PRODUCT REQUIREMENTS DOCUMENT

NYC 311 Trends

What's getting worse in NYC, where, and which agencies are improving or deteriorating in response time

Version: 1.0 (draft for engineering review)

Status: Proposed

Owner: Senior PM / Technical Lead

Audience: Engineering team; product stakeholders

Specifies product objective, users, workflows, functional & data requirements, the analytical methodology for distinguishing reporting increases from genuine worsening and response-performance trends, visualization and insight-generation requirements, performance and quality expectations, risks, acceptance criteria, and a suggested technical architecture. No application is to be built from this PRD alone.

PRD — NYC 311 Service Request Trends

Product Requirements Document

Product name	NYC 311 Trends (working title)
Version	1.0 (draft for engineering review)
Status	Proposed
Owner	Senior PM / Technical Lead
Audience	Engineering team; product stakeholders

1. Executive Summary

NYC 311 is the city's non-emergency service-request system. Every day it records hundreds of thousands of complaints across dozens of agencies and complaint types, each with a location, a creation time, a closure time, and an agency owner. This dataset is a uniquely rich, continuously-updated signal of **what the city's residents are experiencing** — but raw, it is a ~30-million-row CSV that is nearly impossible to reason about directly.

NYC 311 Trends is a web application that ingests the raw NYC 311 Service Requests dataset and turns it into an explorable, evidence-based view of three questions:

1. **What kinds of problems are getting worse in NYC?**
2. **Where are they getting worse?** (borough, community district, neighborhood)
3. **Which agencies appear to be improving or deteriorating in response time?**

The product's core intellectual contribution is **rigorous trend interpretation**: it does not just count complaints and plot lines. It distinguishes, with explicit, transparent logic, between (a) *more complaints because more people are active or reporting more*, (b) *genuinely worsening underlying conditions*, and (c) *worsening agency response performance*. Every "worsening" claim the product surfaces is labeled with the evidence for that claim and its confidence — never a bare number.

This PRD specifies the product's objective, users, workflows, functional and data requirements, metrics and analytical definitions, visualization and insight-generation requirements, performance and quality expectations, and a suggested technical architecture. **It is a specification document only — no application is to be built from this PRD alone.**

2. Product Objective

To make the NYC 311 dataset legible and decision-useful — to answer, in plain language backed by transparent methodology, where and how the city's service conditions and agency performance are changing over time.

The product succeeds when a user can, in under five minutes, go from a question like *"Are heat complaints getting worse in the Bronx, and is HPD getting slower to respond?"* to a defensible, explained answer.

Success metrics (product-level):
- **Answer time:** median time from question to defensible answer ≤ 5 minutes.
- **Claim confidence:** every surfaced insight carries a stated confidence and evidence path; 100% of

automated insights include an "why this is real vs. a reporting artifact" explanation.
- **Adoption:** target users return weekly for trend monitoring.

---

3. Target Users

User	Role	Primary need
City agency analysts (HPD, DEP, DOT, DSNY, NYPD, DOB, Parks…)	Monitor their agency's portfolio	"Is my agency's response time getting better or worse, and where?"
Community board / neighborhood advocates	Monitor their district	"What is getting worse in *my* community, and is the city responding?"
Journalists & researchers	Investigative analysis	"Find me the surprising or worsening patterns, with method I can cite."
Policy / elected-office staff	Oversight & budgeting	"Which conditions are worsening enough to warrant action or budget?"
Product PM / evaluator (internal)	QA the insight engine	"Are the automated insights correct, and are they labeled honestly?"

Assumed user skill: comfortable with web dashboards and basic filtering; **not** assumed to be a data scientist or SQL-literate. Methodological transparency matters more than statistical depth in the UI.

