is low and bots do not check directives at all [1]; a companion study reports a significant tightening of site-level restrictions since generative AI began to get widespread usage [2]. Over time, platforms are implementing increasingly robust technical measures, making pipelines that rely on scraping increasingly unreliable. Official place-

search interfaces are also available, providing a stable, structured and authorised alternative, but do not support full-scale region harvesting.

Each query returns a small page of results (at most 20 places per page for the current Text Search interface [12]), only a bounded number of pages can be requested, charges depend on the data fields requested, and request rates are limited. Turning such an interface into a dependable regional data source is an algorithmic and engineering problem in its own right. It is also an under-studied one: existing work analyses scraping and its exposure [1], [2], integrates points of interest (POIs) that have already been collected [3], [7]–[9], or analyses reviews that have already been collected [4], [5], [10], whereas the acquisition problem under official constraints is left to practitioners.

This paper makes the following contributions:

- **Saturation-driven adaptive tiling (AQTS).** A density-agnostic search strategy that uses a full result page as an online saturation signal, with a completeness proposition and analytical bounds on request count (Section 4.2).
- **Cost-aware retrieval planning.** A closed-form break-even rule for choosing between rich-field search and IDs-only discovery with selective enrichment (Section 4.3).
- **Compliance-by-design governor.** Pacing, recovery, retention and audit placed in the request path so that quota and retention rules are enforced rather than left to user discipline (Section 4.4).
- **Composite entity resolution and record trust.** A fusion method for official and open map data, a Data Quality Index, Bayesian-smoothed ratings and an aspect-level review signal (Sections 4.5–4.7).
- **Reproducible evaluation.** A controlled simulation of documented API behaviour, with baselines, ablations, sensitivity analysis and an explicit account of threats to validity (Sections 5–7).

The rest of the paper is organised as follows. Section 2 reviews related work, Section 3 formalises the problem, Section 4 presents the framework, Section 5 describes the experimental set-up, Section 6 reports results, Section 7 discusses implications and limitations, and Section 8 concludes.

2. Related Work

2.1 Compliance and the scraping environment

Kim et al. [1] conducted the first large-scale controlled study of how scrapers respond to robots.txt directives, using about 3.9 million requests to 36 institutional websites and an eight-week experiment with four directive versions. Crawl-delay directives were honored by bots most of the time, and a full disallow was honoured by the bots least of the time, with several categories of bots not re-checking the file, and user-agent spoofing was detected in 18 known bots. Longpre et al. [2] report that access to the Web has been tightened up quickly since the advent of generative AI. All of these studies suggest that voluntary compliance should not be taken for granted, platforms are becoming increasingly technical, and circumvention-based pipelines are becoming increasingly unsustainable. There's no compliant collection method either of the studies, which GeoTileIQ is supposed to provide.

2.2 POI integration and conflation

The authors of Amiri et al. [3] merge the release of over 100 million Foursquare places across the globe with OpenStreetMap (OSM) because it is rich and often updated, but without a formal process to ensure that a feature is in fact a business. Bertozzi [7] evaluates a tunable rule-based scorer and a supervised random-forest matcher on hand built golden sets and suggests governance templates to assure transparency with regard to quality control. Low et al. [8] suggest a six-step end-to-end conflation process: procurement, schema standardisation, taxonomy mapping, matching, unification and verification, while Novack et al. [9] define POIs as graph nodes that have match possibilities as edges. The explainable composite score and one-to-one assignment used here, and other such works motivate such scores, but all begin with data already gathered. The approach of matching founded in the underlying record-linkage

tradition [14, 15] is based on scoring candidate pair matching under a decision rule, and we proceed along this line with the threshold sensitivity analysis.

2.3 Review analytics and POI recommendation

Zaid et al. [4] constructed an annotated corpus comprising over 3,500 Arabic Google Maps reviews on six aspects, where multi-head BERT models scored better than classical classifiers, reaching F1 = 0.97 for the "cleanliness" aspect against F1 = 0.91 for the best classical model, while the abstract "overall experience" aspect remained challenging. In the context of natural language ambiguity in aspect-based sentiment analysis of hotel reviews, Nadeem et al. [5] focuses on resolving this problem. By analyzing 23,906 reviews of the cities in Atlanta, Li et al. [10] use sentiment analysis based on BERT to investigate attitudes toward density. In the recommender part, both Bellogín et al. [6] and Zhang et al. [11] highlight the importance of the quality of the data sets and their privacy-sensitive processing. These studies demonstrate the usefulness of place-linked analytics when a large and reliable place-set exists, but no one has reported on how such a place set is put together or how complete it is.

2.4 Spatial decomposition and resilient request handling

The recursive quadrant decomposition is a classic method of resolving resolution to local density [13] and the exponential back-off with randomised jitter is a common method for avoiding synchronised retries when a device is contended [16]. These two are both new, and the result cap itself is the density probe that leads to subdivision; the retry policy is inherent in a governor that also sets up retention and audit rules.

Table 1. Related work and the gap addressed by GeoTileIQ

Ref.	Year	Contribution	Limitation addressed by GeoTileIQ
[1]	2025	Large-scale evidence of partial scraper compliance with access directives	No acquisition method; highlights risk of unauthorised collection
[2]	2025	Documents tightening of web access restrictions	No compliant collection pipeline offered
[3]	2025	Global POI fusion of Foursquare and OSM	Pre-released data; no query-level acquisition under caps
[4]	2025	Aspect-based sentiment benchmark on Maps reviews	Assumes reviews already collected; no place-level fusion
[5]	2025	Ambiguity-aware aspect sentiment, hotel reviews	Domain-specific; not linked to acquisition or quality scoring
[7]	2025	Rule-based and supervised OSM conflation with governance	Does not address collection within API limits
[8], [9]	2021, 2018	End-to-end and graph-based POI conflation	Source data assumed available; cost and compliance out of scope
[10]	2024	Urban sentiment from place-linked reviews	Does not describe how the place set was assembled or its completeness

2.5 Research gap

We identify five gaps: **G1** no density-adaptive acquisition strategy for capped official interfaces; **G2** field-tier billing is rarely modelled and no decision rule exists for choosing a retrieval mode; **G3** policy enforcement is usually left to the user; **G4** acquisition, fusion, quality scoring and review analytics are studied separately; and **G5** record-level reliability scores are seldom produced. GeoTileIQ addresses G1–G5 in one framework and evaluates each component quantitatively.

