PRD — NYC 311: What's Getting Worse, Where, and Who's Slowing Down

Product Requirements Document — v1.0

Product name	*Pulse311* (working title)
Status	Draft for engineering review
Author	Product Management & Technical Lead
Date	2026-08-27
Dataset	NYC 311 Service Requests (NYC OpenData), CSV
Core product question	*What kinds of problems are getting worse in NYC, where are they getting worse, and which agencies appear to be improving or deteriorating in response time?*
Build status	PRD only — no implementation in this phase

1. Executive Summary

Pulse311 is a web application that ingests the raw NYC 311 Service Requests CSV and turns it into an explorable, insight-driven view of city service quality over time. It answers three families of questions:

1. **Volume:** What kinds of complaints are increasing or decreasing, in absolute terms and after normalizing for population/activity?
2. **Place:** Where (borough, community district, neighborhood, block) are conditions worsening?
3. **Performance:** Which agencies are getting faster, slower, or staying flat at responding to and resolving complaints?

The product is deliberately built around the *hard analytical problem* at its center: distinguishing **"more people complaining about the same thing"** from **"the thing is genuinely getting worse"** from **"the city's response is getting slower."** The PRD defines explicit, defensible metrics and statistical rules for making that distinction, rather than leaving it to visual intuition.

The target users are civic analysts, journalists, advocacy organizations, and city operations staff. The product emphasizes trustworthy trend signals, transparent methodology, drill-down from city → borough → community district → block, and automated surfacing of unusual patterns — with every claim traceable to a reproducible query.

2. Product Objective

Objective: Build a self-serve analytics application that lets users reliably identify — with statistical and methodological rigor — which NYC 311 complaint categories are worsening, where they are worsening geographically, and which agencies' response performance is improving or deteriorating over time.

Success looks like: a user can, within three interactions, (a) see the top worsening complaint types in a chosen geography and time window, (b) understand *why* the system classified each as worsening (volume vs. rate vs. response time), and (c) export the underlying numbers for their own analysis or reporting.

Non-quantified north star: *Pulse311 is the fastest way for a non-specialist to answer "what's getting worse in my borough?" with numbers they can defend.*

---

3. Target Users

Persona	Description	Primary need	Secondary need
Civic data analyst (nonprofit, community board, city staff)	Comfortable with spreadsheets, not necessarily code	Reliable trend + response-time analysis by geography and agency	Exportable data, reproducible methodology
Journalist / reporter	Needs defensible claims and localizable stories	"What's worsening in [neighborhood]?" with caveats and sources	Downloadable charts/tables, citation-ready numbers
Advocacy org / elected-official staff	Needs evidence for campaigns and constituent services	Borough/community-district worsening rankings, agency performance tracking	Regular digest/report export
Agency operations lead *(secondary)*	Interested in own agency's response-time performance	Response-time trends by complaint type and community district	Comparisons vs. other agencies, backlog health
Curious resident *(tertiary)*	Wants "what's going on near me"	Simple neighborhood-level view	Plain-language explanations

Design stance: optimize for the first three personas. The resident experience is a simplified subset, not a separate product.

---

4. Key User Questions

The product must answer, at minimum:

1. What complaint types increased the most in [geography] over [window] — in raw volume and per capita?
2. What complaint types are *statistically significantly* worsening (not just noisy)?
3. Where (community district / neighborhood / block) is a given complaint type worsening fastest?

4. Which agencies' median response times are improving, deteriorating, or flat over [window]?
5. Which agency/complaint-type combinations have the worst and best response-time trends?
6. How does this month/quarter compare to the same period last year (seasonally adjusted)?
7. What unusual or surprising patterns appeared recently that I should look at (e.g., a spike, a new complaint type, a geographic hot spot)?
8. How much of a volume increase is explained by population/activity change vs. real change in conditions?
9. What's the current backlog — how many complaints are open, and how old?
10. Are there data-quality issues that undermine a particular trend (coverage gaps, missing geography, agency renames)?

---

5. Primary Workflows

W1 — Trend exploration (primary). Land on dashboard → select geography scope (city / borough / community district) → view top worsening complaint types → click one → drill into its trend chart, normalized rate, geographic distribution, and agency response-time trend → save/share/export.