---

4. Key User Questions

The product must be able to answer (with evidence):

1. What is the volume of complaints per complaint type, and how is it trending?
2. Which complaint types are **increasing** (volume and/or normalized rate) — and is the increase real or a reporting artifact?
3. Which complaint types are **decreasing** (improving)?
4. Where (borough / community district / neighborhood) is a given complaint type getting worse?
5. Which **agencies** are improving or deteriorating in **response time** (and resolution time)?
6. Which agencies have growing **backlogs** (open complaints aging)?
7. What is **unusual** right now — anomalies, emerging types, geographic clusters, seasonality breaks?
8. How do current conditions compare to the same period last year (trailing 12-month vs. prior 12-month)?

---

5. Primary Workflows

WF-1 — "What's getting worse?" (monitoring)
User lands on the dashboard → sees top worsening complaint types with normalized-rate trend and confidence labels → filters to a borough/district → sees geographic breakdown → exports a report.

WF-2 — "Is my agency getting slower?" (agency performance)
User selects an agency → sees response-time trend (median / p95 close time), backlog trend, and geographic variance → compares to city-wide baseline → drills into a specific complaint type.

WF-3 — "What's unusual?" (surprise detection)
User opens the "Anomalies" view → sees auto-detected unusual patterns (spikes, emerging types, clusters) → each with an explanation and an artifact-check → saves/investigates.

WF-4 — "Is this real?" (methodological drill-down)
On any worsening claim, user expands the "Why this claim" panel → sees normalized vs. raw, the trend test, the confidence, and the caveats (reporting-channel changes, seasonality).

WF-5 — Custom comparison / research
User builds a custom filter set (complaint type × district × time range) → sees the trend and comparison → exports the underlying data for external analysis.

---

6. Functional Requirements

Priorities: **P0** (must ship), **P1** (should ship), **P2** (nice to have).

FR-1 (P0) Data ingestion & refresh. Ingest the NYC 311 CSV into a queryable store; support initial full load and periodic incremental refresh (daily). Documented pipeline; idempotent re-runs.

FR-2 (P0) Trend dashboard. Top-level view of complaint volume, complaint types, boroughs, agencies, response/resolution times, and change over time. Filterable by time range, complaint type, agency, borough, district.

FR-3 (P0) Worsening/improving identification. Automatic identification of complaint types, geographies, and agencies that are worsening or improving, using the metric definitions in §9. Every claim surfaces its confidence and the underlying evidence.

FR-4 (P0) Agency response-time view. Agency-level median/p95 response time, resolution time, and backlog, with trend and comparison to city baseline.

FR-5 (P0) Drill-down & filtering. Click-through from any chart to filtered views; cross-filtering across dimensions (type × geography × agency × time).

FR-6 (P0) Anomaly detection. Automated detection of unusual patterns (spikes, emerging types, clusters, seasonality breaks), each with explanation.

FR-7 (P1) Export. Export of filtered data (CSV) and rendered reports (PDF/image).

FR-8 (P1) Alerts. (Optional) Scheduled alerts for a user's saved filters ("heat complaints in Bronx exceeded threshold").

FR-9 (P1) Community-district & neighborhood geography. Drill-down below borough to community district and neighborhood where the data supports it.

FR-10 (P2) Geospatial map. Choropleth/bubble map of complaint intensity and response time by district.

FR-11 (P2) Custom annotation / sharing. Save views, share links.

---

7. Data Requirements

7.1 Source

NYC 311 Service Requests dataset, published on NYC Open Data (SOCRATA). The raw CSV export is the canonical input. **Assumption:** the raw export can be several GB (on the order of 20–40M rows over the full history); see Performance (§13).