3. Problem Formulation

Let Ω be a rectangular region containing an unknown finite set of places $\mathcal{P}$ with $|\mathcal{P}| = N$. A place-search oracle $\mathcal{Q}(q, B, j)$ takes a keyword q, a bounding box $B \subseteq \Omega$ and a page index $j \in \{1,\ldots,L\}$, and returns up to p places ranked by prominence. The per-query cap is therefore $K = p \cdot L$. We adopt the following assumptions, which reflect documented behaviour of the interface [12].

Assumption A1 (cap exhaustiveness). If $|\mathcal{P} \cap B| < K$, the pages of $\mathcal{Q}(q, B, \cdot)$ together list every place in B; otherwise they list exactly K places chosen by an unknown prominence ranking.

Assumption A2 (tiered billing). A search request that asks for identifiers only costs cids; one that asks for rating and website fields costs crich > cids; a per-place detail lookup costs cdetail.

Assumption A3 (rate limit). Requests arriving faster than a provider limit ρ are throttled and fail; throttled requests may be retried.

The acquisition objective is to find a set of queries that minimises the number of requests R (and cost) subject to coverage $|S \cap \mathcal{P}| / N \geq 1 - \varepsilon$, where S is the set of retrieved places, using no mechanism that circumvents the interface's protections. Because N and its spatial distribution are unknown, the strategy must discover the required resolution during the run. A second stage must fuse S with an open-data source O into unified entities and attach to each a reliability score and, where licence permits, aspect-level insight.

4. The GeoTileIQ Framework

4.1 Overview

GeoTileIQ consists of eight cooperating modules (M1–M8) arranged in five layers (Fig. 1). Requests travel downward from the user and results flow back up; each layer communicates only with its neighbours, and the compliance and control layer lies between the user and the acquisition layer, so no request can reach an external service without passing policy checks. M1 validates the job and estimates its cost from a pilot of one request on the root tile and a few sub-tiles; M2 plans the field mask; M3 is the sole gateway to external services; M4 runs AQTS; M5 wraps the official Places API and the OSM service behind one connector interface; M6 resolves entities; M7 scores quality and extracts aspect sentiment; and M8 stores, exports and audits.

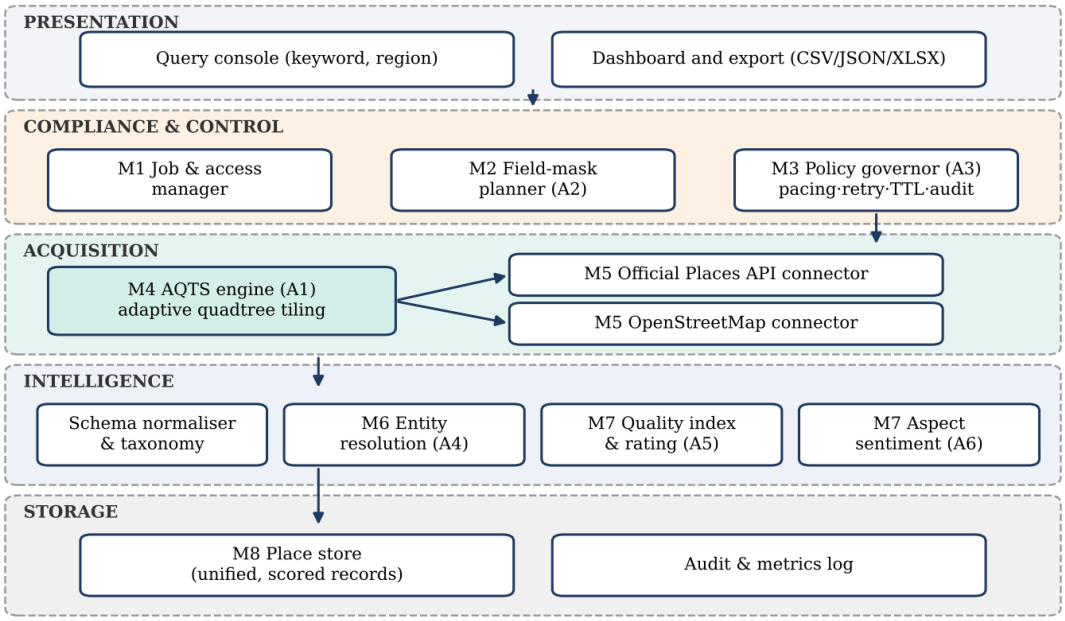


Fig. 1. Layered architecture of GeoTileIQ. A1–A6 denote the algorithms of Sections 4.2–4.7.

4.2 Adaptive Quadtree Tiling Strategy (AQTS)

AQTS (Algorithm A1) starts with Ω as the root tile. It dequues a tile and continues to request subsequent pages until until the number of free places on a page is less than p or the page limit L is reached. If the tile returned the full cap K and is larger than the minimum cell size m, then the tile is saturated and is divided into 4 equal children, and they are enqueued, otherwise it becomes a leaf. Results are deduplicated where possible based on place identifier, so that places along tile borders can return the results from the adjacent tiles. When a minimum-size tile is saturated, resolution-limited is entered in the audit log, along with an estimate of perhaps omitted places.