W2 — Agency performance review. Select agency → view response-time trend (median/p75/p90) with statistical classification (improving/deteriorating/flat) → drill by complaint type and community district → compare two agencies side by side → export.

W3 — "What's getting worse near me." Enter address or drop pin → resolve to community district + neighborhood → show localized worsening rankings + recent unusual patterns → shareable summary card.

W4 — Insight digest. View the auto-generated "patterns worth your attention" panel → read each insight (plain language + supporting chart + methodology note) → accept/dismiss → export digest.

W5 — Data-quality triage (power user). Open the data-quality panel for a given filter → see coverage %, missing-geography %, status distribution, known anomalies (agency renames, backfill gaps) → decide whether to trust the trend.

---

6. Functional Requirements

6.1 Ingestion & pipeline

- **FR-1.** Ingest the NYC 311 Service Requests CSV (full historical load) and support incremental updates (weekly by default; configurable).
- **FR-2.** Persist the raw data unchanged (immutable raw layer) plus a cleaned/typed analytics layer.
- **FR-3.** Provide a one-command re-ingest/rebuild pipeline with idempotent behavior (re-running produces identical results).
- **FR-4.** Record ingestion metadata: source URL, file hash, row count, ingestion timestamp, per-column null counts. Expose in UI (data-quality panel) and API.

6.2 Exploration & analytics

- **FR-5.** Global filters: date range (with seasonality-aware presets: “past 12 months”, “this quarter vs last”, “YoY”), geography (city, borough, community district, neighborhood, custom draw), agency (multi-select), complaint type (multi-select, searchable), status.
- **FR-6.** Trend views per complaint type / agency / geography: monthly volume, normalized rate, response-time percentiles, open-backlog counts — with comparison to prior-year same-period.
- **FR-7.** Worsening/improving classification (see §10) shown as a labeled badge with confidence level and the governing metric.
- **FR-8.** Ranking views: “Top worsening complaint types”, “Top improving”, “Agencies by response-time trend”, sortable and filterable.
- **FR-9.** Geographic views: choropleth by community district; point heatmap (aggregated, privacy-safe) at high zoom; drill-down block-level for selected type.
- **FR-10.** Side-by-side comparison mode: two geographies, agencies, or time windows compared on the same charts.
- **FR-11.** Every chart supports hover tooltips with exact values, period, and denominator.

6.3 Automated insights

- **FR-12.** Generate a ranked list of machine-discovered patterns (see §12) on a schedule and on demand, scoped to current filters.
- **FR-13.** Each insight card contains: plain-language claim, supporting chart, the exact query that produced it, methodology note, and a confidence label.
- **FR-14.** Users can dismiss/snooze insights; dismissals are per-user and do not affect other users.

6.4 Export & reporting

- **FR-15.** Export current view as CSV (raw underlying rows or aggregated table), PNG/SVG chart, and a shareable URL snapshot.
- **FR-16.** Generate a “methodology appendix” PDF/HTML page describing every metric and rule used in the current view.
- **FR-17.** Scheduled digest (weekly email or saved report) for registered users: top N worsening items in their saved geographies.

6.5 Administration

- **FR-18.** Admin page for triggering ingestion, viewing pipeline logs, and overriding data-quality flags (e.g., marking an agency rename period).
- **FR-19.** Feature-flag system for new metrics/insight rules (canary rollout).

---

7. Data Requirements

7.1 Source dataset

NYC 311 Service Requests (data.cityofnewyork.us, “311 Service Requests from 2010 to Present”), provided as CSV via NYC OpenData (Socrata) download or SODA API.

Documented assumptions: - **A-1.** Full history from 2010-01-01 forward; ~30M+ rows as of 2026, growing ~4–5M rows/year. *(Verify actual row count at ingestion; pipeline must not assume a size.)* - **A-2.** Raw CSV is several GB uncompressed. All processing is designed for this scale (see §15). - **A-3.** Updates arrive with a lag: the most recent 1–2 weeks are incomplete. The system marks the trailing partial period and excludes it from “current” trend calculations by default (configurable).

