The Living Map: A Location Knowledge Graph Maintained by Confidence-Gated LLM Agents for Property Search

Filipe Brito Ferreira | Independent researcher | https://www.fbritoferreira.com

July 2026

Abstract. Property search depends on knowing where things are, and in fast-growing cities that knowledge decays weekly. Administrative hierarchies differ per country; place names are ambiguous across entity types; boundaries are social conventions rather than facts; the ground truth moves as developments launch, rebrand, and dissolve. Location errors are uniquely expensive because users can personally verify them: a buyer knows the tower they toured is not in the community the listing claimed.

This paper presents the design of a living location knowledge graph that replaces the manually curated location database: per-market typed entities assembled from open map data, developer sources, and launch brochures (including locations that exist commercially before any map has them), maintained by a cascade of LLM research agents whose every conclusion carries a source-derived confidence score. The central discipline is a three-region decision gate in the tradition of Fellegi–Sunter record linkage: auto-accept above an upper threshold, discard below a lower one, and route the band between to human review with evidence attached. Automation earns autonomy per decision and per confidence level. Components of this design operate in production; operational metrics are withheld under employer confidentiality, and the paper states which components run and which remain design intent. In their place, the paper includes a fully reproducible public-data pilot of the decision gate on OSM Dubai: 322 entities routed, with zero false accepts in a 40-item audited sample, bounding the accept-region error below ~7.5% at 95% confidence (Section 6.1). We describe the data model, the trust and confidence machinery, the failure modes the field imposes on the design, and the evaluation contract a system of this class owes its operators.

For decision-makers.

- *The problem:* location errors corrupt search, price statistics, and valuation models simultaneously, and users personally catch them, taxing platform credibility, while manual location curation scales linearly with market growth.
- *The proposal:* a typed location graph maintained by LLM research agents under a confidence gate, with humans reviewing only what the machine cannot settle; new developments enter at announcement time, ahead of any map.
- *What runs today vs. what is design:* Section 4.5, Table 2.
- *Adoption path and order:* Section 4.6, which puts entry-time listing validation first; it defends revenue immediately and needs only the location data you already have.
- *An explicitly hypothetical sizing sketch* (replace every number with your own): a portal with 100k listings, a 5% misplacement rate, and manual curation at 15 minutes per verified location is carrying ~5,000 wrong-community listings into search results daily and paying roughly two curator-FTEs per fast-moving market; the cost sketch of Section 4.4 shows tokens are not where the economics live (the review band and amortized engineering are), and only your own audited precision figure, from Section 6's scorecard, decides the business case.

- *The governing rule:* automation earns autonomy per decision; everything else is a recommendation to a human, with evidence.

---

1. Introduction

Ask a property portal a simple question (*show me apartments in Dubai Marina*) and you have asked one of the hardest questions in real-estate search. Not because search engines are bad at matching strings, but because *where a property is* turns out to be surprisingly difficult to know, and increasingly expensive to be wrong about. In a fast-growing market, the entities that make up "location" (areas, communities, buildings, developments) are created, renamed, merged, and dissolved at a pace that manual data curation cannot match, while a meaningful share of listing inventory carries some form of location error (a practitioner observation Section 3.6 returns to).

The industry's dominant answer is manual: curation teams updating location records by hand, anchored by geocoding of varying quality. Manual curation cannot keep pace with a city that changes weekly, does not scale across markets, and does nothing about query-time ambiguity: it patches the database that queries run against without addressing why the database keeps going stale. Geocoding services, meanwhile, embody an assumption this problem violates: that a location reference is an *address*. In markets without standardized addressing, and for entity types such as developments and developers, converting a string to a coordinate is the mechanical part of the work; the hard remainder is deciding *which typed entity* the string meant, which is a resolution problem.

This paper describes a different answer, a **living location knowledge graph**: per-market graphs of typed location entities, assembled from open map data, commercial map data, developer sources, and launch brochures, and maintained continuously by a cascade of LLM research agents. The system is designed around two convictions. First, location truth is a moving target, so keeping it is a standing capability rather than a cleanup project. Second, LLM output is never privileged truth: every fact entering the graph carries a confidence score derived from the trust of its source, and a three-region decision rule (after Fellegi–Sunter [9]) routes everything below an autonomy threshold to human review, with supporting evidence attached.

Contributions. This paper's contributions are named, testable design mechanisms:

1. **A working formulation of the location problem** as *typed entity resolution over a moving ground truth* (formalized in Section 2.2), with the design requirements it imposes.
2. **Per-facet confidence with fact-type-split source trust:** an entity can be semantically strong and geographically weak, and a source can be authoritative on identity while adversarial on location (Sections 3.1, 3.3). This is what makes pre-map locations representable and the announced-meets-built merge survivable.
3. **The monotone-demotion reviewer contract with mechanical evidence grounding:** a reviewer that can only lower confidence, fed and audited on grounded evidence rather than generated rationales (Sections 3.4–3.5).
4. **Single-lineage geometry as a joint engineering and licensing invariant:** one polygon, one source, one license, per claim, extended to derived spatial relations (Sections 3.8, 4.2).
5. **An evaluation contract** for confidence-gated curation: both error regions audited (precision of the accept region, seeded-error recall of the miss region), calibration per fact-type stratum, and economic units that make the automation claim falsifiable (Section 6).

Which of these is the thesis. Two of the five carry the paper's central claim. The first is the **fact-type split of source trust with per-facet confidence** (contribution 2), the mechanism that lets a developer be simultaneously authoritative on identity and adversarial on location, and therefore lets locations exist in the graph before any map has them. The second is the **evaluation contract that makes confidence-gated automation falsifiable** (contribution 5). The remaining three are enablers: the formulation (1) frames the problem, and the reviewer contract (3) and geometry lineage rule (4) keep the thesis mechanisms honest under LLM noise and license law respectively. Section 7 reaches the same division from the other direction:

of everything here, the pre-map channel and the fact-type trust split are the pieces with no close published analogue; the rest is disciplined combination.

A note on evidence. This design is not hypothetical: versions of the components described here operate in production, and Table 2 (Section 4.5) states plainly which ones. Employer confidentiality prevents publishing operational detail (scale, metrics, thresholds, costs); what the constraint buys is the employer's, not the author's, to spend. Following the established practice of industrial systems papers that withhold operationally sensitive detail, the failure modes of Section 5 are therefore described as patterns and composites rather than incidents, and every number the paper does publish is fully public and reproducible: the OSM coverage measurements of Section 3.8 (Appendix A) and the audited decision-gate pilot of Section 6.1. Section 6 specifies the evaluation a deployment of this design must provide, precisely because this paper cannot publish one externally.

Roadmap. Sections 2–6 move from the problem and its requirements through the design, implementation, production lessons, and the evaluation framework. Section 7 situates the work against the gazetteer, geoparsing, entity-resolution, and industrial place-system literatures; Sections 8–10 cover future directions, limitations, and conclusions, Appendix A gives the reproducibility material for the public measurements, and Appendix B develops a secondary consequence of the design, the graph as an organic-search asset.

---

2. Background and Requirements

2.1 Why location is hard

Hierarchies do not standardize. Every country decomposes space differently, and most decompose it inconsistently even within themselves. Dubai is the instructive case: an emirate that is also a city. It contains no smaller cities; instead it breaks into *areas*, and a single area can be known by more than one name. There is no postcode system to anchor an address, and no canonical gazetteer (a reference directory of places, their names, and their identifiers) to defer to. A platform operating across countries cannot assume `country → state → city → neighborhood → street`; each market has its own decomposition, naming habits, and gaps.

It is tempting to assume this is a maturity problem that official addressing eventually fixes. Dubai in fact operates one of the more ambitious official systems anywhere: Makani, a 10-digit geo-address assigned to every building entrance and used by emergency services and utilities [8]. Makani answers "where exactly is this entrance", a coordinates problem. It says nothing about whether that entrance is *in* JBR (Jumeirah Beach Residence) or Dubai Marina, which development it belongs to, who built it, or what people call it colloquially. Search runs on that semantic layer, and no government issues identifiers for semantics. Postcode countries exhibit the same gap in milder form: a UK postcode anchors delivery, not the boundary of what locals consider the neighborhood. Official addressing, where it exists, solves logistics. It does not solve *place*.

The same name means different things. Location names in real estate are overloaded across entity types. "Meydan" in Dubai plausibly refers to an area, a specific development, or the developer behind it; a user typing "Meydan" may want any of the three and may not know which they mean. A user searching "MBR District One by Meydan" wants a development; a user searching "Meydan" probably wants the area containing it. Collapsing these into one string match produces confidently wrong results. The general form of the problem, and the formulation this paper builds on, is that **a location query is an ambiguous reference to an entity of unknown type**, and resolving it requires context the query string does not carry. This is the classic toponym-resolution problem [14, 15] with an additional axis: the candidate entities differ not only in position but in *kind*.

The ground truth moves. Cities like Dubai change at a pace that breaks static data. New towers launch; projects are renamed mid-construction; developments are cancelled; area boundaries shift as districts mature. A location database that was correct last quarter is wrong today. Any solution therefore has to be a system for tracking a moving target; a data-cleaning exercise cannot keep up.

Boundaries are social conventions. Even with coordinates in hand, assignment stays ambiguous. Adjacent communities sit directly against each other and their de-facto boundaries overlap; a tower on the seam between JBR and Dubai Marina is claimed by both. Coordinates cannot settle a question that is partly social: which community a building "belongs to" is decided by how residents, agents, and marketers talk about it, and they do not agree. Compounding this, colloquial *area* names and official *community* names diverge, so the name a person searches is often not the name in any registry. That real-estate boundaries are manufactured rather than found is not a new observation: Zillow had to hand-build and publish neighborhood boundary files for US cities in 2008 because no authoritative source existed [44], and Airbnb employed a cartographer to draw its neighborhood geometries by hand [43].

All of this lands on the listings, where the forces above compound. Listings get attached to the wrong community (often because two communities have similar names, sometimes deliberately; Section 3.6), and the error is invisible at entry time yet expensive forever after: it pollutes search results, corrupts area-level price statistics, and misleads valuation models that aggregate by location. Location errors sit upstream of nearly everything a property platform computes. And location trust is unusual among data-quality problems in that users can *personally verify* the failure: they know where they live, and they know the tower they toured is not in the community the listing claimed. Every error a user catches is a credibility loss no ranking adjustment recovers.

The same graph that fixes the errors also compounds on the demand side: location pages assembled from verified, per-location data are structurally unique, update whenever the graph does, and exist for announced developments while the location exists nowhere else, an organic-search position competitors cannot template their way into (Appendix B develops this). All five forces reduce to one sentence the rest of the paper depends on: location in property search is *typed entity resolution over a moving ground truth*, and geocoding is only the mechanical part of it.

2.2 Requirements

The problem statement above yields five requirements that drove every design decision in this paper, plus two explicit non-goals.

Per-market hierarchies (R1). Model each country's actual decomposition of space. No universal schema: the moment the schema lacks a slot for the thing locals search by (*development*, in the Gulf), the system either abuses an existing type or loses the entity.

Typed entities (R2). "Meydan" the area, the development, and the developer are three nodes, not one string. Resolution is ranking over typed candidates.

Continuous change absorption (R3). New, renamed, and dead locations must flow in from many sources as routine operation, not as cleanup campaigns, including locations that exist commercially before any map has them.

No silent wrong data (R4). Anything uncertain is flagged, reviewed, or visibly correctable. False confidence is the worst failure mode, because users price it into their trust of the whole platform.

Sublinear curation (R5). Adding a market or a thousand new buildings cannot mean hiring proportionally more curators. This is an economics requirement, and Section 6 treats it as one.

Non-goals. The system does not attempt to *decide* socially disputed boundaries; its ceiling is representing the dispute (Section 3.8). And it does not replace human judgment: it rations it, spending reviewer attention only where the pipeline is genuinely unsure.

The problem, formally. Stated as a definition the rest of the paper can be checked against: given a universe of typed location entities E whose membership, attributes, and geometry change over time, a set of evidence channels S each with unknown reliability, and a stream of references q (queries, listings, documents) each denoting some $e \quad E$ of unknown type, the system must (i) maintain an estimate of E and its relations from S, (ii) resolve each reference to its entity, and (iii) route every automated conclusion into {accept, review, reject} such that, for chosen thresholds, the expected error of the accept region and the human cost of the review region are both bounded. Part (iii) is the decision problem Fellegi and Sunter formalized for record

pairs [9]; this system applies the same decision *shape* to heterogeneous facts, with the honest differences from the original formalism spelled out in Section 3.4.