Algorithm A1: AQTS

Input: region R, keyword q, page size p, page limit L, min cell size m

Output: place-ID set S, leaf-tile map T

```
K <- p*L; S <- {}; T <- {}; Q <- queue(R)
while Q not empty:
    tile <- dequeue(Q); found <- {}
    for page = 1..L:
        batch <- governedSearch(q, tile, page)   // via A3
        found <- found U batch
        if |batch| < p: break
    S <- S U found
    if |found| = K and size(tile) > m: enqueue(Q, 4 children of tile)
     else: T <- T U {tile}
return S, T
```

Proposition 1 (completeness). Under Assumption A1, when AQTS terminates every place in Ω either belongs to S or lies in a resolution-limited leaf.

Proof sketch. Children partition their parent, so the leaves partition Ω. A leaf that is not resolution-limited returned fewer than K places, so by A1 it was listed exhaustively. The union of these listings, together with the resolution-limited leaves, covers Ω. ■

Proposition 2 (request bounds). Let I be the number of internal (split) tiles. Each split replaces one leaf by four, so the number of leaves is $\ell = 3I + 1$ and the total number of tiles is $T = 4I + 1$. Every tile costs between 1 and L requests, hence

$$T \le R \le L \cdot T, \quad \ell \ge \lceil N / (K - 1) \rceil \quad (1)$$

where the second inequality is true, when there is no leaf that is resolution-limited, because a non-saturated leaf lists at most K − 1 places. For the evaluation setting (Section 5), $\ell \ge 5000/59 = 85$ (which is the ceiling of 5000/59), and thus a lower bound on leaves is independent of the details of the strategy.

A single tile with exactly K cells is unnecessarily divided, a minor loss which is the cost of a guarantee that no tile is ever accepted with its cells filled. The second design decision is at what point to make a split. Splitting after the first full page saves requests for that tile, but creates a lot of tiles if the tiles have only 20/40 places to fill; fetching all L pages before deciding trades a few more requests per tile for a much smaller tree. The quantification of this trade-off is presented in Section 6.2.

Worked example. Assume that the root tile returns 60 results, which is divided into 4 children. Depending on the result returned by the other child, after two requests the first child would know whether it was a leaf or a split, with the first case yielding 31 results on two pages, and the second yielding 60 results on two pages. The internal tile that is split gives it 3I + 1 leaves: with I = 58 splits it produces the 175 leaves counted in Section 6.1.

4.3 Cost-aware field-mask planner

For a job needing only identity, location and category, the planner selects the cheapest field mask. When richer fields are needed it compares two retrieval modes. With R search requests, N discovered places and a fraction s of places requiring enrichment, the costs are

$$C_{rich} = R \cdot c_{rich}, \quad C_{two\text{-}phase} = R \cdot c_{ids} + s \cdot N \cdot c_{detail} \quad (2)$$

and two-phase retrieval is cheaper when $C_{two\text{-}phase} < C_{rich}$, that is, when

$$s < s^* = R(c_{rich} - c_{ids}) / (N \cdot c_{detail}) \quad (3)$$

The intuition is that a rich-field search returns many places per request and so has a low cost per place, whereas a detail lookup is billed per place and is expensive per place; two-phase retrieval pays off only when few places need enrichment, as in a *refresh* where most places are unchanged. In refresh runs the planner prefers IDs-only discovery: new identifiers are enriched, identifiers that vanished are flagged as possibly closed, and unchanged identifiers are left alone. Unit costs are configuration read from the provider's current price list, so the rule adapts when prices change.

4.4 Policy governor

The only way to external services is through the governor (Algorithm A3), which has 4 duties. Pacing: admission of requests to a token bucket with refill rate r < the provider limit ρ; while a token is available, recovery is possible, but a small burst is allowed, as long as the number of requests is at least r and the number of burst is at most C, the jitter prevents parallel workers from retrying in lockstep. A request that fails all its attempts is added to the dead-letter list and retried at the end of the job, without silently "losing" the tile. Retention: Each provider attribute stored in the store has a fetch time and a time-to-live; content will be purged or refreshed from the sweep by the store when it has expired, retaining only permitted identifiers. Audit: All requests, retries, delays and purges are recorded with time, tile and outcome. The governor never tries to break a limit, it only slows down to honor the limit. Because

different providers implement different rules for retention and/or attribution, they are not stored as static data but as an administrator reviewed configuration.

Algorithm A3: governedSearch(query, tile, page)

```
State: tokens, capacity C, refill rate r, attempt counter a
1  wait until tokens >= 1                // refill at rate r up to C
2  tokens <- tokens - 1; log(request)
3  resp <- send(query, tile, page)
4  if success: return resp.places
5  if throttled or transient failure:
6      a <- a + 1
7      if a > maxAttempts: deadLetter(tile, page); return {}
8      delay <- min(dMax, d0 * 2^a) * (0.5 + random())
9      log(retry, delay); wait(delay); goto 1
10 if permanent error: log(error); escalate to administrator
```

4.5 Multi-source entity resolution

Records come from the official source G and open source O and have been normalised to a common schema (identifier, name, coordinates, category, rating, rating count, source, fetch time): names are lowercased, punctuation and abbreviations standardised, categories mapped to a small common taxonomy, and coordinates projected to metres. Exhaustive pair comparison is quadratic, and records are stored in a 250m space map (250 m cells, blocking), and candidates are searched in the 3x3 neighbourhood of each record. A candidate pair (o, g) with distance d receives the composite score

$$\sigma(o,g) = w_n \cdot ns(o,g) + w_d \cdot e^{-d/\tau} + w_c \cdot 1[cat(o) = cat(g)] \quad (4)$$

where *ns* is name similarity, the mean of a prefix-weighted character similarity (robust to typographical errors) and a token-overlap measure (robust to reordering and dropped suffixes). When the open-source name is missing, a fallback $\sigma = 0.75 \cdot e^{-d/\tau} + 0.25 \cdot 1[cat] - 0.05$ links unnamed features by place and type. Pairs with $\sigma \geq \theta$ are accepted greedily in descending score order, and a pair is accepted only if neither record is already matched, which yields one-to-one links. Matched records are merged into one entity retaining all source identifiers and the best attribute from each source; unmatched records become single-source entities with lower agreement. Chain branches sharing a name are separated by the distance term; since a perfect-name chain branch 200 m away still scores $0.5 + 0.4 \cdot e^{-200/60} + 0.1 \approx 0.61 > \theta$, correct separation relies on greedy ordering assigning each record to its closer counterpart first, and a branch whose true counterpart is absent can still be mislinked (Section 6.6).