7.2 Fields used (subset of the full schema)

Field	Type (target)	Use	Notes
`unique_key`	string (PK)	Identity, dedup	May contain non-numeric values in newer exports
`created_date`	timestamp	Time of complaint; trend axis	Parse with timezone (America/New_York)
`closed_date`	timestamp nullable	Resolution time	Null = open/not resolved
`resolution_action_updated_date`	timestamp nullable	Last action; fallback for “responded”	Use as secondary signal
`agency` / `agency_name`	string	Agency analysis	Map code→name; handle renames
`complaint_type`	string	Primary category	~150+ types; normalize case/whitespace
`descriptor`	string	Sub-category; insight signal	High cardinality; free text
`status`	enum	Open/Closed/Pending/Assigned	Define open = not closed
`due_date`	timestamp nullable	SLA reference	Sparsely populated; use cautiously
`borough`	string	Geography level 1	Normalize (BRONX, Bronx, etc.)
`community_board`	string	Geography level 2	Parse number from e.g. “Bronx Community Board 3”
`incident_zip`	string	Fallback geography	Dirty; use only as fallback
`latitude` / `longitude`	float nullable	Point mapping, block-level	Missing for a minority of rows; see §8

Field	Type (target)	Use	Notes
`open_data_channel_type`	string	Channel mix analysis (optional)	Mobile/phone/web share over time

7.3 Derived/analytics schema (precomputed)

- **Complaint-day table:** one row per complaint with cleaned attributes + derived flags (open, within-SLA, response-hours).
- **Monthly aggregates:** by (year-month, borough, community district, agency, complaint type): volume, volume_per_capita, median/p75/p90 response hours, resolution rate, open-backlog count/age, prior-year comparison, trend statistics (slope, p-value, classification).
- **Geography reference:** community district → neighborhood → borough mapping; population estimates per district (see §10.2).
- **Calendar/seasonality reference:** month-of-year seasonal factors per complaint type.

---

8. Data Cleaning & Normalization Requirements

8.1 Cleaning rules (applied in pipeline, documented in methodology appendix)

- **C-1. Dedup:** drop exact duplicate rows on `unique_key`; keep first occurrence. Log count.
- **C-2. Timestamps:** parse ISO/US formats; reject unparseable → null with flag. Convert to America/New_York. Drop rows with no `created_date` (cannot be placed in time) — log.
- **C-3. Borough normalization:** canonical map for 5 boroughs + “Unspecified”/null. Typos corrected via match table; unknown → “Unspecified” (kept, flagged, excluded from geographic analysis by default).
- **C-4. Community board parsing:** extract district number from `community_board` string; invalid → null (flagged).
- **C-5. Agency normalization:** map `agency` codes to canonical names; maintain a rename/merge table (e.g., agency code changes over the dataset) applied as of effective dates.
- **C-6. Complaint type normalization:** trim, collapse whitespace, title-case; maintain a curated alias table for known inconsistencies; unknown values kept as-is with flag.
- **C-7. Coordinates:** validate lat ∈ [40.4, 41.0], lon ∈ [-74.3, -73.6]; invalid/missing → null; if `incident_zip` + `borough` valid, impute community district only (never impute point coordinates). Log imputation.
- **C-8. Test/junk rows:** flag rows matching known junk patterns (e.g., test descriptors, “zzzz”, placeholder addresses); excluded from default views, retained in raw layer.
- **C-9. Response time computation:** `response_hours = (closed_date - created_date)`; if negative or > 10 years, flag as anomaly (likely data-entry error) and exclude from response-time statistics.
- **C-10. Trailing-period handling:** mark the most recent partial week; exclude from “current” trends by default; show in charts with a dashed/hatched style.

8.2 Data-quality reporting

- Compute and display per-filter coverage metrics: % rows with valid geography, % with valid timestamps, % open vs closed, % excluded as anomalies.
- Surface known period anomalies (backfill gaps, agency renames, channel mix shifts) in the data-quality panel so users can make an informed trust decision.
- **Quality gate:** if valid-geography coverage in a filter drops below 80%, show a prominent warning before geographic analysis in that scope.