2.3 Design principles

Three principles recur throughout the design. **Trust is a property of sources; facts carry their own confidence** (Section 3.4). **Automation earns autonomy per decision, per confidence level**: everything else is a recommendation to a human. And **every entity decision taken automatically must be reversible**: creation implies demotion paths, merging implies unwinding, and identifiers outlive the operations performed on them.

Table 1 summarizes the problems the design addresses and the techniques it answers them with.

Table 1: Summary of problems and techniques.

Problem	Technique	Advantage
Hierarchies differ per market (R1)	Per-country typed graphs	Market-exact schema; no lowest-common-denominator model
Names ambiguous across entity types (R2)	Typed nodes + alias edges; context-aware resolution	Resolution becomes ranking over typed candidates
Ground truth moves (R3)	Multi-channel research cascade + reviewer agent	Change absorbed as routine, cheap sources first
Locations exist before maps (R3)	Developer/brochure pre-map channel	Node live at announcement; demand captured day one
LLM and source noise (R4)	Source-derived trust; three-region decision gate	Hallucination routed to humans, not into the graph
Correlated sources	Provenance deduplication before agreement scoring	Press-release echoes counted once
Adversarial location claims	Per-lister trust; entry-time correctness scoring	Prestige misplacement caught at the gate
Boundary disputes	Per-source claim polygons (single lineage each); impact-gated adoption	Geometry changes gated on what they flip
Curation cost (R5)	Confidence-gated automation + human review band	Reviewer attention spent only on genuine ambiguity

3. Design

This section presents the graph itself (3.1), how it is fed (3.2–3.3), how it decides what to believe (3.4–3.5), how it defends the inventory layer (3.6), how queries resolve against it (3.7), and how it handles geometry (3.8).

3.1 Data model

Each market is its own graph with its own decomposition (for the UAE, `emirate → area → community → building`) rather than an instance of a universal schema. This is a deliberate rejection of the global-gazetteer instinct: GeoNames [1] and Who's On First [2] pick one placetype ontology and fit every country into it, which suits a global reference dataset but not search inside a specific market. Per-country graphs cost more to design; the payoff is exactness in the queries.

Nodes are typed as *area, community, building, development, developer,* and *POI* (point of interest: cafés, schools, hotels); edges express *contains, alias-of, developed-by,* and *adjacent-to* relationships. An area, a

development, and a developer are distinct entities even when they share a name, which is what makes references like "Meydan" resolvable: the query maps to candidate nodes of different types, and resolution becomes a ranking problem over typed candidates rather than a string match (Figure 1).

```
graph TD
    UAE[UAE · country] -->|contains| DXB[Dubai · emirate/city]
    DXB -->|contains| MAR[Dubai Marina · area]
    DXB -->|contains| JBR[JBR · area]
    DXB -->|contains| MEY[Meydan · area]
    MAR -->|contains| TWR[Tower X · building]
    JBR -.->|adjacent-to| MAR
    MEY -->|contains| MBR[MBR District One · development]
    MEYDEV[Meydan · developer] -->|developed-by  | MBR
    MEY -.->|alias-of| MEY2[Meydan City]
    TWR -->|near| POI[Café · POI]
```

Figure 1: The typed location graph. Area (blue), development (green), and developer (red) are distinct entities that share a name; alias and adjacency edges carry the relationships string matching cannot.

Nodes are rich objects rather than labeled points. A hotel POI carries room count, room types, and accepted payment methods; a café carries a different attribute shape. Nodes also carry user-generated attachments (reviews above all), and those are typed too: what users rate about a community (schools, noise, walkability) is not what they rate about a tower (elevators, gym, management), a distinction whose consequences Section 5.1 reports. The graph's value compounds as nodes hold real attributes, because it stops being an address registry and becomes a model of *what it is like to live somewhere*, which is what property seekers are actually searching for.

Finally, facts carry **per-facet confidence**: a node can be simultaneously well-established semantically (its name, developer, unit mix) and weakly established geographically (no polygon, an approximate coordinate). Section 3.3 shows why this decomposition is essential.

3.2 The research cascade

When the system researches a location, it runs a cascade of progressively more targeted acquisition (Figure 2):

1. **OSM extract** [3]: POIs and whatever structured data the open map carries for the location.
2. **Web search**: news articles and announcements; the primary channel for detecting *new* locations, such as a tower announced but not yet on any map.
3. **Developer-targeted search**: if the location is attributed to a known developer, the developer's own site is searched for authoritative detail: unit mix, official naming, delivery dates.

Cheap broad sources run first; expensive targeted research runs only when there is a specific entity to chase. Each step feeds the LLM layer, which assembles the findings into a location profile.

```
graph LR
    A[1. OSM extract<br/>POIs + structured data] --> B[2. Web search<br/>news, announcements,<br/>new l
    B --> C[3. Developer-targeted search<br/>official site: unit mix,<br/>naming, delivery dates]
    C --> D[LLM layer<br/>assembles location profile]
    D --> RV[Reviewer agent<br/>compares all findings,<br/>cross-checks sources]
    RV --> E{Confidence<br/>gate}
    E -->|high| F[Auto-approve<br/>→ graph]
    E -->|low| G[Manual review queue<br/>+ LLM reasoning attached]
    G -->|approved| F
```

Figure 2: The research cascade. Broad sources run first; a reviewer agent cross-checks all findings before anything reaches the confidence gate; low-confidence conclusions route to human review with evidence attached.

The cascade does not feed the decision gate directly. Between them sits a **reviewer agent** that takes everything the previous stages produced and compares it, the way a group of researchers compares notes before publishing. Where sources agree, the composition's agreement weights raise confidence; where OSM says one thing and the developer's site another, the reviewer resolves the conflict: prefer the more trusted source, merge the compatible parts, or push the whole profile down in confidence so a human decides. Section 3.5 states the reviewer's contract precisely, because an unconstrained reviewer is a hazard of its own.

Fresh truth enters through more channels than the cascade: developer feeds and websites, launch brochures arriving by email (Section 3.3), OSM change monitoring, news coverage, and user submissions (a lister who cannot find their building can propose it). Every channel passes through the same validation layer: the system checks proposed changes rather than trusting any single source. That is the direct answer to the two root causes running through Section 2.1, slow updates and no single source of truth.

3.3 Locations that exist before the map

In a fast market, the most valuable moment to know about a location is *before any map has it.* An off-plan development (one sold before construction) exists commercially, with buyers committing money, months or years before it exists physically, and search demand for it starts the day it is announced. Waiting for open map data means being late for exactly the inventory that matters most to a new-projects business.

The pipeline therefore runs in reverse for announced projects. The entry point flips from map data to **developer sources**: the websites of major developers (in the UAE, household names such as EMAAR and Binghatti) are monitored for project announcements, and an announcement *initiates* graph-node creation rather than enriching an existing node. The research cascade then runs forward from that seed: web search for coverage and naming variants, and map checks that will keep coming up empty until construction appears on the ground.

The second pre-map channel is messier and richer: **brochures**. Developers distribute launch brochures by email as designed PDF documents (floor plans, unit mixes, payment plans, masterplan maps, delivery dates), often carrying detail no public webpage has yet. These arrive as unstructured marketing artifacts and are processed by the LLM layer into structured location facts: entity name and aliases, developer attribution, unit types, location within the parent area (Section 4.3 covers the extraction machinery). A brochure is, in effect, the developer telling you the future of the map in a format built for humans; extraction turns it into a graph node whose facts carry confidence derived from the developer channel's trust level, subject to the same reviewer and confidence gates as everything else.

The channel prior scores the *source*, but extraction is its own error-prone step: a misread unit mix must not inherit the developer's high trust wholesale. Extraction confidence is therefore **per-field**, not per-document: a clearly printed project name and a number read out of a floor-plan table are different facts with different confidence, even from the same PDF.

The deeper subtlety: **a source's trust must split by fact type, because reliability and incentive are different things.** A developer is the most authoritative source alive on its own project's name, unit mix, and delivery date, and at the same time a party with a direct commercial interest in its *location* claims, for the prestige reasons Section 3.6 describes. The brochure channel therefore carries a high prior on identity facts and a deliberately skeptical one on locational claims; per-facet confidence (Section 3.1) encodes this split.

Pre-map nodes are born with low geographic confidence (often no polygon; an approximate coordinate from a masterplan render) and high semantic confidence (nobody knows the project's name and unit mix better than its developer). As construction reality catches up (the project appears in OSM, coordinates firm up), the geographic facet fills in through the normal channels. The graph does not treat "announced" and "built" as different worlds; they are one entity at two confidence stages.

The announced-node-meets-built-node moment is the system's purest record-linkage decision [9, 11]: deciding that a brochure-born node and a newly mapped building are the *same entity.* The match evidence is the classic feature set with a domain twist: name and alias similarity, developer attribution, coordinate proximity weighted gently because the announced coordinate came from a marketing render. The merge

follows a **survivorship rule** (survivorship: which source's value wins for each field when records merge [11]) **that respects per-facet confidence**: semantic fields keep the developer-sourced values; geometry adopts the map-sourced values.

Identifier policy matters as much as match accuracy: the announced node's identifier survives the merge, because that identifier has carried a public location page since announcement day, and identifiers that break on merge break URLs. A wrong merge requires an unwind path for the same reason garbage entities require demotion (Section 5.2): entity decisions taken automatically must remain reversible. Candidate generation for this linkage has an unusually good natural **blocking key** (blocking: cheaply pre-filtering candidate pairs so the system never compares every node against every other [11]): developer attribution × parent area, refined by name n-grams, a scheme cheap to compute because developers and areas are already typed entities in the graph. Completeness is a separate question: the key's own fields are uncertain claims at announcement time, so blocking is multi-pass (a name-n-gram-only fallback pass catches candidates the primary key misses), and blocking recall is estimated with the same seeded-pair mechanism the evaluation contract already commits to (Section 6).

The 1-to-many case (one announcement becoming several towers) is the usual case in phased masterplans, and it has a natural shape: the announced node persists as a **parent project node** (type *development*), keeping its identifier and its URL history, while each physically materializing tower links in as a phase child under it, matched by the same evidence with phase designators ("Tower 2", "Phase 3B") as additional name features. Survivorship then runs per child (geometry from the map, semantics inherited from the parent unless the child's own sources override), and a *split* (the reverse operation, one node discovered to be several) reuses the demotion-and-redirect machinery of Section 5.2. The design is stated here; its hardening sits in the maturity table (Section 4.5) and the limitations (Section 9).

3.4 Trust and confidence

Two numbers run the system, so we define them once. **Sources have trust levels**: a standing prior reflecting how reliable a channel has proven; an official developer feed sits above a lone web mention. **Facts carry confidence scores**: derived from the trust level of the source that delivered them, adjusted by cross-source agreement and the reviewer layer. Trust belongs to channels; confidence belongs to facts. Every gating decision in this paper is one of these two numbers meeting a threshold. (Three nearby terms to keep apart: a **fact** is one claimed value, such as "Tower V's parent area is Meydan"; a **fact type** indexes the trust split, since identity facts and locational facts from the same source carry different priors; a **facet** is a per-entity dimension, where the same node's *semantic* facet can be strong while its *geographic* facet is weak.)

Every fact entering the graph therefore arrives scored by where it came from, not by how plausible it sounds. This is the system's hallucination defense: hallucination is a structural property of large language models rather than a bug better prompting removes [22], so the architecture assumes it. LLM-derived findings are just another source with their own trust level. A model inventing a building does not silently enter the graph with full confidence.

The formal ancestor is Fellegi–Sunter [9], the 1969 decision model that gives record linkage its canonical shape: evidence composed into a score, and a **three-region rule** (above an upper threshold, accept automatically; below a lower one, reject; between them, route to clerical review). The confidence gate in this system is that decision *shape* in modern form: auto-approve, manual queue, discard.