4.6 Data Quality Index and Bayesian-smoothed rating

Each entity receives a Data Quality Index (DQI) in [0, 1] combining four normalised components:

$$DQI = 0.35 \cdot Completeness + 0.20 \cdot Freshness + 0.25 \cdot Agreement + 0.20 \cdot Reliability \quad (5)$$

Completeness is the share of required attributes present; *Freshness* decreases with fetch age relative to the time-to-live; *Agreement* is 1 when two sources confirm the place and agree on name and location and lower for single-source records; *Reliability* grows with the rating count and saturates at a configured value. The weights emphasise completeness and cross-source agreement and are configurable. To prevent small samples from dominating rankings, ratings are shrunk toward the global mean μ:

$$r_{smooth} = (m \cdot \mu + n \cdot r) / (m + n) \quad (6)$$

where r is the observed average, n the number of ratings and m the prior weight in pseudo-ratings. With $\mu = 4.1$ and $m = 25$, a 5.0 average from three ratings becomes 4.20, whereas a 4.6 average from 410 ratings becomes 4.57.

4.7 Clause-level aspect sentiment

Where the source licence allows review text to be processed, each review is split into clauses at punctuation and at the words “but” and “and”, which isolates opposing opinions in one sentence (“the food was delicious but the bill was expensive”), a recognised source of error in aspect-based sentiment analysis [4], [5]. Clauses are mapped to aspects (lists of keywords: food, service, price, cleanliness, ambience); opinion words add +1 (or −1 if preceded by a negator) to the score of a clause; clause scores are added up by aspect and mapped to positive, negative or neutral. The per-place aspect is all that is maintained, and no identity is kept for the reviewers. The analyser is deliberately light and easy to interpret; a trained transformer can take its place via the same interface where labelled data exist.

4.8 Complexity, scalability and refresh

AQTS is also the result of most L·T requests, and since the tiles are independent, multiple workers can consume the queue simultaneously with just the governor's one global token bucket to ensure that the combined rate does not exceed the provider's limit, while scaling simply by adding more workers. Blocking reduces the number of comparisons to $O(|O|\cdot k)$, where k is the average number of records in a 3×3 neighbourhood of 250 m cells. The queue, result set and tile map are checkpointed to allow a interrupted job to resume from where it left off, and the tile map stored enables a later refresh to re-query only tiles with changed results. No completeness is ever overstated as persistent failures are listed in the final report and tiles that are still wet when cell size is minimum are reported as resolution-limited. If the source is not available the pipeline goes on with the remaining sources, and reduces the agreement portion of the affected records.

5. Experimental Set-up

Evaluating against a live service would be costly, rely on provider behavior that changes over time, and necessitate the redistribution of provider content. We thus created a controlled simulation of documented interface behaviour: our results get ranked according to prominence, are returned 20 per page, and we cap the number of pages returned to 3 (60 results), which causes a request-rate limit error. The synthetic city is a 20 km × 20 km square with 5,000 places, 70% drawn from six Gaussian clusters (dense commercial centres) and 30% uniform. The open-data source contains 65% of the places with realistic imperfections: typographical errors, dropped or reordered name tokens, abbreviations, 40 m coordinate noise with 5% outliers near 200 m, 8% missing names and 10% category disagreement. It also contains 240 chain-branch places sharing eight brand names and 300 features that exist only in the open source, giving 3,525 open records of which 3,225 correspond to true links. All experiments use a fixed random seed. Table 2 lists the parameters. Baselines are a single query, uniform grids of increasing resolution, distance-only, name-only and distance-plus-category matching, and an uncontrolled client without pacing.

Reproducibility. The simulator is deterministic given the seed, so every table in Section 6 can be regenerated exactly; changing the seed alters absolute counts but, by construction of the saturation signal, not the guarantee of Proposition 1. We deliberately avoid redistributing provider content: only synthetic places are used, which also removes any retention or attribution concern from the experiments themselves.

Table 2. Configuration parameters used in the evaluation

Parameter	Meaning	Value
Page size p / page limit L	Places per page / pages per query (K = p·L)	20 / 3 (K = 60)
Minimum cell size m	Smallest tile edge	0.05 km
Rate limit ρ; bucket (r, C)	Provider limit; refill rate and capacity	10 req/s; 8 req/s, 4
Maximum attempts	Retries before dead-letter	6
Weights (w_n, w_d, w_c)	Name, distance, category	0.5, 0.4, 0.1
Decay τ / cell size / threshold θ	Distance penalty / blocking / link threshold	60 m / 250 m / 0.45
Prior (μ, m)	Global mean rating, prior weight	4.1, 25
Unit costs (c_{ids}, c_{rich}, c_{detail})	Illustrative relative costs	1.0, 3.0, 2.0

6. Results

Absolute figures are a demonstration of algorithmic behaviour under stated assumptions, not measurements of any live service; Section 7 lists the validity limits.