---

9. Metrics & Definitions

All metrics are defined precisely so an engineer can implement without ambiguity. Time zone: America/New_York. Periods: calendar month primary; calendar quarter secondary.

Metric	Definition	Notes
Volume (V)	Count of complaints with `created_date` in period	Raw
Volume per 1,000 residents (V_N)	V / population × 1,000	Population per §10.2
YoY change	(V_period / V_same-period-prior-year − 1) × 100%	Primary comparison; seasonality-safe
Median response time (MRT)	Median of `response_hours` for complaints closed in period	Use *closed* cohort (resolved), not created cohort — avoid right-censoring bias; see caveat
p75 / p90 response time	Same cohort percentiles	Tail performance
Resolution rate	Closed-in-period / created-in-period (lag-adjusted)	Report with caveat
Backlog	Count of open complaints as of period end	Also: median age of open complaints
% within 30 days	Share of closed complaints with response_hours ≤ 720	Standard “slow vs fast” cut; configurable
Repeat-complaint rate	Share of complaints whose location+type appeared ≥3× in 30 days	Corroborating signal for genuine worsening
Trend slope & significance	Theil-Sen slope of monthly series; Mann-Kendall p-value	Robust to outliers; see §10
Seasonal factor	Mean ratio of month M volume to 12-month centered average, per type	Used for adjustment
Seasonally adjusted volume	V / seasonal factor	For detecting non-seasonal change

Response-time cohort caveat (documented): using *closed* cohorts means the most recent period under-represents slow complaints (they're still open). The system displays both closed-cohort MRT and, for the trailing months, a note when open share > 15%. This is a known, documented limitation — not silently hidden.

---

10. Distinguishing "More Complaints" from "Worsening" from "Slower Response"

This is the analytical heart of the product. Three distinct signals must never be conflated in the UI:

10.1 The three-signal framework

Signal	Question	Primary metric	Corroborating metrics
S1: Activity/scale increase	Are more people/activity generating more complaints about the same underlying state?	V_N (per-capita volume), YoY of V_N	Channel mix, control-complaint comparison
S2: Genuinely worsening conditions	Is the *condition* itself worse, after accounting for scale?	Trend significance of V_N (Mann-Kendall on seasonally adjusted per-capita series)	Repeat-complaint rate, unique-location share, descriptor concentration
S3: Worsening response performance	Is the agency slower to respond/resolve, regardless of volume?	Trend significance of MRT and p90	Backlog growth, % within 30 days, resolution rate

Classification rules (default thresholds, documented and adjustable):

- **Worsening conditions (S2=yes):** Mann-Kendall $p < 0.05$ on seasonally adjusted V_N over the selected window **AND** |Theil-Sen slope| ≥ 5%/year relative to baseline (configurable). Requires ≥ 12 monthly points (fewer → "insufficient data").
- **Volume-driven only (S1=yes, S2=no):** V increased but V_N flat/declining → label *"increase explained by population/activity"*.
- **Worsening response (S3=yes):** Mann-Kendall $p < 0.05$ on MRT **AND** MRT increase ≥ 10% YoY (configurable) **OR** p90 increase ≥ 15% YoY. Requires ≥ 12 closed-cohort monthly points.
- **Noise (neither):** label *"no significant trend"* — important to show, prevents false positives.

10.2 Population/activity normalization (documented assumptions)

- **A-4.** Use NYC Department of City Planning population estimates by community district (latest available year; ~2020 census baseline, interpolated for intercensal years). Document the estimate vintage prominently.
- **A-5.** Where activity (not population) is the driver — e.g., tourist-heavy complaint types, construction-related types — offer optional normalization by proxy series: tourist arrivals (NYC Tourism), active construction permits (DOB), or vehicle volumes where available. **Default: per-capita; activity proxies are an opt-in advanced mode.**

- **A-6.** Control-complaint comparison: when a user examines a specific type, the system automatically compares its V_N trend against a basket of "control" complaint types expected to track general activity (e.g., generic street-maintenance categories), and reports whether the target type is moving *faster* than the control basket. This is the product's most defensible answer to "is it really worse or is everything up?"