7.2 Core schema (fields we rely on; exact names confirmed from the source manifest)

Field (canonical)	Description	Use
`unique_key`	Row identifier	Dedup / primary key
`created_date`	Complaint opened (timestamp)	Time dimension
`closed_date`	Complaint closed (timestamp; null if open)	Response/resolution time
`agency` / `agency_name`	Owning agency code / name	Agency dimension
`complaint_type`	Category (e.g., Heat/Hot Water)	Complaint-type dimension
`descriptor`	Subcategory detail	Drill-down
`borough`	Borough name	Geography
`community_district`	Community district	Geography
`incident_zip`, `incident_address`, `street_name`, `city`	Location detail	Address-level (PII-scoped, see §18)
`latitude`, `longitude`	Coordinates	Geospatial
`open_data_channel_type`	Reporting channel (Phone, Mobile, Web…)	Reporting-bias control
`status`	Open / Closed / Pending	Backlog
`resolution_description`	Resolution note	Analysis (free text, P2)

7.3 Data volume & cardinality assumptions

- Full-history rows: **tens of millions** (order 20–40M).
- Complaint types: ~**dozens** of top-level categories.
- Agencies: ~**20–30**.
- Boroughs: **5**; community districts: ~**60**; ZCTAs/neighborhoods: ~**200+**.

---

8. Data Cleaning & Normalization Requirements

D-1 (P0) Dedup & key integrity. `unique_key` must be unique; drop or quarantine duplicate/null-key rows. Log the count removed.

D-2 (P0) Timestamp parsing & validation. Parse `created_date` / `closed_date` with documented format handling. Quarantine invalid timestamps. Compute derived fields:
- `response_hours` = `closed_date − created_date` (hours), defined only when `closed_date` present.
- `is_open` = `closed_date` is null.
- `backlog_age_days` = now – `created_date` for open records.

D-3 (P0) Categorical normalization. Map `agency` / `agency_name` to a canonical agency key (handle code/name mismatch and historical renames). Map `complaint_type` to canonical categories; preserve original for audit.

D-4 (P0) Geography normalization. Normalize `borough` to canonical 5 + "Unspecified/Other". Map `community_district` to canonical codes. Keep rows with missing geography in a visible "unspecified" bucket — never silently drop.

D-5 (P0) Reporting-channel normalization. Canonicalize `open_data_channel_type` (Phone, Mobile, Web, etc.) — this is the primary control for reporting-bias analysis (§9).

D-6 (P0) Outlier handling for response time. Response-time percentiles are robust to outliers, but for sanity: cap implausible values (e.g., negative durations, `closed_date` before `created_date` → quarantine; duration > N years → flag, exclude from response-time metrics but keep in volume).

D-7 (P1) Time-zone handling. Explicitly define the dataset's timezone (assumed America/New_York); store all timestamps in UTC internally, render in local time.

D-8 (P1) Address/PII handling. Address-level fields are potentially sensitive. For the product's aggregate views they are **not required**; store with restricted access and aggregate only (see §18). Latitude/longitude are required for geospatial views.

D-9 (P1) Change-of-definition detection. If complaint-type taxonomy or agency codes change over time, document and expose the change so trends across the boundary are not falsely attributed (see §18, risk R4).

---

9. Metrics & Definitions (the analytical core)

9.1 The three-way distinction

The product's central analytical problem: **a rising complaint count does not mean conditions worsened.** The PRD defines three mutually-exclusive explanations, and every "worsening" claim must be classified into one of them.

Explanation	Definition	Detection logic
E1 — Reporting/ activity increase	More complaints because more people, more activity, or more willingness to report — not because the underlying condition worsened.	Normalized rate (per capita / per household / per area) is **flat or falling** while raw volume rises; or volume tracks a known reporting-channel change (e.g., a jump in `open_data_channel_type=Mobile`). Labeled **"reporting increase — not evidence of worsening."**
E2 — Genuinely worsening conditions	The underlying problem is increasing in frequency/ severity.	Normalized rate **rises** sustainedly (statistically meaningful trend, see §9.3), not explained by seasonality or a reporting-channel change. Labeled **"genuine worsening."**
E3 — Worsening response performance	The agency is taking longer to respond/resolve (independent of complaint volume).	Response-time metrics (median/p95 close time) **rise** and/or backlog (open complaints aging) grows, controlling for complaint mix. Labeled **"agency response performance worsening."**

A single trend can exhibit more than one: e.g., volume up (E1), per-capita flat, *and* response time up (E3) → the product reports both, separately and honestly.