6.1 Coverage and request efficiency

Table 3. Coverage and requests for different acquisition strategies (N = 5,000)

Strategy	Requests	Coverage	Note
Single query, whole region	3	1.2%	Capped at 60 results
Uniform grid 4 × 4	48	19.2%	Too coarse
Uniform grid 8 × 8	168	61.4%	Too coarse
Uniform grid 16 × 16	391	98.5%	Misses dense cells
Uniform grid 32 × 32	1035	100.0%	Complete but wasteful
Proposed AQTS	**516**	**100.0%**	175 leaf tiles

AQTS reached complete coverage with 516 requests, whereas the smallest complete uniform grid (32 × 32) needed 1,035, so AQTS used 50.1% fewer requests for the same result (Fig. 2). The 16 × 16 grid used fewer requests (391) but left 1.5% of places undiscovered; moreover, a practitioner cannot know in advance that 16 × 16 is nearly sufficient, because that depends on the unknown density. The resolution AQTS finds is used during runtime, creating small tiles in high density areas and large tiles in low density areas. The observed tree agrees with Proposition 2: 175 leaves means that I = 58 and T = 233; the observed number of requests, 516, is within the range $233 \leq R \leq 699$.

The efficiency is due to two features. First, the cap acts as a free density probe: a tile whose number of unoccupied places is below K is certified complete at no additional cost, and thus sparse regions are spanned by a few large tiles (such as the root, 20 km × 20 km, which only needs a few requests for which there is no cluster). Secondly, subdivision is local: a dense cluster causes a deep subtree, while its neighbours form shallow subtree, which means that requests are not wasted on empty land as they are in the 32 × 32 grid, where 1,035 requests are made for 1,024 cells, yet each request will ask an empty cell. The disadvantages of this adaptivity are that coverage increases more slowly at the beginning of a run (Section 6.3) since the first requests are made at coarse levels..

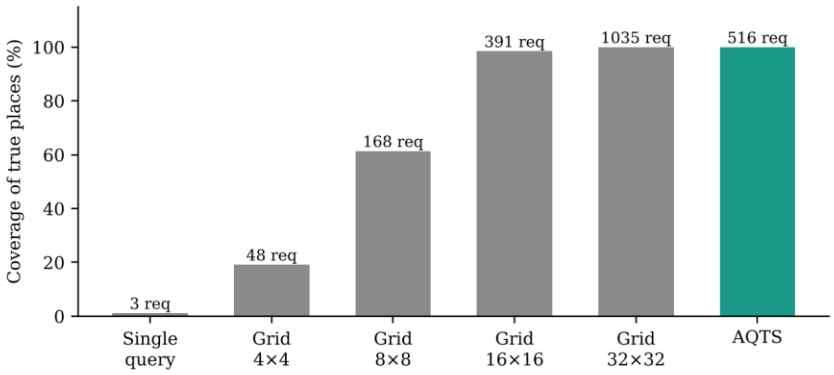


Fig. 2. Coverage and requests for uniform grids and AQTS.

6.2 Ablation: when to split

We varied the policy so that a tile is split after fetching one, two or all three pages (Table 4). All policies reach 100% coverage, so the choice affects only cost. Fetching all pages before splitting needs 516 requests, 33.2% fewer than splitting after the first page (773). The derived columns confirm the structural identity of Proposition 2: with one request per tile the split-after-page-1 policy has $R = T = 773$ exactly, and the mean requests per tile rise to 1.49 and 2.21 as more pages are fetched while the tree shrinks.

Table 4. Effect of the number of pages fetched before the split decision

Policy	Requests	Leaves	Tiles T*	Req./tile*	Coverage
Split after page 1 if full	773	580	773	1.00	100.0%
Split after page 2 if full	590	298	397	1.49	100.0%
Fetch all pages, then split (A1)	516	175	233	2.21	100.0%

6.3 Behaviour under a fixed request budget

Fig. 3 shows coverage achieved within a fixed budget, comparing AQTS with the best uniform grid that fits the budget. With 100 requests, a partially scanned coarse grid found more places (37.3%) than breadth-first AQTS (28.6%) because AQTS spends its first requests on coarse tree levels. From about 500 requests, AQTS matches or exceeds the best grid and at 1,000 requests is the only strategy to reach 100%. This identifies a limitation: for budgets below roughly 20% of the requests needed, a best-first variant that expands the tile with the highest expected yield would be preferable.

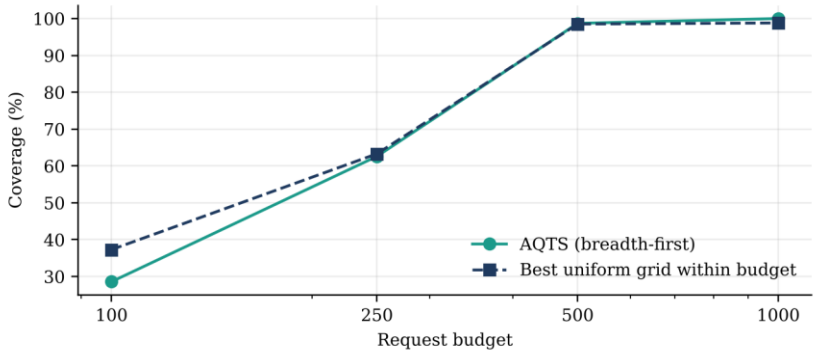


Fig. 3. Coverage within a fixed request budget.

6.4 Cost-aware planning

With R = 516, N = 5,000 and illustrative relative unit costs of 1.0 (IDs-only request), 3.0 (rich-field request) and 2.0 (detail lookup), Eq. (3) gives s* = 516·2 / (5000·2) = 10.3%. Below this share of places needing enrichment, IDs-only discovery plus selective details is cheaper; above it, rich-field search wins (Fig. 4, Table 5). In a monthly refresh in which about 3% of places are new, a full rich-field re-crawl costs 1,548 units whereas the planner's choice costs 816 units, a 47.3% saving. Unit costs are illustrative ratios; real savings depend on the provider's price list, which the planner reads from configuration.