10.3 How the UI presents this

- Each complaint type row shows three independent badges: **Volume** (up/flat/down with normalized qualifier), **Conditions** (worsening/improving/flat/insufficient-data), **Response** (slower/faster/flat/insufficient-data).
- Tooltip on each badge explains the metric, thresholds, and p-value.
- A "Why is this flagged?" panel per row expands to show the governing series, the statistical test result, and the control-basket comparison.

---

11. Dashboard & Visualization Requirements

11.1 Dashboard layout (desktop)

1. **Global filter bar** (sticky): date window presets, geography, agency, complaint type, status.
2. **Hero panel:** top 5 worsening complaint types in scope (conditions-classified), with badges and mini sparklines.
3. **Volume panel:** stacked monthly volume by complaint type (top N), with prior-year overlay toggle and seasonally adjusted toggle.
4. **Geography panel:** choropleth of selected metric (default: V_N trend classification) by community district; click-to-drill.
5. **Agency panel:** response-time trend table (MRT/p90, classification, backlog), sortable.
6. **Insights panel:** top auto-generated patterns (FR-12).
7. **Data-quality strip:** coverage indicators; warning banner when gates trip (§8.2).

11.2 Visualization rules

- **Charts:** line/area for time series (with prior-year comparison), bar for rankings, choropleth for geography, box/percentile bands for response-time distributions.
- **Dual-axis discipline:** volume and rate never share an axis silently; normalized views are labeled "per 1,000 residents."
- **Color semantics:** consistent scale (improving = green, worsening = red, flat = gray, insufficient data = hatched) across all views; colorblind-safe palette; never color alone — always pair with shape/label.
- **Uncertainty:** trend classifications show p-value; series with < 12 points show "insufficient data" treatment.
- **Trailing partial period** rendered dashed (C-10).

---

12. Automated Insight-Generation Requirements

The insight engine runs on precomputed aggregates + a nightly scheduled job, plus on-demand for the current filter scope.

Insight classes (v1):

Class	Rule (default)	Example output
I-1 Top worsening	Rank by S2 classification + slope magnitude	“Rodent complaints worsened in Bronx CD 5 (V_N +34% YoY, p=0.003)”
I-2 Emerging spike	Recent 3-month V_N ≥ 2× prior 12-month baseline	“Standing water complaints spiked in Queens in July”
I-3 Response slowdown	S3 classification, largest MRT increases	“DOT response time up 22% YoY for street-light complaints”
I-4 Response improvement	S3 improving, largest decreases	“DOHMH improved p90 response 31% for food-safety complaints”
I-5 Geographic hot spot	Community district whose V_N slope ≥ 2× borough slope	“Noise complaints concentrating in Manhattan CD 3”
I-6 Novel type	Complaint type appearing with meaningful volume that was absent/rare in prior year	“New ‘e-scooter parking’ complaints appearing in Brooklyn”
I-7 Backlog warning	Open share > 15% or median open age rising ≥ 30% YoY	“Backlog aging: open-share at 22% for HPD complaints”
I-8 Data anomaly	Coverage drop, agency rename period, bulk backfill detected	“Coverage warning: 22% of March rows lack coordinates”

Requirements: - Each insight includes: claim, chart, reproducible query (parameters + SQL/query spec), methodology note, confidence label (high/medium/low based on p-value & sample size). - Dedup/aggregate near-identical insights; cap the panel at ~10; user dismissals persist. - Insights are **rule-based in v1** (deterministic, auditable). LLM-assisted narrative is explicitly a v2 option, *never* the source of truth for the claim itself.

---

13. Filtering & Drill-Down Requirements

- **Hierarchy:** city → borough → community district → neighborhood (from district mapping) → block (from coordinates, high zoom only).
- **Drill-down is always additive:** charts, rankings, and insights re-scope to the current filter set; breadcrumbs allow step-back; “clear all” resets.
- **Dimension filters:** date range, geography, agency, complaint type, descriptor (sub-type), status, channel (optional), community board.
- **Cross-filtering:** selecting a bar in a chart filters all other panels (coordinated views).
- **URL state:** every filter combination is encoded in the URL (deep-linkable/shareable); back/forward works.

