



UPPER Model & Algorithm Transparency Statement

Version 1.0 · 10 October 2026 · Reference UPPER-TS-1.0 · Prepared from the production code and database on this date

What this document is. A plain-language, verifiable description of how UPPER finds and scores candidates for your requisitions: the data we hold, the model we use, how the score is calculated, how location is handled, and what we have and have not yet done to test the system for fairness. Every statement in sections 2 to 8 is traced to a specific source file and line in the appendix. This is a disclosure, not a legal opinion, and it does not claim that the system is "bias-free" or "compliant"; it tells you exactly what the system does so you and your advisors can judge.

1. Scope

UPPER is a **candidate sourcing** service. For each requisition you create, UPPER searches public and licensed professional sources, builds a list of people who appear to fit, produces a numeric fit score with written reasons, and contacts those people to ask whether they are interested. **UPPER does not receive or process job applications and does not make hiring decisions.** UPPER does rank people and applies one automated threshold before showing them to you (section 4.3); that threshold is disclosed here as an automated filter. Every person UPPER shows you is then reviewed and advanced (or not) by a human in your account.

2. What UPPER collects about a candidate

UPPER's candidate record holds the following fields, verified against the production database schema on 10 October 2026:

Category	Fields held
Identity and contact	Name, work email, phone, LinkedIn profile URL, GitHub profile URL (when public)
Professional	Current title and employer, skills as listed on the public profile, professional bio or headline, CV file (only when the candidate sends one), stated work-authorization status (only when the candidate states it)
Location	Location as stated on the public profile; derived city and state
Process	Source channel, score and written rationale, stage, outreach history, replies, opt-out timestamp
Display only	Profile photo, stored in a separate photo table and shown on the Results and Candidates pages. The photo is not given to the scoring model (section 4.4).

No demographic fields. UPPER's database has no column for sex, gender, race, ethnicity, age, date of birth, religion, disability, veteran status, sexual orientation, or national origin, and no code that sets out to infer them (a search of every column name in the production database for these terms found no candidate field). Profile text, CVs and names can nevertheless carry clues to such characteristics, and the language model reads that text. Whether that produces any measurable effect on scores is exactly what the planned adverse-impact testing is for (section 8).

Sources. Candidates come from, in priority order: UPPER's own pool of previously sourced people, LinkedIn, GitHub, Hacker News, Reddit, Apollo, Hunter, Prospeo, and (for eligible requisitions) Ladders.

3. The model

- **Language model:** Anthropic Claude Haiku 4.5, pinned to release `claude-haiku-4-5-20251001`. The model version does not change without a code change.
- **Role of the model:** it **judges each criterion**. For each one it returns a rating on a fixed five-step scale (none, weak, partial, strong, exceptional), a **verbatim quote** from the candidate's material that supports the rating, or "unknown" when the material does not allow a judgement. It is instructed that "inventing a total is the one thing that fails this task". Its ratings determine the points; an instruction is not a guarantee of accuracy, which is why the audit in section 8 matters.
- **Role of the code:** a deterministic arithmetic module (the "kernel") turns those ratings into the number. The same ratings always give the same score. The kernel makes no network calls and reads no database.
- **Status:** the kernel has been the live scorer for every account since 26 August 2026 (switch `SCORER_V2_KERNEL` is on for all tenants). Of the scored records created since 1 September 2026 that carry an engine label, all are labelled as kernel-scored and none as the older path. If the model does not return the required format or its service is unavailable, the engine falls back to an older single-number path or a provisional placeholder score; no stored record carries either marker.

4. How the Rockstar Score is calculated

4.1 The eight criteria

Criterion	Kind	Base weight (of 100)	Observable at Gate 1 (sourcing)?
Must-have skills demonstrated	Hard	30	Yes
Role / function fit	Hard	14	Yes
Seniority congruence	Hard	12	Yes
Depth and recency of the relevant work	Hard	14	No (needs a CV)
Scope and level of ownership	Soft	8	No
Domain or industry transfer	Soft	10	Yes
Evidenced outcomes	Soft	6	No
Pattern of engagements	Soft	6	No

"Soft" weights shift by a few points depending on the role family and are re-normalised so the soft total stays at 30. For the engineering family, the only one in use today, the effective Gate-2 weights are 30, 14, 12, 14, 8, 12, 6 and 4 in table order. Gate 1 uses only the four criteria observable from sourcing evidence (must-have skills 30, function fit 14, seniority 12, domain transfer 12); because the score is a share of the observed weight (section 4.2), those four are scaled to a 0–100 result. Other families are declared in code and would take effect if a requisition were classified into them.