Table 5. Relative cost by retrieval mode and enrichment share

Share s	Rich-field	IDs-only + details	Planner choice
100%	1548	10516	Rich-field
50%	1548	5516	Rich-field
10%	1548	1516	Two-phase
3%	1548	816	Two-phase
1%	1548	616	Two-phase

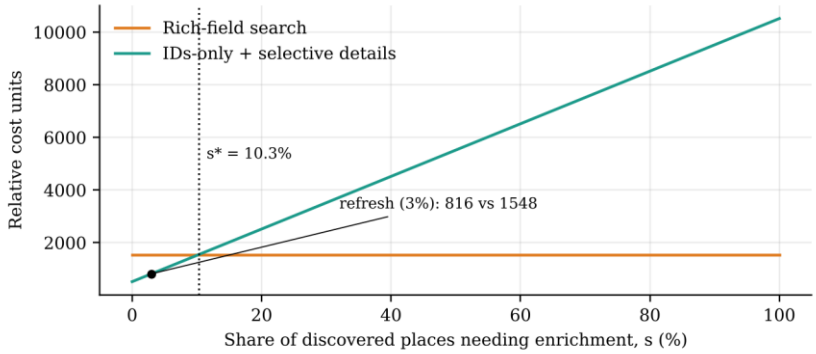


Fig. 4. Relative cost of the two retrieval modes versus enrichment share.

6.5 Policy governor

The same 516 tile requests were issued against a simulated service allowing 10 requests per second. An uncontrolled client was shot at approximately three times the allowed rate, and terminated a request after two attempts, followed by a repeat shot with a jittered back-off (Table 6). Without governance, 170 of 516 tile requests were lost, leaving about 67.1% of the region uncovered (using lost tiles as a proxy for coverage). The elapsed time was 34.9 s for the governor and 64.1 s for the governed client; 578 was the number of tries and 516/8 ≈ 64.5 s is the elapsed time implied by the refill rate. A longer, but full, run is definitely better for data collection purposes than a fast, but short, run.

Table 6. Request outcomes with and without the policy governor

Metric	Uncontrolled	With governor	Change
Requests completed	346 / 516	516 / 516	+170
Success rate	67.1%	100.0%	+32.9 pts
Requests dropped (lost tiles)	170	0	
Retries	578	1	

Elapsed time (simulated s)	34.9	64.1	+29.2 s

6.6 Entity resolution

The official source contained 5,000 records, while the open source contained 3,525, with 3,225 of those being actual links. Matching strategies are compared in Table 7 and Fig. 5(a). The composite score went from 39.4% (distance only) to 62.1% (name only), and was raised to 96.2%. Distance-only matching will also give false matches to businesses that are near each other but not the same name, and will overlook businesses that are the same name but have a different location; name-only matching will be accurate but will miss businesses with nearby names but different locations, as well as businesses with no name but the same location. The complementary signals were fused together using a composite function, which generated just 33 false links among 3,050 accepted (precision 98.9%, recall 93.6%). The remaining false links are similar to the chain-branch effect mentioned in Section 4.5..

Table 7. Entity-resolution accuracy for alternative scoring strategies

Method	**Precision**	**Recall**	**F1**
Distance only (≤ 100 m)	78.1%	26.4%	39.4%
Name only (≤ 250 m)	99.8%	45.1%	62.1%
Distance + category	90.8%	42.3%	57.7%
Proposed composite (A4)	**98.9%**	**93.6%**	**96.2%**

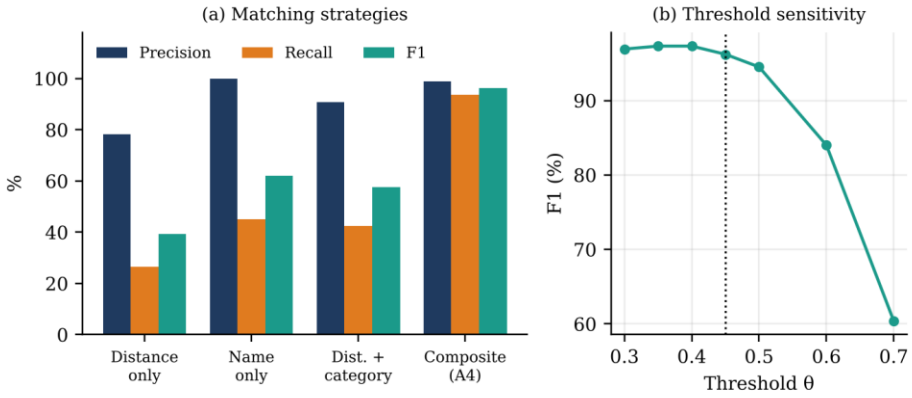


Fig. 5. (a) Precision, recall and F1 of matching strategies; (b) F1 versus acceptance threshold θ.

F1 is plotted as a function of θ in Fig. 5(b). This method was found to be insensitive to the exact value in a wide range of values, with F1 between 96.2% and 97.3% for values from 0.30 to 0.45, and performance deteriorating rapidly above 0.60 (94.0% and 60.4% at θ = 0.60 and 0.70, respectively). θ = 0.45 was chosen, as it offered a slight preference for precision over recall. The dataset had 5,475 entities (all 5,000 official places and 475 open-only places), 3,050 of which were confirmed by both sources. These numbers are optimistic for real data since θ was chosen from the same synthetic data.

6.7 Quality scores and review analytics

Table 8 illustrates Eq. (6) with μ = 4.1 and m = 25. The difference between the two averages goes from raw ranking (A, B, C, D, E) to (C, D, B, A, E), and the difference in the averages are better; the three-rating 5.0 drops 0.80, but the 1,270-rating 4.4 drops just 0.01. The mean value of DQI for entities confirmed by 2 sources was 0.844, compared to a mean of 0.743 for entities confirmed by only one source, thus revealing that the difference is the agreement component and simulated attribute availability, and is not actually a finding about real data.