- **Comparisons:** “compare to prior year” and “compare two geographies/agencies” as first-class toggles (§FR-10).
- **Search:** complaint types and descriptors are searchable with typeahead; geography searchable by name or ZIP.

---

14. Performance Expectations (Large CSV)

Documented assumption A-7: full dataset ~30M rows, several GB raw; monthly deltas ~400k rows.

Operation	Target (p95)	Notes
Full ingest (initial load)	≤ 60 min on a 4-core/16 GB dev box; ≤ 20 min on production spec	Parallel chunked load; Parquet intermediate
Incremental ingest (weekly)	≤ 10 min	Append + rebuild affected monthly aggregates only
Dashboard initial load (city scope, 12 months)	≤ 2 s	Precomputed aggregates; no on-the-fly full scan
Filter change / drill-down	≤ 1 s	Aggregates keyed by (month, geo, agency, type)
Custom date range or cross-filter	≤ 3 s	Falls back to OLAP cube scans (columnar)
Block-level point view (borough scope)	≤ 3 s	Pre-aggregated tiles/grid; no raw-point scatter at city zoom
Insight engine (nightly, full scope)	≤ 30 min	Batched over precomputed cubes
Export of current view	≤ 5 s for aggregated; streaming for raw rows	CSV streaming for large raw exports

Architectural consequence: no user-facing query scans the raw CSV. All dashboards read precomputed cubes; raw rows are served only for explicit export/drill-to-rows, streamed.

---

15. Error Handling & Data-Quality Behavior

- **Ingestion failures:** pipeline is retryable and idempotent; on failure, the app keeps serving the last good dataset and shows a “data as of [timestamp]” banner; admin alerted.
- **Schema drift (new/removed columns, format changes):** ingestion validates schema against a versioned expectation; drift triggers a review flag and, if non-breaking, a best-effort parse with per-column null logging; breaking drift pauses ingest and pages admin.
- **Partial/truncated downloads:** validate row count + file hash against NYC OpenData metadata; mismatch → reject and retry.

- **Empty results:** any filter producing zero rows shows an explicit empty state with suggested relaxations (widen date range, remove filters), never a blank canvas.
- **Missing geography:** rows without valid geography are excluded from maps with a visible count and a "include unspecified" toggle.
- **Open complaints:** never shown as resolved; response-time stats clearly indicate closed-cohort basis; open-share warning per §9.
- **Timeline gaps:** if a month is missing entirely (backfill gap), charts show a gap marker and trend tests use available months with a minimum-count rule.
- **User-facing error states:** all API errors map to friendly messages with a correlation ID; client shows retry for transient failures.

---

16. Accessibility & Responsive Design

- **Accessibility:** WCAG 2.1 AA target. Keyboard-navigable charts with data tables beneath every visualization; aria-labels on all interactive elements; focus states visible; colorblind-safe palette (color + shape/label encoding); text alternatives for insight cards; contrast ≥ 4.5:1.
- **Charts are not the only interface:** every chart has an accessible data table (sortable) as its canonical accessible form.
- **Responsive:** mobile (≥ 360 px) and tablet layouts; on mobile, panels stack, maps become scrollable full-width cards, and heavy tables collapse to key columns with "view all" drill-through. Filter bar becomes a bottom sheet on small screens.
- **Performance on low-end devices:** charts render via lightweight SVG/Canvas with progressive loading; insights lazy-load below the fold.

---

17. Export & Reporting Requirements

- **Chart export:** PNG (2× for print), SVG; includes title, date range, and methodology footnote.
- **Table/data export:** CSV of the current aggregated view; raw-row CSV export (streamed, with column subset selection); both include a metadata header (source, query params, generated timestamp, dataset version).
- **Report export:** printable HTML/PDF "briefing" containing current view + methodology appendix (§FR-16).
- **Scheduled digest:** weekly email (or saved report) with top worsening items per saved geography; opt-in; respects dismissals.
- **Citation support:** every export can include a "methodology & sources" appendix line (dataset link, access date, metric definitions).