9.2 Normalization methodology

- **Default rate denominator:** complaints per **100,000 residents** per month, using the most recent ACS (American Community Survey) population estimate per borough/community-district. Document the ACS vintage and that it is an annual estimate (so monthly rates use a fixed population denominator for the year).
- **Alternate denominators (selectable):** per **household**; per **square kilometer** (for geographically-concentrated types like graffiti or street condition); per **dwelling unit** (for housing-related types like heat or plumbing — the strongest control for housing complaints).
- **Rule:** raw volume is always shown *alongside* the normalized rate; the UI never presents raw counts as the sole evidence for "worsening."

- **Reporting-channel control:** if a trend coincides with a meaningful change in the share of `open_data_channel_type`, surface a **"reporting-channel change detected"** flag and offer a channel-adjusted view (normalize within channel share).

9.3 Trend & change detection

- **Comparison window (default):** trailing **12 months vs. prior 12 months** (year-over-year), which absorbs seasonality. Also offer trailing-3-month for freshness.
- **Trend test:** for sustained trend claims, use a **Mann-Kendall test** on the monthly normalized-rate series with a significance threshold ($\alpha = 0.05$) and a minimum change magnitude (e.g., ≥10% change) to avoid flagging noise. Linear-regression slope on log-rate may be used as a secondary measure.
- **Statistical caveats surfaced:** small counts (denominator-based flag when monthly count < threshold, e.g., < 30) are marked **"low confidence — small sample."** Multiple-comparison awareness: when scanning hundreds of type×district cells, some "significant" results are false positives; the product states this and can apply a Bonferroni/FDR note on the highest-volume automated lists.

9.4 Response-time metrics (agencies)

- **Median close time** (robust central tendency) and **p95 close time** (tail) in hours/days, per agency × complaint-type × time.
- **Backlog:** count and median age of currently-open complaints, per agency.
- **Comparison:** each agency's metric vs. the same period a year prior, and vs. the city-wide baseline for comparable complaint types.
- **Mix control:** agency response time can change because its *complaint mix* changed (e.g., more complex cases), not because performance changed. Provide a **"mix-adjusted" view** that compares like-for-like complaint types across periods, and label unadjusted trends with this caveat.

9.5 Severity/urgency

- The dataset does not directly encode severity. **Assumption:** complaint type is used as a proxy for severity/urgency (e.g., Heat/Hot Water is treated as high-urgency in cold months). The product labels this as a proxy, not ground truth.

---

10. Dashboard & Visualization Requirements

V-1 (P0) KPI headline. Top-level cards: total complaints (period), trending vs. prior period, most-worsening type (normalized), worst agency response time (median & p95), backlog count.

V-2 (P0) Trend charts. Time-series (monthly) of volume, normalized rate, and response-time percentiles; always with a year-over-year comparison overlay.

V-3 (P0) "Worsening/improving" board. Ranked lists of complaint types and agencies by trend direction and confidence, with color-coded status (worsening / improving / flat / insufficient data).

V-4 (P1) Choropleth map. Community-district-level choropleth of complaint intensity and response time, with the 5 boroughs for higher-level view.

V-5 (P1) Composition charts. Stacked/100% bars for complaint-type share over time; agency response-time distribution.

V-6 (P0) Chart design rules. Data-ink-dense, no distracting 3D/gradients; every chart has a title, axis labels, unit, and a source/timestamp note. Time-series use a clear reference line for the year-prior comparison.

Colorblind-safe palette; not red/green as the *only* signal (use shape/position as well). Tooltips expose the underlying numbers and the metric definition.

V-7 (P0) Empty/low-data states. Clear "insufficient data" states rather than misleading zeros or gaps.