Table 8. Effect of rating smoothing on example places

Place	Raw rating	Ratings n	Smoothed	Change
A	5.0	3	4.20	−0.80
B	4.8	12	4.33	−0.47
C	4.6	410	4.57	−0.03
D	4.4	1270	4.39	−0.01
E	3.9	56	3.96	+0.06

To demonstrate how the aspect analyser works, four illustrative reviews were written and then analysed. The food was very tasty but the service was slow and the prices were high translates to food positive, service negative and price negative, which separates opposing opinions in one sentence; Seating was tight and loud, but the food was not bad translates to food positive, service and price negative, negation handling. This is not a proof of accuracy, but a demonstration: the accuracy needs to be checked against an annotated corpus like the one of [4].

6.8 Summary of findings

Table 20. Summary of principal results

Component	Principal result
AQTS (A1)	Full coverage with 516 requests; 50.1% fewer than the cheapest complete uniform grid (1,035)
Split policy	Fetching all pages before splitting saves 33.2% of requests relative to splitting after page 1
Planner (A2)	Break-even s* = 10.3%; 47.3% saving for a refresh with 3% new places
Governor (A3)	
Entity resolution (A4)	F1 96.2% versus 39.4% (distance only) and 62.1% (name only)
Quality and sentiment (A5, A6)	Reliability-aware ranking; aspect polarity with negation handling (demonstrated, not benchmarked)

7. Discussion

7.1 Implications and compliance by design

The main conclusion is that there is no conflict between responsible design and technical performance. Where places are dense, spending request effort to get good coverage per request, and where the proposed bounds make the cost predictable before a job starts, which enables M1 to refuse jobs whose pilot-estimated number of requests exceeds the day's budget. The other point the governor makes is that, if you treat rate limit as a "design constraint", then you're making it a property that can be audited. The mechanisms are directly correlated to compliance: Only connector interfaces are permitted, and there is no page rendering or parsing component, so compliance is achieved by rate limits: pacing and back-off, spending: by pilot estimate, budget check, planner, retention: by per-attribute time-to-live and scheduled purge, accountability: by an append-only audit log, and reviewer privacy: by only keeping an aspect count and no logs, and credential security: by an encrypted configuration and masked logs. These mechanisms are in the request path, so compliance is not dependent upon user discipline. Provider terms and conditions still need to be reviewed by operators as the retention and attribution criteria are different and evolve over time.

Table 9. Scraping-based acquisition versus GeoTileIQ

Aspect	Browser-scraping pipeline	GeoTileIQ
Legality	Conflicts with platform terms; legal and reputational risk	Authorised interfaces only; governor and audit log

Coverage	Depends on scrolling behaviour; no guarantee	Tiling continues until no tile is saturated (Prop. 1)
Stability	Breaks on page-layout change	Stable structured responses
Cost control	None	Pilot estimate, budget check, break-even planner
Rate limits	Evasion attempts, blocking	Token-bucket pacing and jittered back-off
Integration	Single source, duplicates	Multi-source fusion, one-to-one matching
Trust	None	Data Quality Index and smoothed rating
Auditability	None	Complete request and policy log

Alternatives to quadrant splitting include k-d style median splitting (uses an estimate of the spatial distribution before asking a question) and density-weighted grids, but AQTS does not require an estimate of the spatial distribution; its primary practical advantage over the alternatives is that when the sole source of density information is the interface, then AQTS does not need it. However, AQTS always divides into 4 equal children, so it can also develop more finely than an AQTS median split in highly anisotropic situations (linear commercial corridors, for example), which is an extension of the AQTS simulation.

7.2 Practical guidance for deployment

There are three parameters that determine behaviour. The cap K is a fixed value, determined by the provider, but as can be seen from Proposition 2, with decreasing K the lower bound on leaves will increase, so operators should assume that their costs will increase roughly as they decrease K. The smaller the cell size m, the more complete the data will be in very dense areas, but at the cost of potentially unlimited subdivision; this size is set at 0.05 km for the evaluation and very small leaves, limited by resolution in this instance, should be audited and, if there are many, further subdivided with additional keyword or category partitions. The budget check in M1 should be based on a pilot of root and some of the sub-tiles to get a preliminary estimate of R before spending starts. For refresh cycles, the operators should redo the computation of s based upon the current prices and the expected fraction of new places as it is shown in Eq. (3), cheaper detail-lookups and a higher rate of changed places affect the break-even point.

7.3 Threats to validity and limitations

- **Simulated environment.** The service and data is man-made. The order of the results, the cap, the spatial patterns and the naming patterns of real services can differ. Conclusions must be drawn with validation against an authorised (live) interface.
- **Ranking independent of location.** Prominence is assumed to be unrelated to position in the simulation; in reality, prominence may be correlated with density, and Assumption A1 may not be precisely true for each provider.
- **Illustrative costs.** The unit costs used are relative ratios selected to show Eq. (3) and not the current costs.
- **Threshold tuning.** The matching threshold is set based on the evaluation data; it needs to be tuned before deployment on a separate labelled data.
- **Low-budget behaviour.** If the budget is less, e.g. < 20%, than the number of requests needed (Fig. 3), breadth-first AQTS is not optimal.
- **Sentiment evaluation.** However, aspect analysis does not get benchmarked, and only trained transformers do outperform lexicon methods on abstract aspects [4]. For a single run, the results are given with a fixed seed and do not include variance across random seeds.

7.4 Ethical and legal considerations

A compliant path is the only path in the framework and compliance is applied to the deployment. GeoTileIQ represents the rules as configuration and time-to-live rules and a purge sweep, but it is not able to determine what rules are applicable to each specific licence. The review of text is done only where the licence permits, the identity of the reviewer is never recorded, and only the aggregates and allowed identifiers should be published in the published outputs. Last but not least, an acquisition method that reduces the price of acquiring regional datasets should not be interpreted as a carte blanche to violate quotas and/or retention policies of the provider; the audit log is there for the sole purpose of ensuring that the administrator can show that quotas and/or retention limits were observed.