The inheritance is partial, and the boundary matters. Fellegi–Sunter's theorem (that the three-region rule minimizes clerical effort at fixed error rates) holds for scores built as likelihood ratios over match/non-match hypotheses; this system's score is not (yet) such a ratio, so the gate inherits the *decision structure*, not the optimality guarantee.

What makes the framing more than branding is the estimation path it opens: restate the composition as **additive log-weights per fact type** (a log-prior per channel, an agreement log-weight per provenance-independent corroboration, a bounded negative reviewer adjustment) and the weights become estimable quantities rather than settings. The calibration plan then splits cleanly by subproblem: for the *pairwise linkage* decision of Section 3.3 (announced node vs. built node, a true match/non-match problem), Winkler's

EM estimation applies directly and needs no labels [10]; for the *fact gate* at large, which has no latent match class, the right tool is supervised post-hoc calibration (Platt scaling for small adjudicated sets, isotonic regression for larger ones [31, 32]) fitted against human-review outcomes (Section 6). Until those steps land, the honest name for the score is an *ordinal routing priority*, not a probability.

The composition's form is not confidential, so here it is as an equation rather than prose. For fact f of type t from channel c:

```
score(f) =  ( w(c,t)                                  # channel prior, split by fact type
            + Σ  w(t, c ) · [corroboration i is provenance- and family-independent]
            - Σ  w_d(t, c ) · [contradiction j is provenance- and family-independent]   # disagreement
            - r(f) )                                    # reviewer demotion, bounded: 0   r   r_max
route(f) = accept   if score   _hi
           review   if _lo   score <  _hi
           reject   if score <  _lo
```

Some definitional notes on the terms. *The agreement weight* is indexed by the *corroborating* channel c : agreement from a low-noise channel is worth more than agreement from a lone web mention. This echoes, by analogy rather than inheritance, Fellegi and Sunter's principle that the weight of an agreement depends on what is doing the agreeing; their weights are log-likelihood ratios of the comparison field, not source-trust judgments, so the channel priors serve only as a heuristic initialization for these weights until adjudicated outcomes allow real estimation.

The disagreement term mirrors the agreement term: contradiction from an independent channel enters as its own negative log-weight *w_d*, indexed by the contradicting channel and subject to the same independence tests. The term earns its place in the equation because without it, a fact corroborated by one weak channel and contradicted by two strong ones nets positive, which is backwards. The reviewer's bounded demotion handles the conflicts the weights cannot see, and disagreement weights are estimable from the same adjudicated data as agreement weights.

The terms live in log-odds, and is the logistic squash, applied now rather than deferred to calibration. The reason is saturation. An additive composition clipped to [0, 1] pins every twice-corroborated fact at the ceiling, which destroys the ordering the gate routes on; the logistic keeps stacked evidence ordered at every level. Routing is unchanged because is monotone, and the worked example's numbers read as squashed outputs. Calibration later re-fits the same log-odds weights; it does not change the algebra. Categorical extraction-confidence levels (the fact record's `"high"`) map to fixed offsets on w per channel, and those offsets are **derived from measured per-field extraction accuracy on labeled samples per document class, never from model self-report**, since verbalized confidence is exactly the miscalibrated quantity the design refuses elsewhere [24, 25]. The weights w , w , the bound *r_max*, and the thresholds are the operational, still-to-be-fitted, part; the functional form is the estimable skeleton the calibration program targets.

One correction the naive version of "agreement raises confidence" requires: **sources must be independent to count.** In this domain they frequently are not: three news articles and an aggregator listing routinely all derive from the same developer press release, so counting them as four agreeing sources counts one source four times, with correlated errors dressed as consensus. Agreement between OSM and a developer feed is evidence; agreement among rewrites of one announcement is provenance. The pipeline deduplicates derivation chains before scoring agreement.

Two constraints on the corroboration term matter more than its weight. **Provenance-independence** (the dedup above) removes *derivation* correlation, but not *family* correlation: a developer's feed and the same developer's brochure are one party; open and commercial maps can share survey lineage. The corroboration bonus is therefore **capped at one per source family** (party or lineage class), the cheap defense against the conditional-independence assumption this additive form inherits from its Fellegi–Sunter ancestry; log-linear interaction terms, as in Winkler's non-independence extensions, are the principled upgrade if family-capping proves too blunt. And when a decision needs multiple facets (a merge reads both semantic and geographic facets), facets are **gated separately, never averaged**: a strong name cannot buy back a weak location.

No prior without grounding. One laundering hole must be closed explicitly, because it defeats the

whole trust model if left open: confidence derives from the channel a fact is *attributed* to, and the extractor also produces the attribution, so a model that misreads or invents "the developer's site says X" would mint a hallucination at developer-channel trust. The rule, uniform across channels (not just brochures): a channel prior applies only to facts carrying **mechanically verifiable grounding**, meaning a URL plus an extracted snippet that string-matches the fetched source, or a document region crop (Section 4.3). A fact whose grounding fails the mechanical check is scored as an LLM-inference-channel fact regardless of its claimed provenance. Extraction is its own error source, and each channel's prior carries an extraction-error component on top of its reliability judgment about the source behind it.

A residual gap remains: string-match grounding proves the snippet *exists*, not that the fact *follows from it*; a model can attach a real URL and a real sentence to a claim the sentence does not support, and no mechanical check catches that. Entailment is not mechanically verifiable at trust-assigning quality, so the defense is statistical rather than structural: an **entailment spot-audit** (of grounded facts sampled from the accept region, does the snippet actually support the fact?) joins the Section 6 scorecard as the check on what the grounding rule cannot reach.

Calibration plan, in one box (glosses for the non-specialist):

- *Label acquisition comes first, and it needs its own sampling design.* Adjudicated outcomes accrue naturally only in the review band, which means a naive calibration fit is estimated where humans already look and *extrapolated* into the accept region, exactly where calibration matters most. The calibration set is therefore drawn by **stratified random sampling across the full score range** (accepts, review band, and a sample of rejects) with inverse-probability weighting, and the Section 6 accept-region precision audit doubles as the accept-stratum labels.
- *Fact gate:* fit predicted score against adjudicated outcomes using **Platt scaling or isotonic regression** [31, 32] (standard methods that turn raw scores into true probabilities using labeled outcomes), fitted to the *post-reviewer* score, **per fact-type stratum**: a 0.45 location facet and a 0.45 unit-mix fact imply different real error rates, so thresholds are ultimately per-stratum, not global.
- *Pairwise linkage (Section 3.3):* **Winkler's EM** [10] (which estimates match/non-match probabilities from the candidate pairs themselves, no labels needed) applies because linkage is a true match/non-match problem, with its conditions stated as gating checks: it runs on *post-blocking* candidate pairs (so the estimated non-match parameters are blocking-conditional), it needs adequate pair volume and match prevalence to find the right mode, and per-market announcement counts are tens, not thousands. Pooling is asymmetric: the scarce *match* parameters are **hierarchically pooled** (shrunk toward a cross-market fit) while *non-match* parameters are estimated per market, since they are driven by local name-frequency distributions and abundant unlabeled pairs. Two caveats keep the pooling honest: match-parameter pooling additionally needs script strata (Section 3.7's transliteration axis breaks cross-market invariance), and the EM's own comparison features (name similarity, developer attribution) are conditionally dependent for true matches, the same assumption the corroboration term polices elsewhere. Clerically labeled pairs or informative priors remain the fallback below viable volume.
- *Label quality:* human adjudications are not noiseless ground truth (the clerical-review tradition documents reviewer error [11], and reviewer complacency (Section 5.4) predicts it), so a subsample is double-blind adjudicated, inter-annotator agreement is reported, and label noise propagates into the calibration fit.
- *Dedup audit:* provenance dedup's false-merge rate (independent sources wrongly collapsed) is sampled and reported on the same scorecard; it is tuned toward over-merging by design; the audit verifies the tuning. Because forfeited corroboration biases scores low just as the demote-only reviewer does, the measured false-merge rate feeds the same per-fact-type prior-adjustment loop, so the two downward biases are corrected jointly.

Bootstrapping the gate. Here is the day-one procedure the design implies for an adopter with zero adjudicated outcomes. Order the channels by *class* (official registry developer feed for identity facts commercial map open map targeted web lone mention LLM inference) and assign initial priors by rank; exact values matter less than the ordering, because the gate routes on relative position. Set the initial review band deliberately wide (a low accept threshold catches too little; err toward over-reviewing), and let the first weeks of human adjudication do double duty as the first calibration set. The thresholds move on

evidence, in one direction at a time: widen the band when the audit (Section 6) finds false auto-approvals, narrow it when sampled review outcomes show the queue is mostly rubber-stamps.

Some biases are predictable in advance. First, a reviewer that can only demote, stacked on conservative priors, produces scores that are miscalibrated *low*, inflating the review band and eroding the very attention economics Section 5.4 defends. Calibration must therefore be fitted to the **post-reviewer** score, and persistent under-confidence concentrated in specific fact types is an audit finding that feeds back into priors.

Second, provenance deduplication is an inference too, and it can err in both directions: derivation is detected by near-duplicate text overlap, publication-time ordering, and explicit citation or syndication markers. The failure modes are asymmetric: missing a derivation manufactures false consensus (the dangerous direction), while over-merging genuinely independent sources merely forfeits a deserved confidence bonus (the safe direction). The dedup is accordingly tuned toward over-merging, and its false-merge rate belongs on the same audit scorecard as everything else.

A worked example (numbers deliberately illustrative: read them as positions on the routing scale, not probabilities; making them probabilities is Section 6's calibration job). A brochure arrives announcing a tower in Meydan by a known developer. Extraction pulls the project name (clearly printed, per-field confidence high) from a channel whose identity-fact prior is high (say 0.85). Web search finds two articles; provenance deduplication traces both to the developer's own press release, so they collapse into the channel already counted, earning no agreement bonus. OSM has nothing, as expected pre-map. The reviewer agent cross-checks and finds that the developer's site names the parent area as Meydan while the masterplan render's coordinate falls near the seam of the *neighboring* area; it demotes the location facet (say to 0.55) and attaches the two source URLs and the conflict. Against illustrative thresholds of 0.80 and 0.45: the name fact clears the upper bar and auto-applies; the location facet lands in the review band and queues for a human with evidence attached; nothing here reaches auto-reject. One document in, two verdicts out, which is the point of scoring per fact rather than per document.

3.5 The reviewer layer

The reviewer agent arbitrates the pipeline, so its contract must be explicit. Its inputs are every upstream finding *with provenance attached*; its output is structured: a verdict per fact, an adjusted confidence, and the reasoning for any demotion.

The reviewer can lower confidence but never raise it. A reviewer that can promote becomes a hallucination amplifier. This monotone-demotion rule also bounds the damage of the known weaknesses of LLM evaluation: LLM judges exhibit position and verbosity biases [26, 27], and models systematically favor their own generations [28]. The design therefore commits to a reviewer from a **different model family** than the extractor (or a panel of diverse models [29]): a small extractor and a large reviewer from the same family share training lineage and do not defeat self-preference bias. Because the reviewer judges LLM-extracted facts, its errors still correlate with the extractor's; folding its verdict into the confidence score at full weight would double-count one error source, so its contribution is bounded; the reviewer-demotion audit (Section 6) explicitly measures extractor–reviewer error *correlation*, not just reviewer accuracy, since the bounded weight assumes a correlation it must eventually estimate.

What accompanies a flagged item into the human queue is retrieved evidence (source snippets and URLs) rather than free-form model reasoning alone. Generated rationales are not reliably faithful to the conclusions they justify [30]; a human reviewer anchored by a fluent confabulation is the automation-complacency problem [34] wearing a smarter mask. The reviewer's reasoning is shown as *one more piece of evidence*, never as the argument of record.

A final pass of **reviewer LLMs** guards the gate itself, scoring incoming conclusions and demoting suspect patterns (stale news, misattributed developers, invented POIs) into the manual queue. This layering does not stop the models erring, and things still slip through; silent failures are by definition unobserved, which is why the user correction-report channel exists as the detector of last resort. What the layering does is make erring cheap: an error routes to a human instead of silently becoming truth.

3.6 Validation at the point of entry

The graph enables a defense manual curation never could: **scoring listings as they arrive.** Each location node accumulates properties, including what kinds of units exist there. An incoming listing receives a correctness score against its claimed location across multiple parameters; if an area contains only villas and a listing claims a three-bedroom apartment there, the mismatch flags back to the lister at entry time. Wrong-community errors are caught before they enter the index rather than after users have seen them.