---

11. Automated Insight-Generation Requirements

The product auto-generates a ranked feed of insights. **Every insight must carry: a title, the supporting evidence (metric values), the trend classification (E1/E2/E3), the confidence level, and a plain-language "why this is (or isn't) real" explanation.**

I-1 (P0) Worsening/improving insight. e.g., *"Heat/Hot Water complaints in the Bronx rose 18% YoY on a per-resident basis — classified as genuine worsening (E2)."*

I-2 (P0) Agency response insight. e.g., *"NYPD median response time rose 12% YoY and backlog grew 30% — response performance worsening (E3)."*

I-3 (P0) Reporting-artifact insight. e.g., *"Street Condition complaints rose 25% YoY, but 80% of the increase is a rise in Mobile-app reporting — classified as reporting increase (E1), not evidence of worsening."*

I-4 (P0) Anomaly insight. Spikes (z-score / IQR-based) beyond a rolling baseline, with the event window.

I-5 (P1) Emerging-type insight. Complaint types growing from a low base (new/rare types with rising normalized rate).

I-6 (P1) Geographic-cluster insight. Districts where a type is significantly above the city rate, or where a type is worsening while the city is flat.

I-7 (P0) Insight ranking & filtering. Insights ranked by magnitude × confidence; filterable by type, agency, geography; suppressable (user can dismiss a false insight, which feeds a feedback signal).

I-8 (P0) Deduplication. Merge overlapping insights (a spike and a YoY rise in the same cell) rather than double-reporting.

---

12. Filtering & Drill-Down Requirements

F-1 (P0) Filter dimensions: time range (with presets: last 30/90 days, trailing 12 mo, YoY, custom), complaint type (multi-select, incl. a "top N" quick-pick), agency, borough, community district, reporting channel, status.

F-2 (P0) Cross-filtering. Selecting a value in any chart filters all other charts (linked views); user can see the active filter set and clear/reset.

F-3 (P0) Click-to-drill. Clicking a bar/point filters to that value and navigates to a detail view (type → descriptor → geography).

F-4 (P1) Saved filters & deep links. URL-encoded filter state so views are shareable and bookmarked.

F-5 (P0) URL-parameter support. All key filter state is reflected in the URL (server- or client-side) so drill-downs and links are stable.

---

13. Performance Expectations (large CSV)

P-1 (P0) Initial load. Full-history CSV ingestion completes in a **documented, bounded window** (assumed ≤ 2–4 hours on the reference architecture), with progress reporting and resumability.

P-2 (P0) Query latency. Interactive dashboard queries return in **< 3 seconds (p95)** for the default 12-month filtered views; pre-aggregated (rollup) tables are the primary path. Raw-row scans are reserved for explicit "deep research" exports, not dashboard queries.

P-3 (P0) Frontend. Charts render without jank on a mid-range laptop; no more than ~1MB payload per dashboard view (aggregates, not raw rows).

P-4 (P1) Concurrency. Reference architecture supports ~50 concurrent dashboard users without degrading p95 latency (multi-user not the core use case; see §17).

P-5 (P0) Incremental refresh. Daily incremental load adds new/updated rows in a bounded window (assumed < 30 min), with a status indicator and last-refresh timestamp surfaced in the UI.

---

14. Error Handling & Data-Quality Behavior

Q-1 (P0) Ingestion failures. Fail loudly: pipeline reports the failed stage and row counts; does not silently serve stale data. UI shows data-currency ("last updated") always.

Q-2 (P0) Row-level quality. Invalid rows are **quarantined** (counted, logged, excluded) — never silently dropped. A "data quality" panel reports quarantine counts by reason (dup, bad timestamp, bad geography).

Q-3 (P0) Missing geography. Rows without usable geography fall into an explicit "unspecified" bucket, visible and filterable — never merged into a wrong district.

Q-4 (P0) Missing close time. Open complaints (no `closed_date`) are shown in the backlog metric and excluded from close-time metrics with an explicit note (not treated as zero).

Q-5 (P0) Small-sample guarding. Cells with counts below threshold are suppressed or flagged "insufficient data," never shown as meaningful zeros.