7.5 Generalisability

While the model evaluates one specific interface, the framework only uses three properties: bounded result cap, field-dependent pricing and request-rate limit. A connector can be used, without altering M2–M8, to wrap any provider that provides a location-restricted search with these properties, such as an open service that's queried by bounding box. The saturation signal is also passed to non-spatial partitions: If a tile is saturated at the minimum cell size, the same logic of splitting on saturation can be applied to any other partition, such as category, keyword or opening status, and thus extend AQTS from purely spatial to multi-dimensional decomposition. Similarly, the quality index and smoothed rating are source-agnostic and can score records from public or government data sources, as in conflation work like [7]. These extensions are speculative and based on the design and not on the results of this study and would require their own validation.

8. Conclusion and Future Work

This paper reformulated place-data acquisition from a scraping task into a constrained, measurable engineering problem and presented GeoTileIQ, a compliance-first framework that builds integrated, quality-scored place datasets from authorised sources. The advantages of AQTS are that it is able to split only saturated tiles, provides a completeness proposition and request bounds, and in a simulation, it achieves full coverage of the grid in approximately half of the number of requests as the costliest complete uniform grid. The closed-form rule of the planner improved the refresh cost almost by half, the governor removed the lost tiles in the rate limit, and the distance-only matching retrieved an F1 score of 39.4% while the composite matcher achieved an F1 score of 96.2%, and the quality index and smoothed rating make the record reliability explicit. Future plans involve (i) live validation in a pilot district under the provider's terms, with multiple seeds and confidence intervals; (ii) best-first tiling, expanding their tiles according to the expected yield, to improve the performance of small budgets; (iii) learned or graph-based matching trained with labelled links [3], [7], [9]; (iv) multilingual aspect analysis using a transformer [4], [5]; (v) change detection and closure prediction from periodic discovery with IDs only; and (vi) privacy-preserving analytics for published aggregates [11].

8.1 Declarations

Data and code availability: the study uses synthetic data generated with a fixed seed; no provider content is redistributed. [Add repository link.] **Funding:** [state or “none”]. **Conflict of interest:** [state or “none declared”]. **Ethics:** no personal data about reviewers are collected or stored.

REFERENCES

1. T. Kim, K. Bock, C. Luo, A. Liswood, C. Poroslay and E. Wenger, “Scrapers selectively respect robots.txt directives: Evidence from a large-scale empirical study,” in *Proc. ACM Internet Measurement Conf. (IMC ’25)*, Madison, WI, USA, Oct. 2025, doi: 10.1145/3730567.3764471.
2. S. Longpre *et al.*, “Consent in crisis: The rapid decline of the AI data commons,” in *Proc. Conf. Neural Information Processing Systems (NeurIPS)*, 2025.
3. H. Amiri, M. Hashemi and A. Züfle, “World-POI: Global point-of-interest data enriched from Foursquare and OpenStreetMap as tabular and graph data,” arXiv:2510.21342, Oct. 2025.
4. S. Zaid, A. H. Alharbi and H. Samra, “Multi-aspect sentiment classification of Arabic tourism reviews using BERT and classical machine learning,” *Data*, vol. 10, no. 11, Art. 168, 2025, doi: 10.3390/data10110168.

5. A. Nadeem *et al.*, “Resolving ambiguity in natural language for enhancement of aspect-based sentiment analysis of hotel reviews,” *PeerJ Comput. Sci.*, vol. 11, e2635, 2025, doi: 10.7717/peerj-cs.2635.
6. A. Bellogín, L. W. Dietz, F. Ricci and P. Sánchez, “Point of interest recommendation: Pitfalls and viable solutions,” arXiv preprint, Jul. 2025.
7. C. Bertozzi, “Matching and conflation of open government data with OpenStreetMap data,” M.Sc. thesis, OST – Eastern Switzerland Univ. of Applied Sciences, 2025.
8. R. Low, Z. D. Tekler and L. Cheah, “An end-to-end point of interest (POI) conflation framework,” arXiv:2109.06073, 2021.
9. T. Novack, R. Peters and A. Zipf, “Graph-based matching of points-of-interest from collaborative geo-datasets,” *ISPRS Int. J. Geo-Inf.*, vol. 7, no. 3, Art. 117, 2018, doi: 10.3390/ijgi7030117.
10. L. Li, S. Hu, A. Shaw and L. Hemphill, “Crowdsourcing public attitudes toward local services through the lens of Google Maps reviews: An urban density-based perspective,” arXiv:2404.13156, 2024.
11. Q. Zhang *et al.*, “A survey on point-of-interest recommendation: Models, architectures, and security,” arXiv:2410.02191, Oct. 2024 (rev. Mar. 2025).
12. Google Maps Platform, “Text Search (New) – Places API,” developer documentation. [Online]. Available: https://developers.google.com/maps/documentation/places/webservice/text-search (accessed Nov. 4, 2026).
13. R. A. Finkel and J. L. Bentley, “Quad trees: A data structure for retrieval on composite keys,” *Acta Informatica*, vol. 4, no. 1, pp. 1–9, 1974.
14. I. P. Fellegi and A. B. Sunter, “A theory for record linkage,” *J. Amer. Statist. Assoc.*, vol. 64, no. 328, pp. 1183–1210, 1969.
15. W. E. Winkler, “String comparator metrics and enhanced decision rules in the Fellegi–Sunter model of record linkage,” in *Proc. Section on Survey Research Methods, American Statistical Association*, 1990, pp. 354–359.
16. M. Brooker, “Exponential backoff and jitter,” AWS Architecture Blog, 2015.