4.2 The arithmetic

- Each rating maps to a fraction: none 0, weak 0.25, partial 0.55, strong 0.80, exceptional 1.00.
- For must-have skills, the fraction is averaged with the real ratio of must-haves matched to must-haves stated (3 of 7 contributes 0.43, not a bucket).
- **Score = points earned ÷ weight of the criteria actually observed × 100.** A criterion the model could not judge is marked *unknown* and **leaves the denominator**: it neither rewards nor penalises that criterion. A thin profile is therefore not given zeros for what we could not see, although scores built on different amounts of evidence are not perfectly comparable, which is why coverage is shown beside them. If **nothing** about a person could be judged, the score is 0, the person is not shown, and the record is flagged as our evidence gap (section 4.3).
- **Coverage** (observed weight ÷ total weight) is published beside the score as a grade: A at 90% or more, B at 80%, C at 70%, D below. Below 70% the score is labelled provisional. The kernel also returns a range around the score whose half-width grows from ±2 at full coverage to ±20 at none; it reflects how much we saw and is **not** a statistical confidence interval.
- **Gate 1 (sourcing evidence) is capped at 82**; the top band is reachable only after a CV (Gate 2), when all eight criteria are in play.
- Scores are stored to one decimal and shown as whole numbers.

4.3 The reveal threshold

Candidates scoring below **40** at Gate 1 are not shown in your account. This is the one point at which the score alone removes a person from view, and it is an automated filter. A person about whom nothing could be judged scores 0 and is also not shown, so missing information can reduce a person's visibility. When coverage is below 60% the record is flagged `needs_enrichment`, meaning we suspect our own retrieval fell short; the flag records the suspicion, it does not prove the cause.

4.4 What carries no weight, and what the model can still see

- **Name.** The name is normalised to plain ASCII (accents and symbols removed) before it reaches the model. It is not a criterion, but the prompt does not instruct the model to ignore it and it remains visible, so a name-related effect on ratings cannot be ruled out from the code alone. Removing the name from the scoring prompt is planned (section 8).
- **Photo.** The scoring prompt contains no photo, image or avatar field.
- **Location.** There is no location criterion in the kernel, so no points are assigned for it. The candidate's and the requisition's locations are passed to the model as context, so an indirect effect on its ratings has not been ruled out and is in scope for the audit. Whether a person is considered at all is decided separately by the location gates in section 5.
- **Work authorization.** The model is instructed that it "is NOT a criterion and NOT a scoring axis". If stated, it is recorded; unknown is treated as the normal answer. A candidate who *states* ineligibility against a requisition that *clearly restricts* is still scored but marked not eligible for submission, with a reason.

- **Email, phone and endorsement counts** are not in the prompt; endorsement counts are dropped from the skills list before scoring. Employer names and the candidate's own profile text are in the material the model reads, as they must be to judge the criteria.