Not all such errors are innocent: in every property market, some listers *deliberately* misplace inventory into the more prestigious neighboring area, because "the premium community next door" is a sales tactic wherever a premium community has a next door. That shifts the threat model from noise to adversary, and the trust machinery extends naturally to meet it: trust levels attach to *listers* as well as to data channels, so an account whose listings repeatedly fail location checks earns a lower prior and closer scrutiny.

One consequence of the adversarial framing must be stated explicitly, because a naive design gets it backwards: **the listing's coordinates are also lister-supplied**, and dragging the map pin across the seam into the prestige polygon is at least as common as lying in the community field. Coordinates therefore cannot serve as the trusted anchor that community claims are validated against; both are claims. The trusted anchor has to come from the graph side: a resolved building entity (matched by name and alias against the claimed community's known buildings), an entrance-level identifier where official addressing provides one (Makani [8]), or a registry mapping where a land-registry linkage exists. The entry check then becomes a *consistency test across independent claims* (name, pin, and building identity must agree with each other and with the graph) rather than a validation of one lister-supplied field against another. Validation against an adversary is still validation, with memory, and with an anchor the adversary does not control. The precision-first posture here follows hard-won industrial practice: Yelp's duplicate-merging pipeline deliberately trades recall for precision because wrong merges are expensive to reverse [40]; wrong location assignments carry the same asymmetry.

3.7 Query-time resolution

Name resolution runs on fuzzy matching, absorbing abbreviations ("JBR") and misspellings of long official names, with string-similarity machinery the record-linkage literature has refined for decades [11, 12, 13]. The alias edges in the graph do the heavy lifting the string metrics cannot: no edit distance connects "JBR" to "Jumeirah Beach Residence"; only a curated alias relationship does.

Disambiguation across entity types is the harder half, and the first-cut mechanism costs nothing: **read intent from where the user is standing.** The same query "Meydan" resolves to the *area* on rent and buy surfaces and to the *developer* on new-projects surfaces; the page supplies the type signal the string is missing, with no extra question asked and no model in the loop (Figure 3). This is a production instance of the toponym-resolution literature's standard answer (use the context the reference arrives in [14, 15, 17]) and of the lesson of map-prior geocoders such as CamCoder, which showed that explicit geographic priors beat text-only disambiguation [16]; here, the graph itself (hierarchy, aliases, adjacency, popularity) is the prior. Where the surface carries no signal (a query typed into a generic homepage search), the system falls back to popularity-weighted ranking over the typed candidates: popularity computed from listing volume and page traffic per node, with an explicit **cold-start rule for pre-map nodes** (which have zero history yet matter most, per Section 3.3); a freshly announced development inherits a recency prior in place of the traffic term, so announcement-week queries resolve to the new entity rather than to whatever old node shares a token with it. Behavioral signals deepen this ranking as they accumulate (Section 8.2).

```
graph TD
    Q["query: 'Meydan'"] --> FM[Fuzzy match<br/>+ alias table]
    FM --> C1[Meydan · area]
    FM --> C2[MBR District One by Meydan · development]
    FM --> C3[Meydan · developer]
    CTX{Page context} -->|rent / buy| R1[rank area first]
    CTX -->|new projects| R2[rank developer first]
    C1 --> CTX
```

```
C2 --> CTX
C3 --> CTX
```

Figure 3: Disambiguation by context. Fuzzy matching and alias edges produce typed candidates; the surface the user is standing on ranks them.

One axis the alias machinery must own explicitly in this market: **script and transliteration**. Queries arrive in Arabic, in romanized Arabic with unstable spellings, and in mixed script, and no curated alias table enumerates the transliteration space, so the materialized index carries script normalization plus generated transliteration variants per name, and Arabic-script queries form their own stratum on the resolution scorecard (Section 6), since aggregate precision@1 can hide a script-specific failure mode. The pre-map cold-start prior also needs its bound stated: the recency term applies only to query tokens *novel to the graph* and ranks strictly below an established exact-match candidate's popularity; announcement week must not let "Marina Vista" hijack "Marina."

Behavioral evidence belongs in this ranking already: a marketplace's query logs already record which typed candidate users select after each ambiguous query, and co-click patterns are simultaneously a ranking signal, an alias-discovery channel (users teach the system "JBR" long before a curator does), and, as Section 8.2 develops, boundary evidence. The popularity-weighted fallback is the degenerate version of this; the full version treats the click stream as one more source with its own trust level, which is how everything else in this design already works. Airbnb's location-retrieval work reports the same evolution, from heuristics toward learned behavioral ranking [42].

On serving: resolution runs against a **materialized alias index** derived from the graph; the graph is the system of record, not the query path. Alias edges, typed candidates, and popularity priors compile into the search index offline, so query-time cost is an index lookup plus ranking, and graph maintenance never sits on the query path's latency budget. Resolution quality is measurable and belongs on the evaluation scorecard (Section 6): a labeled query set with precision@1 by entity type, and the null-result rate for queries the graph should have answered.

3.8 Boundaries and geometry

Nodes carry polygons sourced from OSM [3] and 2GIS [4] (a commercial map provider with strong regional coverage), and drawn manually where neither matches the ground truth, which is the pragmatic answer to seams like JBR/Marina where no official boundary exists. The source mix, in verifiable numbers: as of 2026-07-05, Dubai in OSM carries **130** administrative boundary relations at sub-emirate levels (admin_level 9–11), **322** sub-city place features mapped as *points* (suburb/neighbourhood/quarter nodes), and **310** mapped as ways or relations carrying place tags (verbatim queries, date, instance, and methodology caveats in Appendix A; all three counts are reproducible by anyone).

Read together and with their caveats, the counts say something more precise than "no polygons": geometry exists for many places, but only 130 carry *administrative boundary* status under a tagging convention the UAE has never settled, roughly half of named places have no geometry at all, and which polygon is authoritative for any given community is the contested question. **Named-but-unbounded and bounded-but-uncanonical are the two states that make listing assignment and boundary disputes hard**, and both dominate the open map here. In practice, then, OSM contributes POIs, some footprints, and a patchwork of place geometries of uneven authority, consistent with the volunteered-geography quality literature [51], while canonical community polygons lean on the commercial source and manual drawing, whose licensing discipline Section 4.2 covers.

Geometry lineage is single-source by rule. Cross-source comparison may gate whether a polygon is *accepted*, but a polygon is always adopted wholesale from exactly one source: coordinates from different sources are never blended, and geometry is explicitly exempt from the reviewer agent's merge-compatible-parts behavior. The rule has two independent justifications: blended geometry is untraceable (no version to roll back to, no source to blame), and blended geometry is a license event (Section 4.2: conflating open and commercial coordinates manufactures a derivative database no license permits).

When two sources disagree, the current rule is an explicit **interim stopgap**: boundary change under 1%

by area, adopt the newer polygon automatically; anything larger routes to manual review. Its failure modes are known:

- **Wrong metric.** Area-delta misses the case that matters: a sliver of change along a dense seam flips towers while staying under any area threshold. The gate this rule is evolving toward is **assignment impact**: auto-adopt only when no building or listing would change location.
- **Wrong recency semantics.** "Newer" is only meaningful *within* one source (an OSM edit timestamp and a commercial quarterly release date are not comparable signals), so recency-based auto-adoption is restricted to same-source version updates.
- **Wrong quality signal.** Most OSM edits are legitimate, but *boundary-relation* edits skew toward the problem cases (bulk imports, revert wars, armchair simplification), so acceptance weighs **changeset provenance** (a changeset being OSM's unit of edit, carrying its editor's history, the number of boundaries touched at once, and version stability) rather than timestamps.
- **Wrong change detector.** A boundary's geometry usually changes without the relation itself being edited: moving a *member way* re-shapes the polygon with no new relation version and no changeset touching the relation. Change detection therefore runs on a **hash of the fully assembled geometry** (relation + member ways + member nodes), with provenance drawn from the changesets of whichever members moved; relation-version monitoring alone is blind to the dominant edit vector.
- **Unsafe adoption.** Any accepted boundary edit needs topology validation against its neighbors: adopting one community's new polygon without re-checking shared edges silently creates the gaps and overlaps that corrupt point-in-polygon assignment (deciding which polygon a coordinate falls inside).

Representing the dispute needs a representation. Section 2.2's non-goal (represent contested seams rather than decide them) is only honest if the data model can actually hold a dispute, so: a contested boundary is stored as **multiple claim polygons per seam, each with its own source lineage and geometric confidence**, and consumers choose per use case. Search may index a seam's buildings under both communities and a location page can show the overlap zone, but **statistics use exactly one canonical assignment** (the highest-confidence claim): dual indexing improves recall, while double-counting a seam tower into two communities' price aggregates would corrupt the numbers the versioning discipline protects. This is the vernacular-geography posture (place extent as a distribution of claims, not a line, as the user-generated-content literature established by deriving city-core extents from photo tags [50]), applied with the same per-facet confidence machinery the rest of the graph already runs on.

Display is the least of what the polygons are for. They drive listing-to-location assignment, adjacency computation (which areas count as neighbors for search expansion), and the commute-time search of Section 8. A wrong polygon quietly corrupts all three, which is why boundary changes receive the same confidence treatment as any other fact instead of being trusted as "just map data."

An accepted boundary change is also not a row update; it is a **reprocessing event with a blast radius**, and the design needs a contract for it. Geometry is versioned; adopting a new polygon triggers, in order: re-assignment of affected listings (the impact set the gate already computed), backfill of area-level statistics computed under the old geometry (price aggregates, comparables), and invalidation of caches and search-index entries keyed by the affected nodes. The reassignment is transactional against a geometry version, so every downstream consumer can state which version of the boundary its numbers were computed under, and a bad adoption unwinds the same way a bad merge does (Section 3.3). During propagation there is a window where the location page and the search index briefly disagree; the design's position is that a *stated, bounded* inconsistency window beats the unversioned alternative, where nobody can say which geometry produced which statistic.

Assignment itself is not strictly point-in-polygon, because buildings are *footprints*, and the seam-straddling tower is where a centroid test and a footprint-intersection test disagree; assignment needs a stated rule (majority-footprint or entrance-anchored), and the entrance anchor is, fittingly, the one thing Makani [8] does define. Two lines of plumbing discipline round this out: area and distance computations require a *declared method* (an equal-area projection or geodesic computation on the ellipsoid; either works, but a "1% area change" is meaningless until one is named), and OSM boundary relations arrive broken often enough that geometry validation on ingest is a standing pipeline stage.

4. Implementation Notes

4.1 The graph lives in Postgres

Not a graph database: typed location nodes with explicit edge tables in Postgres, with PostGIS carrying the spatial half; polygon containment and spatial joins are what make the boring-database choice viable for this workload.

The reasoning generalizes, because "knowledge graph" reflexively suggests a dedicated graph engine. Graph databases earn their complexity on *unbounded traversal*: friend-of-friend queries where depth is unknown at query time. A location hierarchy is the opposite: depth is bounded at four or five levels, the traversals are known in advance (ancestors, descendants, siblings, aliases), and the overwhelming majority of production queries are "resolve this reference" and "join these listings to their location", indexed lookups and joins that Postgres has spent thirty years being excellent at. Choosing the boring store also keeps the location graph inside the same operational envelope as the rest of the platform: the same backups and replicas, and people who already know how to run it. Infrastructure that shares an envelope gets maintained; infrastructure that needs its own priesthood gets stale. A market graph is tens of thousands to low hundreds of thousands of nodes; nothing in this workload strains a relational store.