Q-6 (P1) Source-format drift. If NYC Open Data changes the CSV schema (new/renamed columns), the ingestion pipeline detects the mismatch and fails with a clear message rather than mis-mapping columns.

---

15. Accessibility & Responsive Design

A-1 (P0) WCAG 2.1 **AA** target: keyboard navigable; focus states visible; all interactive charts have a text/table equivalent; screen-reader-accessible labels on charts and data tables.

A-2 (P0) Colorblind-safe palette; no color-only encoding of the worsening/improving status (use position, icons, and labels in addition to color).

A-3 (P0) Responsive: usable on desktop, tablet, and mobile browsers. The primary experience is desktop; mobile supports read-only browsing and the KPI/trend views.

A-4 (P0) Legible contrast and scalable text; touch targets ≥ 44px on touch devices.

A-5 (P1) Print/PDF-friendly styling for report export.

---

16. Export / Reporting Requirements

E-1 (P0) CSV export of the currently-filtered dataset (aggregated or raw, user-selectable), with a row/column cap for raw exports and a queue/notify for large pulls.

E-2 (P0) Report export (PDF or image) of the current dashboard/filtered view, including the filter set, metric definitions, and a data-currency note so the report is self-explanatory.

E-3 (P1) Scheduled/email report for a saved filter (weekly digest).

E-4 (P0) Export provenance. Every export includes the data range, source file/version, and ingestion timestamp, so recipients can reproduce or verify.

---

17. Non-Goals

- **Not a city-agency ticketing/CRM system.** This product analyzes historical/trend data; it does not file, route, or resolve 311 complaints.
- **Not a real-time alerting system.** Insights are batch/periodic (daily refresh), not real-time streaming.
- **Not a predictive/forecasting engine** (beyond simple trend tests). Forecasting future complaints is out of scope for v1.
- **Not a general NYC analytics platform.** Scope is the 311 service-request domain.
- **Not multi-tenant SaaS** (initially). Single-tenant web app; multi-user concurrency is modest (see §13).
- **Not an address-level drill-down for the public.** PII/address-level views are restricted (see §18).
- **No machine-learning-based severity classification** in v1; complaint type is the severity proxy.

---

18. Risks & Ambiguities

#	Risk / ambiguity	Impact	Mitigation
R1	**Dataset size** (tens of millions of rows)	Slow ingest/query if architecture wrong	Pre-aggregated rollups; partitioned store; documented batch window (§13)
R2	**Reporting-bias confounds volume trends** (app adoption, awareness campaigns)	False "worsening" claims	Channel-normalization + explicit E1 classification (§9.2, §11-I3)
R3	**Response-time interpretation** (mix changes, complexity changes)	Misattributed agency performance	Mix-adjusted view; percentiles + backlog, not just mean (§9.4)
R4	**Taxonomy/definition drift over time** (complaint types, agency renames)	False trend breaks	Change-of-definition detection + documentation (§8-D9)
R5	**Population denominators are annual & lagged** (ACS)	Per-capita rates are approximate	Document ACS vintage; treat as directional
R6	**PII / address data exposure**	Privacy risk	Address-level access restricted; aggregate-only public views; lat/long rounded to district granularity for public use (§8-D8)
R7	**Statistical false positives** across many cells	Over-reporting anomalies	Multiple-comparison note; confidence labels; manual suppression feedback (§11-I7, §9.3)
R8	**"Worsening" is value-laden / politically sensitive**	Misuse or over-claiming	Strict labeling (E1/E2/E3), confidence, and evidence path on every claim; no bare numbers
R9	**Open-data source reliability / availability**	Ingestion gaps	Idempotent pipeline; data-currency surfaced; quarantine and fail-loud behavior (§14)
R10	**Ambiguity of "resolution quality"**	Can't measure whether a complaint was *resolved well*	Explicitly out of scope; the product measures volume, geography, and response *time*, not outcome quality

Documented assumptions: timezone is America/New_York; ACS population estimates are the per-capita denominator; complaint type is the severity proxy; the raw export is several GB and refreshed daily; the product is a single-tenant web app targeting a modest set of professional users.