5. Location: a feasibility gate, not a quality score

- **You choose the radius** when creating a requisition: 25, 50, 65 (default), 100, 150 or 250 miles from the requisition's city, or "anywhere in the state". Fully remote requisitions skip the radius; the country rule below still applies, and a remote role tied to a named city keeps a preference for people near that city.
- **Distance is measured by coordinates** from a city-level atlas, not by ZIP code. ZIP codes are not scored and are not used to filter; where a profile's location text contains one (e.g. "TX 78701") it is read only to find the state. The location text itself is stored as the candidate wrote it, so it may contain a ZIP.
- **How strict the gate is depends on your wording.** A requisition marked on-site with no exception ("onsite NYC, no exceptions") is *strict*: people outside the radius are excluded. Hybrid, "preferred", or any conditional wording makes it *preferred*: people outside the radius are kept, not excluded, and their score does not change. When the system cannot classify the wording it defaults to *preferred*; it never defaults to strict.
- **Unknown is not excluded**, with one exception. If a candidate's location cannot be resolved to a place, the gate passes them through as "unverified", unless the requisition is strict, tied to a specific city, and the distance that could be measured is definitively outside the radius.
- **Country match.** A candidate whose stated location is in a different country from the requisition is excluded. This is based on where the person says they are, not on citizenship or origin. When a requisition names no country, the United States is assumed.
- **Canada.** To comply with Canada's anti-spam law, UPPER never emails a person whose profile indicates Canadian residence; such people may still be contacted through LinkedIn's own messaging.
- **Sanctions list.** A sanctioned-country check exists in code; it is currently in *shadow mode* (logs matches, blocks nobody) for every account.

6. Exclusions you define

You may list up to 20 job titles or phrases to exclude. An exclusion applies to the candidate's **current role only**; a past job never excludes anyone. The model's exclusion verdict is advisory: code accepts it only if the term is one of your exact words, the role is current, and a verbatim quote of at least 12 characters is found in the candidate's material. When no proof is found the candidate is kept and labelled "unknown".

Disclosure. Exclusion terms are free text and are not yet screened automatically for words that can stand in for a protected characteristic (for example terms about age or graduation date). Terms pre-filled from a job description are not screened either. Automatic screening with a refusal message is planned (section 8). Until then, enter only job-related terms and review anything pre-filled from a job description.

7. Human oversight and candidate rights

- Every candidate moves through visible stages: Reviewed, Pooled, Screened, Engaged, Considering, Interested, Submitted, Placed. Outreach to Screened candidates is automatic (that is the Engaged stage).

Movement from Considering onward requires a human action in your account or the candidate's own reply.

- Every score carries the model's per-criterion ratings and the quoted evidence, so a reviewer can see why a number is what it is.
- Candidates can opt out of all contact at any time; the opt-out is recorded on their record and honoured across all your requisitions.
- Your users can rate and re-order candidates; those ratings are stored per account and never feed back into the score.

8. Known limitations and what is planned

Item	Status on 10 October 2026
Independent bias audit	Not yet performed. UPPER is selecting an independent auditor.
Adverse-impact testing of the score and the location gate	Not yet performed. Planned as part of the audit.
Product safeguards in progress	Automatic screening of exclusion terms for protected-class proxies, with a refusal message (section 6); removal of the candidate's name from the scoring prompt (section 4.4).

This statement is re-issued when any of the above changes, or when the criteria, weights, model version, reveal threshold or location rules change.

9. Version history

Version	Date	Change
1.0	10 Oct 2026	First issue. Derived from engine commit <code>70dc03f</code> and the production database on this date. Issued 10 October 2026.

UPPER · UpperLev Inc. · This statement describes the system as built on the date shown. It is re-issued when the scoring criteria, weights, model version, reveal threshold, or location rules change. Questions: compliance@upperlev.com. Reference UPPER-TS-1.0.

Appendix A. Evidence map

Repository `upperlev/upper-migration-node`, commit `70dc03f` (10 Oct 2026). Paths are under `crew6_integration/runtime/services/`. Database: production Postgres, read-only queries run 10 Oct 2026.

Statement	Source
Eight criteria, weights, Gate-1 observability	<code>scoreKernel.js</code> lines 49–67 (CRITERIA)
Rating scale none/weak/partial/strong/exceptional	<code>scoreKernel.js</code> line 72 (ORDINAL)