4.2 Working with licensed geo data

A production graph built partly on open map data carries license obligations that need engineering attention. The plain-language stakes: open and commercial map licenses do not mix freely, and mixing them carelessly can obligate opening your own data. OSM ships under the ODbL, whose operative interpretations live in the OSMF Community Guidelines (Collective Database, Substantial Extract, Produced Work) [49]. The traps, each with its one-sentence rule:

- **The attribution obligation.** Rendered pages built from OSM-derived facts are Produced Works; "© OpenStreetMap contributors" must appear *visibly on the pages*, not only in a colophon.
- **The conflation trap.** A pipeline that compares OSM and commercial polygons and merges them is *creating* a derivative database, and per-fact provenance tags cannot un-derive geometry once sources have informed each other, which is why Section 3.8's single-lineage rule is a license requirement as much as an engineering one: separated layers form a Collective Database, the defensible structure.
- **The derived-relations trap.** The lineage rule must extend past polygon adoption to spatial facts *computed across* lineages: an adjacency edge or dual-claim overlap zone calculated between an OSM polygon and a commercial polygon is a product of both databases, so derived relations carry both parents in their provenance and inherit the stricter obligations.
- **Structured markup.** Markup generated from the graph (Appendix B) exposes *data*, not just rendering; a page emitting machine-readable OSM-derived facts can re-enter derivative-database territory and needs the same lineage discipline as the store.
- **The tracing trap.** "Hand-drawn from imagery" is not automatically clean-room: imagery carries its own license, major providers prohibit derived vector data, and the tracing permissions some grant to OSM contributors do not extend to a proprietary graph. Manual polygons must rest on own survey data, openly licensed imagery, or an imagery license that explicitly permits derived vectors.
- **The API-terms trap.** Commercial map providers' standard API terms typically prohibit extracting and persistently storing geometry at all; that channel must rest on a negotiated data license, not developer ToS.
- **The geocoding trap.** Resolving references against OSM-derived data and storing the results is also governed by the OSMF Geocoding Guideline [49]: stored geocoding results over OSM data remain ODbL-governed and carry the same lineage discipline as the store.
- **The one-way valve.** Commercially sourced and brochure-derived data must never be contributed back into OSM.

4.3 Document extraction

The brochure channel (Section 3.3) rests on multimodal document understanding: designed PDFs mixing text, tables, floor plans, and masterplan renders. The lineage here is the document-understanding literature: OCR-free end-to-end extraction of the Donut family [36] evaluated against benchmarks such as DocVQA [35] and, for the diverse-real-PDF setting this channel actually faces, OmniDocBench [37], which documents the fine-grained failure modes (tables, reading order, mixed layouts) that make per-field confidence necessary. Extraction grounds each emitted fact in the source region it came from (the retrieval-grounding posture of RAG systems [38] applied to document extraction), which is what makes the "evidence snippets over free-form reasoning" rule of Section 3.5 implementable: the queue shows the reviewer the crop of the floor-plan table, not the model's paraphrase of it. The cost: this is the most expensive channel per unit of engineering (a multimodal extraction pipeline with per-field grounding is a subproject with ongoing model-drift maintenance, not a prompt), and it earns that cost only in markets where off-plan inventory matters commercially (Section 3.3).

Every channel's extraction converges on one artifact worth showing, because every downstream section presupposes it. The **fact record** below is the worked example's location facet from Section 3.4, with the composition's terms carried in log-odds (values illustrative):

```
{ "entity": "loc_9f2e",                "facet": "geographic",
  "fact": "location.parent",           "value": "area:meydan",
  "source": "brochure:dev_emaar:2026-06-12",
  "extraction_field_conf": "high",     "channel_prior_logit": 0.25,
  "corroboration": [],                 "reviewer_adjustment_logit": -0.40,
  "score": 0.46,                       # =  (0.25 - 0.40)
  "region": "review",
  "evidence": ["s3://…/brochure_p4_masterplan.png"] }
```

Structured output is schema-enforced at the extraction boundary; a response violating the schema is a failed extraction (retried, then dropped), never a partially trusted one, and an empty cascade result is an explicit `NO_DATA` outcome rather than an absent record.

4.4 Inference economics

R5 is an economics requirement, so the cost architecture is a design surface. Four mechanisms shape the bill. **Model tiering:** cascade stages run on the cheapest model class adequate to the stage (extraction and search-result triage on small models, the reviewer pass on a stronger one), because the reviewer token buys the most (it gates everything) and the broad stages are the highest-volume. **Caching:** research results are cached per (query, source) so re-runs and neighboring entities do not re-pay for the same acquisition; the enrichment run, not the fact, is the unit of spend. **Cascade ordering as cost control:** cheap broad sources run first so that the expensive targeted stages run only on entities that still need them (Section 3.2); the cascade is a cost filter as much as a research strategy. **Re-research discipline:** the open cadence problem (Section 9) is bounded by triggers rather than timers (creation, flags, correction reports, and the demand shifts of Section 8.2), so quiet entities cost nothing at rest. The scorecard units of Section 6 (cost per accepted fact, cost per caught error) are what make this section falsifiable: a deployment whose cost-per-accepted-fact exceeds its curator baseline has automated the wrong thing.

A worked cost sketch makes the shape concrete, as Section 3.4's worked example did for confidence (numbers illustrative, from public list prices, mid-2026). Suppose a cascade run reads ~50k tokens of source material and emits ~5k (extraction + triage on a small-tier model at ~$0.10/M input, $0.40/M output **$0.007**), and the reviewer pass reads ~20k and emits ~2k on a frontier-tier model (~$3/M in, $15/M out **$0.09**); call it **$0.10 of tokens per researched entity**, dominated by the reviewer, which is the design intent. But the token bill is the *smallest* term; the unit that decides anything is the blended formula the scorecard requires:

```
cost_per_accepted_fact = ( token_cost
                         + review_band_fraction × reviewer_minutes × loaded_rate
                         + amortized_engineering / fact_volume ) ÷ acceptance_rate
```

Units: `reviewer_minutes` is minutes per reviewed fact and `loaded_rate` is currency per reviewer-minute, so every term resolves to currency per fact. The second term dominates wherever the band is wide (which the bootstrap deliberately makes it, early on), and the third term is real (Section 4.3's brochure pipeline is a large ongoing engineering cost), which is why this sketch draws no headroom conclusion against a curator baseline. The comparison that decides R5 is the audited one: blended cost per accepted fact at matched precision, versus the fully loaded curator cost per fact on the manual baseline. The sketch's only claim is about *shape*: tokens are not where this system's economics are won or lost; the review band and the amortization are.

4.5 Deployment status

A systems paper should state plainly what runs versus what is design intent; Table 2 does, within the disclosure constraint of Section 1.

Table 2: Component status.

Component	Status
Typed per-market graph; 2GIS/map-verified facts; confidence scoring with three-region routing; schema enforcement and grounding checks	Operating
Multi-variant resolution with alias handling; LLM alt-name and fallback research; human review queue	Operating
Entry-time listing validation with per-lister trust	Operating
Type-migration machinery (enumerated re-evaluation of attachments)	Operating in basic form
Manual + map-sourced polygons; area-delta boundary stopgap	Operating (stopgap acknowledged)
Pre-map channel (developer monitoring, brochure extraction)	Partial: extraction operating, linkage-on-build maturing
Assignment-impact boundary gating; topology validation; dual-claim seam representation	Design, replacing the stopgap
Public-data pilot of the decision gate (open OSM task, audited sample)	Executed: Section 6.1, reproducible
Calibration (EM for linkage, Platt/isotonic for gate); learned source trust; seeded-error audits	Planned: the Section 6 program
Cross-model reviewer panel; behavioral ranking signals; demand layer; specialist roster	Planned (Sections 3.5, 8)

4.6 If starting from zero

The components are separable, and their standalone value orders a build. Effort bands are the author's field judgment, not measurements, and assume a team already running a listings platform.

1. **Typed graph plus entry-time validation:** immediate defense of the inventory layer against the errors of Section 2.1, using whatever location data already exists. *Roughly two or three engineers, one quarter; no ML specialization required.*
2. **Research cascade with the review queue:** the maintenance engine, where curation cost starts bending. *A small team plus the first reviewer hires; a quarter to first market.*
3. **Pre-map channel:** largest commercial payoff, highest build cost (Section 4.3), meaningful only where off-plan inventory matters; adopters in phased-masterplan markets should budget for manual announced-to-built linkage until the 1-to-many machinery of Section 3.3 matures. *A dedicated subteam, ongoing.*
4. **Boundary machinery and calibration program:** the long-tail correctness work. *Continuous, paced by the audit findings of Section 6.*

Building in the reverse order defends nothing until the exotic parts land.

---

5. Production Experience and Lessons

The evidence constraint of Section 1 applies throughout this section. What follows are field-observed failure patterns; where a lesson is narrated as an incident arc, the narrative is a **composite**: the shape of what happens, assembled at pattern level, with identifying detail removed. Each lesson closes with its status: exercised in operation, or prescribed by the design in response.

5.1 Entity types are mutable state

A natural design assumption is that an entity's type is a fact. The ground truth disagrees, and the damage lands far from the change itself.

The composite arc: a large mixed-use tower (one node, type *building*, with years of accumulated attachments) matures. Its podium retail gets its own identity; its residents start writing "the community" in reviews; agents start listing units against sub-addresses inside it; eventually the market, the developer's own marketing, and the map all agree that this is no longer a building but a *community* containing several structures. The type migration itself is one attribute write. Everything attached to the node is now subtly wrong: reviews written against building criteria (elevators, gym, management) render under community criteria (schools, noise, walkability) as nonsense; the entry-time validation rules keyed to *building* unit-mix constraints begin flagging legitimate listings; the polygon that outlined one footprint now under-bounds an area; the location page's template, its structured markup, and its URL semantics all assume the old type. Nothing crashed. Everything downstream of the node quietly stopped meaning what it said, and the failures surfaced over weeks, each looking unrelated, each filed against a different team.

The design response is the lesson: **type is mutable state with migration semantics.** A type change is a lifecycle operation that enumerates everything hanging off the node (reviews, attributes, validation rules, rendering contracts) and re-evaluates each against the new type's contract, invalidating what cannot carry over. *Status: the failure pattern is exercised; the enumerated-migration machinery is the design response, operating in its basic form (Section 4.5).*

5.2 Schemas need demotion paths

The inverse failure class is equally real: entities that should never have existed. Open-data ingestion will happily deliver a roundabout or a dance studio as a *sub-community* (both real patterns, not hypotheticals) because somewhere upstream a mapper tagged generously and no schema rule said a sub-community must contain dwellings. The subtle part is not the garbage; it is what accretes to the garbage before anyone notices. A phantom sub-community acquires a location page. The page gets indexed. A listing or two lands in it via fuzzy match. By the time a human looks, deleting the node means breaking a URL that ranks and orphaning listings that real agents placed: the garbage has *tenure*.

Hence the lesson stated structurally: creation paths are not enough; the schema needs **demotion and deletion paths with the same reversibility guarantee as every other automated entity decision** (Section 2.3), and demotion must handle the accretion problem: reassign the attached listings, redirect the URL, and record the tombstone so ingestion does not resurrect the same phantom from the same source next run. *Status: the failure pattern and manual demotion are exercised; automated demotion remains human-triggered by design (Section 9).*

5.3 Geometry is harder than semantics

The intuitive expectation is that POI acquisition and naming are the hard parts; they are, after all, the parts with the most entities. They are not the hard parts. The semantic layer (what things are called, what they contain, who built them) largely yields to LLM research: the cascade finds names, aliases, unit mixes, and developer attributions with satisfying regularity, because developers publish them on purpose.

The geometric layer is the opposite. Where an area actually starts and stops was never published by anyone, because no one ever needed to commit to it: the developer's masterplan render stops at the road, the municipality's parcel data follows plot lines nobody searches by, OSM has a point where a polygon should be (Section 3.8's counts), and the commercial map's boundary was drawn by a cartographer making the same judgment call this system now has to audit. The composite experience is a research pipeline that closes semantic facts by the hundreds while a human with an imagery layer adjudicates boundary claims one seam at a time, and the seam adjudications are the ones that move listings. This is field experience rather than a measured result, but the recommendation is concrete: **teams budgeting effort for this class of system should invert the intuitive allocation.** *Status: exercised. This lesson is why the boundary machinery of Section 3.8 exists and why its hardening sits at the top of the design queue.*

5.4 Review queues are attention systems

The human-factors literature has known for decades that automation reshapes human vigilance: high false-alarm rates train operators to ignore alarms, and reliable automation breeds complacency that expertise does not cure [33, 34]. Both failure modes apply directly to curation queues. A queue that is mostly obviously-fine changes trains reviewers to approve on autopilot (the reviewing equivalent of alarm fatigue), which is why confidence gating is as much an *attention* design as a workload one: done right, it does not merely shrink the queue, it makes the remaining reviews real. This is also why Section 3.5 insists the queue present evidence rather than fluent model rationales: a persuasive wrong argument is the complacency trap the literature warns about.