---

19. Acceptance Criteria

The release is accepted when all **P0** items pass:

AC-1 — Ingest. The pipeline ingests the full NYC 311 CSV into the queryable store with a documented, resumable batch window and per-reason quarantine counts; a daily incremental refresh runs successfully.

AC-2 — Core answers. A user can answer, in ≤5 minutes: (a) which complaint types are worsening (normalized) in a selected borough; (b) whether that worsening is E1 (reporting) or E2 (genuine) per the classification logic; (c) which agencies' response time is worsening (E3) with median/p95 and backlog.

AC-3 — Classification correctness. On a seeded test dataset with known ground-truth trends (injected reporting-channel jumps, genuine rate rises, response-time rises), the E1/E2/E3 classifier labels ≥90% of cells correctly.

AC-4 — Confidence & evidence. 100% of automated insights display a confidence level and an "why this is real vs. artifact" explanation; every chart shows its metric definition and data range.

AC-5 — Performance. Dashboard queries return < 3 s (p95) for default filtered views on the reference dataset; ingestion completes within the documented window; frontend payloads ≤ ~1MB per dashboard view.

AC-6 — Data quality. Invalid rows are quarantined with visible counts, never silently dropped; missing geography appears in an explicit "unspecified" bucket; open complaints are handled correctly (backlog vs. close-time separation).

AC-7 — Accessibility. P0 accessibility criteria (§15) pass automated and manual review; responsive views render correctly on desktop and mobile browsers.

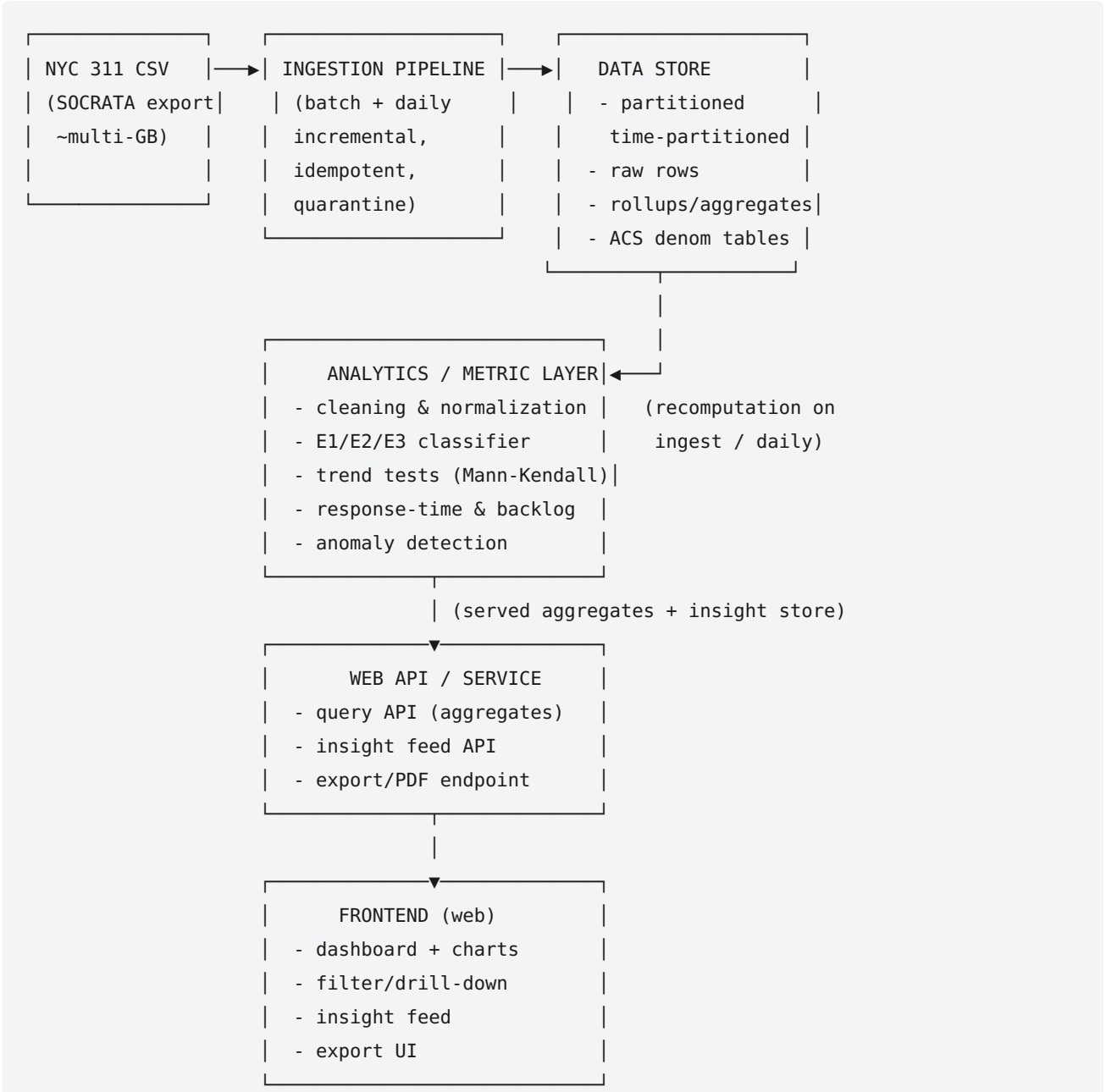
AC-8 — Export provenance. Every CSV/report export includes data range, source version, and ingestion timestamp.