Soft-weight family offsets and re-normalisation	<code>scoreKernel.js</code> lines 78–89, 117–131
Gate-1 cap of 82	<code>scoreKernel.js</code> line 91 (<code>GATE1_MAX_PUBLISHABLE</code>)
Provisional below 70% coverage	<code>scoreKernel.js</code> line 97 (<code>PROVISIONAL_BELOW</code>)
Reveal threshold 40; enrichment-suspect below 60%	<code>scoreKernel.js</code> lines 109–110
Unknown leaves the denominator; must-have facet ratio; weighted mean	<code>scoreKernel.js</code> lines 146–199
Band ± 2 to ± 20 ; one-decimal storage; whole-number display	<code>scoreKernel.js</code> lines 200–212
Model pinned to Claude Haiku 4.5	<code>candidateScorer.js</code> line 52 (<code>MODEL_ID</code>)
Model forbidden from producing a total; verbatim-quote rule; unknown-not-none rule	<code>candidateScorer.js</code> lines 788–823 (<code>_findingsInstruction_</code>)
Work authorization not a criterion	<code>candidateScorer.js</code> lines 812–814
Fields given to the model (no photo field)	<code>candidateScorer.js</code> lines 657–690 (prompt builder)
Name normalised to ASCII, non-evidential	<code>candidateScorer.js</code> lines 587–603 (<code>_promptSafeName_</code>)
Location is not a scoring axis under the kernel	<code>candidateScorer.js</code> lines 1515–1519
Fallback paths when the model does not comply or is unavailable	<code>candidateScorer.js</code> lines 55, 825–827, 1235–1248, 1454, 1490
Kernel on for all tenants since 26 Aug 2026	DB <code>feature_flags</code> , <code>SCORER_V2_KERNEL='true'</code> on every tenant (earliest 2026-08-26)
Engine labels on scored records since 1 Sep 2026; no fallback or placeholder marker stored	DB <code>sequence_queue</code> , grouped by <code>score_engine</code> , <code>created_at >= '2026-09-01'</code>
Radius options and 65-mile default	Account page <code>account.html</code> lines 5594–5600; <code>geoGate.cjs</code> line 22; <code>hr136LocationFilter.js</code> lines 61, 96–115
Coordinates from a city-level atlas; fail-open on unknown place	<code>geoGate.cjs</code> lines 1–16, 22
ZIP parsed only to read the state	<code>sourcing/locationEligibilityFilter.js</code> lines 104–105

Strict vs preferred vs flexible; never defaults to strict	<code>candidateScorer.js</code> lines 2079–2175 (<code>_parseLocationFlexibility_</code>)
Preferred/flexible never hard-exclude; strict excludes outside radius	<code>candidateScorer.js</code> line 2497 ff. (<code>_applyLocationPolicy_</code>), T-6967 block
Country-match gate; US default when the requisition names no country	<code>candidateScorer.js</code> line 2497 ff., D-REMOTE-COUNTRY and T-736 blocks; flags <code>REMOTE_COUNTRY_GATE_ENABLED</code> , <code>REQ_COUNTRY_US_DEFAULT</code> (absent → on)
Canada: no email, LinkedIn permitted	<code>canadaResident.js</code> lines 1–28
Sanctions check in shadow mode	<code>candidateScorer.js</code> lines 2318–2330 (<code>_complianceCheck_</code>); DB <code>feature_flags</code> : <code>COMPLIANCE_DENYLIST_ENABLED</code> absent (default false), <code>COMPLIANCE_SHADOW_LOG='TRUE'</code>
Exclusions: current role only, exact term, 12-character verbatim quote, cap 20	<code>candidateScorer.js</code> lines 693–720 (T-9831)
Sourcing channels and order	<code>sourcing/waveDriver.js</code> lines 1567–1581
Candidate fields held	DB <code>information_schema.columns</code> for <code>candidate_pool</code> and <code>sequence_queue</code>
No demographic columns	DB <code>information_schema.columns</code> searched for gender/sex/race/ethnic/age/birth/dob/religion/disab/veteran/orientation/national: no matching candidate field (five unrelated column names were false positives)
Photo stored separately	DB table <code>candidate_photo</code> (photo bytes, source URL, <code>fetches_at</code>); no photo field in the scoring prompt
Country default when unnamed; strict-city exception for unknown location	<code>candidateScorer.js</code> <code>_applyLocationPolicy_</code> , T-736 and T-6967 blocks
Exclusion terms not screened for proxies	Account page Exclude box and <code>parse-jd</code> <code>skills_exclude</code> ; no screening routine in <code>candidateScorer.js</code> T-9831 block