---

6. Evaluation Framework

This paper publishes no operational metrics (withheld under the disclosure of Section 1, not for lack of measurement), so we are precise instead about the evaluation a deployment of this design must provide its operators. The evaluation design is itself part of the architecture.

The first metric is workload, because the requirement it tests is R5, sublinear curation. The success shape: the majority of location updates, including entirely new locations, flow through the automated confidence-gated path, and the human review queue becomes what it should be, a place for genuinely ambiguous cases rather than a bottleneck every change waits behind.

But workload alone is uninterpretable. A system can trivially eliminate its review queue by auto-approving everything. Any workload claim is meaningful only when paired with a **precision audit of the automated path**: sample a few hundred auto-approved changes, have humans adjudicate them blind, and report the false-auto-approval rate. This is standard clerical-review audit practice from the record-linkage tradition [9, 11], it requires no external benchmark to exist, and it is the single number that separates "the gates work" from "errors ship silently."

The fuller scorecard:

- **Entry-time flag precision:** of listings flagged at entry (Section 3.6), how many were genuinely misplaced;
- **Anchor coverage:** the fraction of incoming listings carrying a trusted anchor the adversary does not control (resolved building, entrance identifier, registry link; Section 3.6), reported per market, with the no-anchor policy stated: anchorless listings route to review rather than accept at degraded trust;
- **Seeded-error recall:** precision audits measure only the gameable half of an adversarial defense: a lister who learns the unit-mix check simply misplaces listings that pass it, and low-traffic areas generate no correction reports. Inject known-misplaced synthetic listings through the entry gate periodically and report the catch rate, complemented by **naturally labeled positives** (user-confirmed corrections, announced-to-built matches later manually verified), because seeded errors only cover the error distribution the seed-writer imagined, and Section 3.6's own adversarial framing predicts listers adapt past it; seeded listings carry a quarantine flag that excludes them from serving, statistics, and lister-trust accounting; only the gate's verdict is recorded;

- **Reject-region sampling:** the fact gate's third region needs its own audit; facts auto-discarded below _lo include *false rejects* (true facts silently lost), which no precision audit or correction report ever surfaces. Periodically sample the reject stream (or seed known-true facts through the cascade) and report the false-reject rate;
- **Entailment spot-audit:** of grounded facts sampled from the accept region, does the snippet actually support the fact value (Section 3.4's residual gap: grounding proves existence, not entailment; a misread table cell passes the string match);
- **Correction-report rate trend:** the user channel is the detector for what the gates miss, weighted by traffic, which is why it complements rather than replaces the seeded audit;
- **Query-resolution quality:** a labeled query set with precision@1 by entity type and the null-result rate for queries the graph should answer (Section 3.7);
- **Time-from-announcement-to-searchable** for pre-map locations (Section 3.3);
- **Source-change-to-graph latency:** time from a source changing (a developer site update, an OSM edit) to the graph reflecting it, sampled per channel; announcement-to-searchable covers only the pre-map channel;
- **Confidence calibration:** predicted confidence versus observed human-overturn rate, fitted to the post-reviewer score (Section 3.4), using standard post-hoc methods [31, 32] once adjudicated outcomes accumulate;
- **Reviewer-agent audit:** sample the reviewer's demotions against blind human adjudication, since a miscalibrated reviewer silently moves the gate (Section 3.5);
- **Economic units**, because R5 is an economics claim: **cost per accepted fact** and **cost per caught error**. Run the comparison the way a finance partner would: fully loaded curator cost per fact touched on the manual baseline, versus pipeline token-plus-review cost per fact accepted at equal-or-better audited precision. The comparison is only valid at matched precision (cheap wrong facts are not savings), and both units should be reported per market, because a market bootstrapping its graph and a market in steady state have different denominators;
- **User-outcome measures:** the operator scorecard proves the machine works; product needs measures that prove the *product* got better: search null-result and reformulation trends on location queries (Section 3.7), location-related complaint and correction-report rates, and time-to-searchable for announced projects tied to the leads they generate.

Among the planned audits, the **reviewer-demotion audit runs first**: it is days of effort and de-risks the component whose errors move the gate most.

The queue these metrics govern is a staffed system, so its operating model belongs in the design: reviewers need market familiarity more than GIS skill for semantic facts and the reverse for boundary claims (two queues, not one, once volume justifies it); the queue carries an SLA because entry-time flags block listers who are trying to publish; and a market launch is a planned queue-surge event: backfill enters as bulk-tentative with sampling rather than flooding the band (Section 3.4's wide-band bootstrap is the same principle at threshold level).

None of these require a public benchmark, which is fortunate, because none exists for location-graph correctness (Section 9). An open benchmark, built on a fully public market from open data, would serve every operator of systems in this class. Rather than only recommending it, this paper executes its first slice.

6.1 A public-data pilot of the decision gate

To put at least one audited number where the contract demands one, we ran the three-region gate on a fully public task: linking Dubai's sub-city **place point features** to its **administrative boundary relations** in OSM (the same 322 nodes and 130 relations counted in Section 3.8). The same real-world place frequently exists as both a point and a relation, so node relation identity is a genuine match/non-match problem, and public data makes every verdict re-checkable by any reader. The pilot (a short standard-library-only script with a fixed random seed; code, data extracts, and adjudication samples: fbritoferreira.com/research/the-living-map) scores each node's best candidate as $0.6 \times$ name similarity (across all name tags, normalized) $+ 0.4 \times$ proximity (2 km exponential decay on centroid distance), with a cheap name pre-filter as blocking, and routes against the bootstrap thresholds of Section 3.4 (_hi = 0.80, _lo = 0.45).

Results (2026-07-06 data): of 322 nodes, the gate auto-accepted **97** (30%), routed **100** (31%) to the review band, and rejected **125** (39%). A fixed-seed sample of 40 accepts and 20 review-band items was adjudicated by the author. **All 40 auto-accepts were correct, for accept-region precision of 1.00 on the sample**, with a one-sided 95% upper bound on the true error rate of ~7.5% (rule of three, zero failures in 40). The review band behaved exactly as designed: 4 items were correct matches under-scored by centroid distance (large areas whose centers sit far from their place nodes; a human approves them), and 16 were genuine ambiguity, dominated by the parent–child pattern of numbered districts (Al Tawar 1 vs. Al Tawar), including one **would-be false accept caught by the band**: Warsan 1's best candidate was its *sibling* Warsan 4, scored 0.475, correctly held for review. The pilot also caught live open-data vandalism (a place node whose Arabic name tag read "data quality improvements") while the match was carried correctly by its English name: Section 5.2's garbage class.

All three regions audited. A second fixed-seed sample of 25 rejects was adjudicated against each node's full candidate list: **25 of 25 were correct rejects** (no admissible match existed among the 130 relations), bounding the false-reject rate below ~12% at 95% confidence on this sample. The reject audit surfaced its own finding: among the correctly rejected nodes were *Dubai Marina, Burj Khalifa, and Dubai Internet City*: the emirate's most searched places have no administrative boundary relation in the open map at all, which is Section 3.8's coverage argument in its most famous examples. One near-miss is worth recording: *Al Quoz Industrial 2*'s best candidate was the similarly named but different *Al Qusais Industrial 2* at score 0.441, kept out of the review band by 0.009, a concrete illustration of why the composition's disagreement weights and per-stratum thresholds matter.

A first calibration datapoint. Fitting isotonic regression (pool-adjacent-violators, stdlib) on the pilot's own 60 adjudicated labels yields a monotone step function whose top block begins near score 0.76, below the bootstrap _hi = 0.80: all 41 adjudicated labels above 0.76 were correct (rule-of-three bound: error below ~7.3% at 95% on that block). By the calibration box's own small-set rule, Platt scaling is the textbook choice at n = 60; isotonic was chosen because pool-adjacent-violators is implementable in the standard library, and the deviation is noted. Re-routing the same 322 entities at the fitted threshold is an in-sample illustration on the same 60 labels, not out-of-sample validation. It shrinks the review band from **31% to 27%** while the accept region grows from 30% to 34% (97 to 109 entities); the original 40-item audit stays inside the enlarged region, the only labeled items among the newly accepted are the 4 under-scored correct matches from the adjudicated band sample, and the enlarged region carries no fresh precision bound of its own. That is the claim Section 3.4 makes abstractly, in a small, real instance: calibration converts an over-wide bootstrap band into recovered automation, on evidence. (Proximity uses haversine great-circle distance, declared per Section 3.8's declare-a-method discipline; Section 3.8's production recommendation is an equal-area projection or ellipsoidal computation. The fit and threshold sweep ship with the pilot code.)

Threshold sensitivity (from the published run; an adopter's queue-sizing tool): _hi = 0.80 routes 30% accept / 31% review; 0.76 routes 34% / 27%; 0.74 routes 34% / 27%, the fitted-threshold row being the calibration datapoint above. Adjudication effort was not instrumented in this run; timing the queue is scorecard work (Section 6's reviewer-minutes term) and belongs in the pilot's next iteration.

What this does and does not show. It shows the decision structure doing its job on a real public task: a clean accept region at bootstrap thresholds, all three regions audited, and the genuinely ambiguous cases (including at least one outright error) landing in the band where a human looks. It does not validate the production system: the pilot task is *easier* than production listing assignment (exact-name pairs are common in it), the score is a two-feature toy beside Section 3.4's composition, thresholds are the uncalibrated bootstrap values, and the adjudication was single-rater, by the author, and not blind to the gate's scores and verdicts, so it does not meet the blind-audit standard Section 6 sets for adopters (85 items across the three region samples, re-verifiable from the published samples); re-verifiability is the compensating control. The 31% review band at bootstrap settings also illustrates Section 3.4's "wide band first" posture and its cost; that is the band calibration exists to shrink. As the first slice of the open benchmark this section calls for, the pilot is small; it is also, to our knowledge, the first published, reproducible precision audit of a three-region gate on open location data, and extending it (more markets, listing-assignment tasks, stronger scores, multi-adjudicator labels) is concrete, fundable community work.

7. Related Work

None of the individual pieces here are new; this section is precise about what already exists and where each layer stops. For the reader whose first question is "why not just buy this," the one-glance version:

Candidate	What it solves	Where it stops for this problem
Google Places / Maps APIs [47]	Global geocoding, POIs, planetary fusion	Location reference address; no market entity types; no off-plan; terms restrict storage
Placekey [45]	Shared join identifier	Standardizes the ID, not the resolution or the semantics
GeoNames / Who's On First [1, 2]	Global gazetteer, stable IDs	Community-paced; no developments/developers; no market freshness
OSM / Overture [3, 5]	Open base map, GERS stable IDs	Coverage thinnest where needed most (Section 3.8); no demand semantics
2GIS-class commercial maps [4]	Strong regional coverage	A source, not a system: no lifecycle, no listing validation, no pre-map
Hybrid: buy substrate (Overture/Placekey) + build validation and the review queue	Fastest defended-listing path; stages 1–2 of Section 4.6 on bought rails	Still needs the gate, the queue, and the audits: the parts nobody sells
Build on this paper's design	Typed truth layer over all of the above	The build cost and audit obligations of Sections 4–6

Gazetteers and open place data. GeoNames [1] and Who's On First [2] solved the "big list of places with stable identifiers" problem years ago, and Who's On First in particular got the philosophy right: a gazetteer as the place where disagreement about places is *managed rather than decided*, which is the seam problem of Section 2.1. What global gazetteers cannot do is keep pace with one fast market or model the entity types property search lives on. OpenStreetMap [3] supplies the structural substrate but, as Section 3.8 reports, its administrative-boundary coverage is thinnest where this system is needed most; Overture Maps [5] is the consortium answer, and its GERS identifiers address the identifier-stability problem Section 9 flags. Foursquare's open Places release [46] demonstrates POI data at industrial scale. We treat all of these as *sources, not the graph*: they feed the cascade and are scored like every other channel.