AC-9 — No-PII leak. Public aggregate views expose no address-level data; address-level fields are access-restricted.

---

20. Suggested Technical Architecture

Reference architecture (not prescriptive — engineering may adjust):



Component notes:

- **Storage:** a database with good time-range and aggregate performance and the ability to hold tens of millions of rows (e.g., a columnar or partitioned store; the exact engine is an engineering decision — Postgres with partitioning, ClickHouse, or DuckDB-in-service are all viable options to evaluate). Pre-aggregated rollup tables by (time-grain × type × agency × borough × district) are the primary query path.
- **Ingestion:** batch pipeline with progress/resume; quarantine table; schema-drift detection; daily incremental.
- **Analytics layer:** deterministic, testable functions for normalization, E1/E2/E3 classification, trend tests, and anomaly detection — run on ingest/daily and materialized to an insights table. **The classifier and metric logic must be unit-testable and versioned** (AC-3 requires seeded ground-truth tests).
- **API + frontend:** standard web stack; charts from aggregated API responses (not raw rows); URL-encoded filter state for shareable views.
- **Export:** server-side generation of CSV and PDF reports.

---

21. Appendix — Metric Quick Reference

Metric	Definition	Used for
Complaint volume	Raw count in period	Headline; always with rate
Normalized rate	Complaints / 100k residents (or /household, /km²)	E1-vs-E2 discrimination
Response time	`closed_date − created_date` (hours); median & p95	E3 agency performance
Resolution time	`closed_date − created_date` for resolved records	E3 (secondary)
Backlog	Count + median age of open complaints	E3 (operational load)
YoY change	Trailing 12mo vs prior 12mo (or trailing 3mo)	Trend direction
Trend significance	Mann-Kendall test (α=0.05) + min-magnitude threshold	Confidence labeling
Anomaly	z-score / IQR beyond rolling baseline	Surprise detection

The product's promise, restated: NYC 311 Trends turns a 30-million-row city signal into clear, explained, defensible answers to *what is getting worse, where, and whether the agencies that own the response are improving or deteriorating — without confusing "more people complaining" for "conditions are worse."*