Spatial indexes and place identifiers. H3 [6] and S2 index *space itself* into hierarchical cells for fast spatial queries; they have no opinion on what a place is or is called; semantics live above the index. Placekey [45] standardizes the *identifier* (an H3-based Where plus an address/POI What) so datasets can join without pairwise entity resolution; it does not resolve vernacular naming, market hierarchies, or marketing-versus-legal aliasing. A location graph of the kind described here is the resolution layer that could *emit* such identifiers.

Geocoders and addressing systems. Google Places, Mapbox, and open-source stacks such as Pelias [7] map strings to coordinates and back, embodying the assumption Section 2.1 rejects: that a location reference is an address. Official addressing systems make the same bet at the government layer: Makani [8] is the strongest version and still answers only the logistics question. Google's own account of its mapping pipeline [47] describes fusing over a thousand sources with heavy operator involvement, planetary scale bought with an operational budget no vertical player has; the design in this paper is the vertical alternative: market-exact quality without that operational machinery.

Geospatial knowledge graphs. WorldKG [18] lifts OSM into a world-scale typed knowledge graph and

is the closest general-purpose ancestor of the data model here; the difference is domain specialization: building/community granularity, market-specific hierarchy, and query-serving semantics that a global OSM lift does not target. KnowWhereGraph [19] demonstrates space, place, and time as the natural integration backbone across data silos; this system is a vertical, consumer-search-serving instance of that thesis rather than a scientific data-integration warehouse. YAGO2geo [20] showed that point-centroid gazetteers are insufficient and precise polygons are required for real containment queries, the same argument Section 3.8 makes operationally, extended to *unofficial, vernacular* geographies no administrative source covers.

Toponym resolution and geoparsing. Resolving "an ambiguous reference to an entity of unknown type" is the toponym-resolution problem, studied from Leidner's foundational treatment [14] through Gritta et al.'s failure-mode analysis (ambiguity, metonymy, weak context use) [15] to the current survey landscape [17]. CamCoder [16] demonstrated that explicit geographic priors beat text-only disambiguation; Section 3.7's design is that lesson in production form, with the graph itself as the prior and the user's surface as the context signal. The literature's benchmark domains are news and social media; short, noisy marketplace queries, where gazetteer quality dominates model choice, remain underserved, which is part of why this paper exists.

Entity resolution. The record-linkage tradition (Fellegi–Sunter's decision model [9], Winkler's EM-based estimation for the no-labels setting [10], the methods corpus [11, 12], and applied address matching [13]) supplies the formal shape of the confidence gate (Section 3.4) and its hardest-won lesson: no similarity measure wins everywhere; context picks the measure.

Truth discovery and data fusion. Source trust itself has a literature this system descends from rather than invents: TruthFinder established iterative source-reliability estimation over conflicting web claims [52], the truth-discovery survey maps the field [53], and Knowledge Vault [54] built web-scale fact fusion with confidence derived from *both* source reliability and extractor error, the same decomposition as Section 3.4's "no prior without grounding" and its extraction-error component. Two deltas locate this system against that line: truth-discovery methods *learn* source reliability from data, which Section 9 concedes this system does not yet do; and they model source *accuracy*, where the property domain forces modeling source *incentive*; the fact-type split in which a developer is simultaneously authoritative on identity and adversarial on location has no analogue in accuracy-only fusion. Contribution 2 is best read as an incentive-aware, adversarial-domain specialization of per-attribute source-reliability estimation, plus the per-facet survivorship of Section 3.3.

LLM-built knowledge graphs and LLM reliability. The research community has converged fast on LLMs as knowledge-graph builders [21], while parallel literatures catalog why unguarded LLM output cannot be trusted as fact [22], why verbalized confidence is miscalibrated [24, 25], why LLM judges are biased (positionally [27], toward their own generations [28]), and why generated rationales are unfaithful [30]. Most construction work is benchmark-evaluated; the reliability work is why Sections 3.4–3.5 look the way they do. AutoKnow [39] is the closest industrial ancestor in spirit (self-driving knowledge collection at catalog scale with minimal labeling), for taxonomic product data; the location domain swaps taxonomy for containment, adds geometry, and faces adversarial location claims product catalogs do not.

Industrial place and marketplace systems. Yelp's duplicate-business pipeline [40] established the industrial pattern this system's entry validation follows: candidate generation, a high-precision classifier, and a human review queue, with precision deliberately favored because merges are hard to reverse. Airbnb built a knowledge graph for contextualizing inventory [41] and reports location retrieval as a conversion-critical, learning problem [42]; its neighborhood geometries were famously drawn by hand [43], and Zillow manufactured and published US neighborhood boundaries in 2008 because no authoritative source existed [44]; both are the manual ancestors of the boundary problem Section 3.8 automates around. This paper stakes no novelty on the ensemble: property portals mostly do not publish their location systems, so absence from the literature is partly absence *of* the literature. The pieces most likely to be genuinely uncommon are narrower and can stand alone: the **pre-map channel** (entity creation at announcement time, from developer monitoring and brochure extraction, with per-facet survivorship when the physical entity arrives) and the **fact-type split of source trust** (a source authoritative on identity and adversarial on location). The rest is disciplined combination; a practitioner's design report of the whole therefore has value to the teams that must build one.

8. Future Directions

8.1 A roster of specialists

The end state extends the logic the system already runs on: a growing roster of **specialized LLM agents, each expert in one slice of the process** (POI research, developer verification, brochure extraction, boundary analysis, transaction interpretation, content generation), with a reviewer layer synthesizing their outputs into the final per-location picture (Figure 4). The multi-agent literature is candid that this shape has open problems: inter-agent error propagation, arbitration, cost [23]. Our answer remains the mechanism of Section 3.4: agents do not resolve disagreements with each other; the reviewer layer and the confidence bar do. Each new agent must pass one admission test: does the marginal quality gain survive the marginal review cost?

```
graph TD
    A1[POI research agent] --> R[Reviewer agent<br/>synthesis + confidence scoring]
    A2[Developer verification agent] --> R
    A3[Boundary analysis agent] --> R
    A4[Transaction interpretation agent] --> R
    A5[Content generation agent] --> R
    R -->|high confidence| G[(Location graph)]
    R -->|low confidence| H[Human review<br/>+ reasoning]
    H --> G
```

Figure 4: The end state: specialist agents per process slice, a reviewer agent synthesizing under the confidence gate, humans above the bar.

8.2 Demand as a signal

Everything in Sections 3–6 treats the graph as supply-side: what exists, where, called what. The next layer is demand: wiring external and internal interest signals onto location nodes (search-trend interest for an area's names and aliases, search-console query data, and the platform's own logs). Once demand lives on the node, several things follow. **Demand-weighted freshness** gives the re-research cadence problem (Section 9) its missing principle: re-research where demand rises, not on a timer. **Early-warning detection** turns the unknown-query stream into a discovery channel; sometimes demand appears before the developer announcement does. **Boundary evidence from behavior**: where users searching one area consistently click into listings across a seam, they are voting on where the boundary actually is: demand data as evidence in the disputes Section 9 calls irreducibly social. Alias handling doubles in value here, because interest signals arrive under whatever name people type: trend volume for "JBR" and "Jumeirah Beach Residence" is one signal, not two.

Events as a node type extend the same pattern to time-bound entities: an events calendar attached to location nodes (exhibitions, festivals, openings), sourced through the existing cascade. Events enrich content (what is happening there *now*) and explain demand: a query surge during a major exhibition is not a structural rise in relocation interest, and a graph that knows the event can tell the difference, keeping both the re-research trigger and demand-weighted prioritization from chasing noise.

8.3 Search by time, and beyond property

A boundary-and-POI-aware graph can answer a query no string index can: commute-based search ("within 20 minutes of X"), which is on the roadmap as the graph's geometric layer matures. Beyond property, the architecture's core (a typed location graph, an LLM research cascade, confidence-gated automation) transfers to any business that depends on knowing what is where in fast-changing cities. The transfer is not free: in mature-addressing markets the source-trust hierarchy inverts, because authoritative registries (parcels, assessor data) exist and the hard problem becomes reconciling authoritative-but-conflicting sources

rather than compensating for their absence. The confidence machinery transfers, but the priors have to be re-derived per market.

---

9. Limitations

- **Open-data coverage bias.** OSM and commercial coverage are thinnest exactly where this system is needed most. The cascade compensates with web and developer sources, but a market with sparse open map data starts the graph from a worse floor (Section 3.8).
- **The confidence score is not yet a calibrated quantity.** Composition and thresholds are engineering judgment with the right decision shape (Section 3.4); fitting them to adjudicated outcomes [10, 31, 32] is the next measurement after the precision audit.
- **Source trust does not yet learn.** Trust levels are assigned, not earned from outcomes; a channel that repeatedly ships wrong delivery dates keeps its prior until a human notices. Closing the loop (review overturns and correction reports feeding per-source trust) converts trust from editorial opinion into an estimated error rate.
- **Merge, split, and identifier stability lack production mileage.** Section 3.3 states the 1-to-many design (parent project node, phase children) and the blocking scheme; what remains is hardening: the machinery operates in basic form, and the split path leans on demotion plumbing that is itself human-triggered. Overture's GERS identifiers [5] are the open ecosystem's attempt at the stability half and worth tracking as an anchor.
- **Re-research cadence has triggers, not a principle.** Creation, flags, correction reports, and demand shifts bound the cost (Section 4.4), but nothing yet measures how long a source change takes to reach the graph, which is why the scorecard adds source-change-to-graph latency (Section 6).
- **Manual review scales sublinearly, not to zero.** Confidence gating shrinks the queue but never empties it, and garbage-entity demotion (Section 5.2) remains human-triggered until far more confidence history accumulates.
- **No public benchmark for location-graph correctness exists.** Every operator audits privately or not at all. Building an open benchmark on a fully public market would serve everyone who touches this problem (Section 6).
- **Context-based intent is coarse.** Surface context disambiguates well between rent/buy and new-projects intents; a bare homepage query carries no type signal and falls back to popularity ranking until behavioral signals (Section 8.2) land.
- **Some truth is irreducibly social.** No pipeline settles which side of the JBR/Marina seam a tower is on, because residents themselves do not agree. The system's ceiling is *representing* the dispute (dual claims, visible boundaries), not resolving it.

---

10. Conclusion

Location is the load-bearing dimension of property search, and in fast-growing cities it behaves like a stream, not a table: entities appear before maps know them, change type while data models assume they cannot, and dissolve while their pages still rank. This paper described a system built around that reality: a per-market typed location graph fed by a cascade of LLM research agents, disciplined by a three-region confidence gate in the record-linkage tradition, defended at the listing gate, and audited by humans exactly where the machine admits uncertainty.

The transferable lesson is not the graph schema or the agent roster; it is the governing rule that made the system operable: **automation earns autonomy per decision and per confidence level, and everything else is a recommendation to a human, delivered with evidence.** Systems that adopt the agents without the audits will have built the failure Section 6 warns about. The pilot of Section 6.1 is small, but it is public and repeatable. Extending it (more markets, listing-assignment tasks, multi-adjudicator labels) is the most useful thing we or anyone else can do next, and we would like to see other operators publish theirs.

References

1. GeoNames geographical database. geonames.org
2. Who's On First: a gazetteer of places. whosonfirst.org
3. OpenStreetMap. openstreetmap.org
4. 2GIS. 2gis.com
5. Overture Maps Foundation. *Overture Maps data and the Global Entity Reference System (GERS).* overturemaps.org · docs.overturemaps.org/gers
6. Uber Engineering. *H3: Uber's Hexagonal Hierarchical Spatial Index.* uber.com/blog/h3
7. Pelias: open-source geocoder. pelias.io
8. Dubai Municipality. *Makani: the official geographic addressing system of Dubai.* makani.dubai.ae
9. Fellegi, I.P., Sunter, A.B. *A Theory for Record Linkage.* Journal of the American Statistical Association 64(328), 1969. doi.org/10.1080/01621459.1969.10501049
10. Winkler, W.E. *Using the EM Algorithm for Weight Computation in the Fellegi–Sunter Model of Record Linkage.* Proceedings of the Section on Survey Research Methods, ASA, 1988. asasrms.org PDF
11. Christen, P. *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection.* Springer, 2012. doi.org/10.1007/978-3-642-31164-2
12. Elmagarmid, A.K., Ipeirotis, P.G., Verykios, V.S. *Duplicate Record Detection: A Survey.* IEEE TKDE 19(1), 2007. doi.org/10.1109/TKDE.2007.250581
13. Koumarelas, I., Kroschk, A., Mosley, C., Naumann, F. *Experience: Enhancing Address Matching with Geocoding and Similarity Measure Selection.* ACM JDIQ 10(2), 2018. doi.org/10.1145/3232852
14. Leidner, J.L. *Toponym Resolution in Text.* Universal-Publishers, 2008 (PhD thesis, University of Edinburgh, 2007).
15. Gritta, M., Pilehvar, M.T., Limsopatham, N., Collier, N. *What's Missing in Geographical Parsing?* Language Resources and Evaluation 52, 2018. doi.org/10.1007/s10579-017-9385-8
16. Gritta, M., Pilehvar, M.T., Collier, N. *Which Melbourne? Augmenting Geocoding with Maps.* ACL 2018. aclanthology.org/P18-1119
17. Hu, X., et al. *Location Reference Recognition from Texts: A Survey and Comparison.* ACM Computing Surveys 56(5), 2023. doi.org/10.1145/3625819
18. Dsouza, A., Tempelmeier, N., Yu, R., Gottschalk, S., Demidova, E. *WorldKG: A World-Scale Geographic Knowledge Graph.* CIKM 2021. arxiv.org/abs/2109.10036
19. Janowicz, K., et al. *Know, Know Where, KnowWhereGraph.* AI Magazine 43(1), 2022. doi.org/10.1002/aaai.12043
20. Karalis, N., Mandilaras, G., Koubarakis, M. *Extending the YAGO2 Knowledge Graph with Precise Geospatial Knowledge.* ISWC 2019. doi.org/10.1007/978-3-030-30796-7_12
21. *LLM-Empowered Knowledge Graph Construction: A Survey.* arXiv:2510.20345, 2025. arxiv.org/abs/2510.20345
22. Huang, L., et al. *A Survey on Hallucination in Large Language Models.* ACM TOIS 43(2), 2025. arxiv.org/abs/2311.05232
23. Han, S., et al. *LLM Multi-Agent Systems: Challenges and Open Problems.* arXiv:2402.03578, 2024. arxiv.org/abs/2402.03578
24. Tian, K., et al. *Just Ask for Calibration: Strategies for Eliciting Calibrated Confidence Scores from Language Models Fine-Tuned with Human Feedback.* EMNLP 2023. arxiv.org/abs/2305.14975
25. Xiong, M., et al. *Can LLMs Express Their Uncertainty? An Empirical Evaluation of Confidence Elicitation in LLMs.* ICLR 2024. arxiv.org/abs/2306.13063
26. Zheng, L., et al. *Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena.* NeurIPS 2023. arxiv.org/abs/2306.05685
27. Wang, P., et al. *Large Language Models are not Fair Evaluators.* ACL 2024. arxiv.org/abs/2305.17926
28. Panickssery, A., Bowman, S.R., Feng, S. *LLM Evaluators Recognize and Favor Their Own Generations.* NeurIPS 2024. arxiv.org/abs/2404.13076
29. Verga, P., et al. *Replacing Judges with Juries: Evaluating LLM Generations with a Panel of Diverse Models.* arXiv:2404.18796, 2024. arxiv.org/abs/2404.18796
30. Turpin, M., Michael, J., Perez, E., Bowman, S.R. *Language Models Don't Always Say What They Think:*

Unfaithful Explanations in Chain-of-Thought Prompting. NeurIPS 2023. arxiv.org/abs/2305.04388
31. Platt, J.C. *Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods.* In Advances in Large Margin Classifiers, MIT Press, 1999. [Author copy via Microsoft Research; widely mirrored.]
32. Zadrozny, B., Elkan, C. *Transforming Classifier Scores into Accurate Multiclass Probability Estimates.* KDD 2002. doi.org/10.1145/775047.775151
33. Parasuraman, R., Riley, V. *Humans and Automation: Use, Misuse, Disuse, Abuse.* Human Factors 39(2), 1997. doi.org/10.1518/001872097778543886
34. Parasuraman, R., Manzey, D.H. *Complacency and Bias in Human Use of Automation: An Attentional Integration.* Human Factors 52(3), 2010. doi.org/10.1177/0018720810376055
35. Mathew, M., Karatzas, D., Jawahar, C.V. *DocVQA: A Dataset for VQA on Document Images.* WACV 2021. arxiv.org/abs/2007.00398
36. Kim, G., et al. *OCR-free Document Understanding Transformer.* ECCV 2022. arxiv.org/abs/2111.15664
37. Ouyang, L., et al. *OmniDocBench: Benchmarking Diverse PDF Document Parsing with Comprehensive Annotations.* CVPR 2025. arxiv.org/abs/2412.07626
38. Lewis, P., et al. *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.* NeurIPS 2020. arxiv.org/abs/2005.11401
39. Dong, X.L., et al. *AutoKnow: Self-Driving Knowledge Collection for Products of Thousands of Types.* KDD 2020. doi.org/10.1145/3394486.3403323
40. Yelp Engineering. *Seeing Double on Yelp.* 2015. engineeringblog.yelp.com
41. Wei, X. *Contextualizing Airbnb by Building Knowledge Graph.* Airbnb Tech Blog, 2018. medium.com/airbnb-engineering
42. Davis, D., et al. *Transforming Location Retrieval at Airbnb.* Airbnb Tech Blog, 2024. medium.com/airbnb-engineering
43. AirbnbEng. *Behind the Scenes: Airbnb Neighborhoods.* Airbnb Tech Blog, c. 2013. medium.com/airbnb-engineering
44. Zillow. *Neighborhood boundary shapefiles for US cities* (CC BY-SA), 2008; distribution retired. Contemporary coverage: gisuser.com
45. Placekey. *A Free and Open Universal Standard Identifier for Physical Places*: encoding specification. placekey.io/technical-whitepaper
46. Foursquare. *Foursquare Open Source Places.* 2024. foursquare.com/resources/blog
47. Lookingbill, A., Russell, E. *Google Maps 101: How We Map the World.* Google, 2019. blog.google
48. Google Search Central. *Creating Helpful, Reliable, People-First Content.* developers.google.com/search
49. OpenStreetMap Foundation. *ODbL Community Guidelines, including the Geocoding Guideline: Collective Database, Substantial Extract, Produced Work.* osmfoundation.org/wiki/Licence/Community_Guidelines
50. Hollenstein, L., Purves, R. *Exploring Place Through User-Generated Content: Using Flickr Tags to Describe City Cores.* Journal of Spatial Information Science 1, 2010. doi.org/10.5311/JOSIS.2010.1.3
51. Haklay, M. *How Good is Volunteered Geographical Information? A Comparative Study of OpenStreetMap and Ordnance Survey Datasets.* Environment and Planning B 37(4), 2010. doi.org/10.1068/b35097
52. Yin, X., Han, J., Yu, P.S. *Truth Discovery with Multiple Conflicting Information Providers on the Web.* IEEE TKDE 20(6), 2008. doi.org/10.1109/TKDE.2007.190745
53. Li, Y., et al. *A Survey on Truth Discovery.* ACM SIGKDD Explorations 17(2), 2016. doi.org/10.1145/2897350.2897352
54. Dong, X.L., et al. *Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion.* KDD 2014. doi.org/10.1145/2623330.2623623

Web references accessed July 2026. Industry engineering blogs [40–43, 46–47] are cited as grey literature: primary accounts of production systems, not peer-reviewed work. Reference [44] is a contemporaneous third-party report; Zillow's original announcement pages are no longer online.

Appendix A. Reproducibility of the Dubai OSM counts (Section 3.8)

Executed 2026-07-05/06 against the main public Overpass instance (`overpass-api.de`; data timestamp 2026-07-05T21:43Z for the first two queries). Mirrors lag the main instance by differing amounts, so the endpoint matters; and the area filter selects by `name:en` tag, which is retag-fragile; pinning the emirate by its OSM relation ID (`area(3600000000 + rel_id)`) is the form that survives retagging, for anyone re-running these.

Boundary relations at sub-emirate levels:

```
[out:json][timeout:60];
area["name:en"="Dubai"]["admin_level"="4"]->.dubai;
( relation["boundary"="administrative"]["admin_level"~"^(9|10|11)$"](area.dubai); );
out count;   // → relations: 130
```

Place point features:

```
[out:json][timeout:60];
area["name:en"="Dubai"]["admin_level"="4"]->.dubai;
( node["place"~"^(suburb|neighbourhood|quarter)$"](area.dubai); );
out count;   // → nodes: 322
```

Place features carrying geometry without administrative status (quantifying caveat (a) below):

```
[out:json][timeout:90];
area["name:en"="Dubai"]["admin_level"="4"]->.dubai;
( way["place"~"^(suburb|neighbourhood|quarter)$"](area.dubai);
  relation["place"~"^(suburb|neighbourhood|quarter)$"](area.dubai); );
out count;   // → ways: 235, relations: 75, total: 310
```

Methodology caveats: (a) place features mapped as ways/relations without `admin_level` sit outside the boundary-relation count; the third query bounds this at 310, which is why Section 3.8 reads the counts as "geometry of uneven authority," not "no geometry"; (b) some counted nodes are `label`/`admin_centre` members of the counted relations, overstating point-only places; (c) the UAE has no settled sub-emirate `admin_level` convention on the OSM wiki, so the 9–11 filter reflects observed tagging, not an agreed standard; (d) the point-feature and way/relation place counts are not deduplicated against each other (OSM sometimes maps one place as both), so the two sets overlap by an unmeasured amount, which is why Section 3.8 phrases the coverage claim as "roughly half" rather than as a ratio of these exact denominators. Two further notes. The published extracts are ODbL-licensed OSM derivatives: © OpenStreetMap contributors, share-alike applies to the data files shipped with the pilot (code, data extracts, and adjudication samples: fbritoferreira.com/research/the-living-map). The vandalized name tag found during adjudication (node 6494686786) has been flagged for upstream correction, as community etiquette requires of anyone measuring the map.

The claims these counts support are deliberately coarse (roughly half of named places carry no geometry, and administrative-boundary status is rare and convention-fragile) and survive all three caveats; specific ratios should not be quoted without them.

Appendix B. The SEO Dividend: The Graph, Rendered

Location pages have a second consumer besides users: search engines. The graph is an organic-search asset because what ranking systems reward [48] is what a location graph produces and a manual content operation cannot sustain.

Uniqueness at scale. The classic failure of marketplace SEO is thousands of near-identical area pages (same template, swapped place names), which modern ranking systems treat as thin content. Graph-assembled pages are structurally different per location because the *data* differs per location: this community's actual POIs, unit mix, transaction history, adjacencies, contained buildings. Depth stops being a copywriting cost

and becomes a query. Programmatic page generation is also the pattern scaled-content-abuse policies target. The line those policies draw is value, not volume. Graph-backed uniqueness is a *defense available*, not a safe harbor granted: it holds only while each page's data genuinely differs and genuinely serves the query, and a team that scales templates faster than the graph's depth earns the penalty the policy exists for.

Freshness for free. Manually maintained area guides go stale the way the location database does. Graph-driven pages inherit the graph's liveness: when a fact changes, every page assembled from that node updates with it. The freshness signal is a side effect of a pipeline that was already running.

Structure machines can read. Typed nodes map nearly one-to-one onto structured-data vocabularies (schema.org Place and kin), so location pages carry machine-readable markup generated from the graph rather than hand-maintained. The same typing that disambiguates "Meydan" for the resolver tells crawlers what kind of entity each page describes.

First to the query. The pre-map channel (Section 3.3) has a demand-side payoff: search interest in a development starts at announcement, and the portal whose graph creates the node that day has a genuine, indexable page live while the location exists nowhere else, capturing the early query market at the moment buyer intent peaks.

Architecture crawlers can walk. The graph's edges are an internal-linking scheme: containment links areas to communities to buildings; adjacency links siblings; developed-by links projects to developer hubs. Internal linking that mirrors real geography gives crawlers a coherent site structure and distributes authority the way the market actually thinks. One obligation attaches: pages assembled from OSM-derived facts are Produced Works under the ODbL, and the "© OpenStreetMap contributors" attribution must render visibly on them (Section 4.2).