---

18. Non-Goals (v1)

- No predictions/forecasts of future complaint volumes (v2 candidate).
- No causal analysis (e.g., attributing changes to policy, staffing, weather) — the product reports signals, not causes. Weather correlation is a possible v2 module, clearly labeled non-causal.
- No complaint *text* NLP beyond descriptor grouping — no topic modeling or sentiment on `resolution_description` (v2).
- No alerting/notifications on thresholds beyond the weekly digest (v2 candidate).
- No multi-tenant city support (NYC-only).
- No mobile native apps (responsive web only).
- No real-time streaming ingestion (weekly batch cadence).
- No agency-facing case management or integration with 311 systems — read-only analytics.
- No user accounts beyond saved views/digests (no social features, no public commenting).

---

19. Risks & Ambiguities

#	Risk / ambiguity	Mitigation / decision
R-1	**Response-time metric bias:** closed-cohort under-represents slow recent complaints	Documented caveat + open-share warning (§9); dual display of open share
R-2	**Population normalization is imperfect** (tourist/commuter activity, demographic shifts)	Control-complaint basket (§10.2) as primary defense; optional activity proxies; explicit labeling
R-3	**Complaint behavior is not condition truth** (reporting propensity changes with channel mix, awareness campaigns)	Channel-mix visibility; control-basket comparisons; insights phrased as “complaint signal,” not ground truth
R-4	**Agency renames/scope changes distort time series**	Versioned rename table with effective dates; data-quality flags at rename boundaries
R-5	**Schema drift from NYC OpenData**	Versioned schema validation + drift review flow (§15)
R-6	**Seasonality can masquerade as trends**	YoY as primary comparison; seasonal adjustment for S2 tests; both available as toggles
R-7	**False positives in insight engine**	p-value thresholds, min-point rules, dedup, confidence labels; rule-based (auditable) in v1
R-8	**“Worsening” is politically loaded** — users may over-claim	UI consistently labels *complaint* signal vs. *condition*; methodology appendix on every export; “insufficient data” treatment
R-9	**Geo imputation drift** (ZIP → district mappings change)	Versioned mapping table; imputation flagged and logged

#	Risk / ambiguity	Mitigation / decision
R-10	**Scale surprises** (dataset larger than assumed)	Parquet + columnar storage; chunked parallel ingest; aggregates keyed for incremental rebuild; load-tested at 2× assumed size
R-11	**Ambiguity: response vs. resolution**	Product tracks both `closed_date` (resolution) and `resolution_action_updated_date` (last action) as separate metrics; UI labels precisely
R-12	**Ambiguity: what "worse" means for different users**	Default rules are explicit and adjustable; advanced settings allow threshold changes with a "custom thresholds" badge

20. Acceptance Criteria

AC-1. Ingestion: Given the production NYC 311 CSV (or a 5M-row synthetic fixture), the pipeline ingests, cleans, and loads aggregates; full ingest completes within the §14 targets; re-running is idempotent and produces identical aggregate counts.

AC-2. Trend correctness: On a labeled synthetic dataset with known injected trends (volume-only increase, genuine per-capita worsening, response-time slowdown), the classification rules (S1/S2/S3) recover the correct labels with ≥ 95% accuracy on $p < 0.05$ cases.

AC-3. Seasonality: A synthetic series with pure seasonality and no trend is classified "no significant trend" (not "worsening") for both volume and response time.

AC-4. Drill-down: From city scope, a user can reach block-level data for any complaint type in ≤ 4 clicks, with breadcrumbs and working back/forward URL state.

AC-5. Insight auditability: Every generated insight includes a reproducible query; an engineer can re-execute the query and obtain the same numbers shown in the card.

AC-6. Performance: Dashboard p95 loads meet §14 targets on production spec hardware with the full dataset loaded.

AC-7. Data-quality visibility: The data-quality panel correctly reports coverage, missing-geography %, open-share, and known anomaly periods for any filter; the < 80% geography-coverage warning triggers correctly.

AC-8. Accessibility: Automated axe-core scans report zero critical/serious violations on all primary views; all charts have accessible data tables; full keyboard navigation works.

AC-9. Export integrity: Every export carries the metadata header and methodology appendix; CSV values match the displayed aggregates.

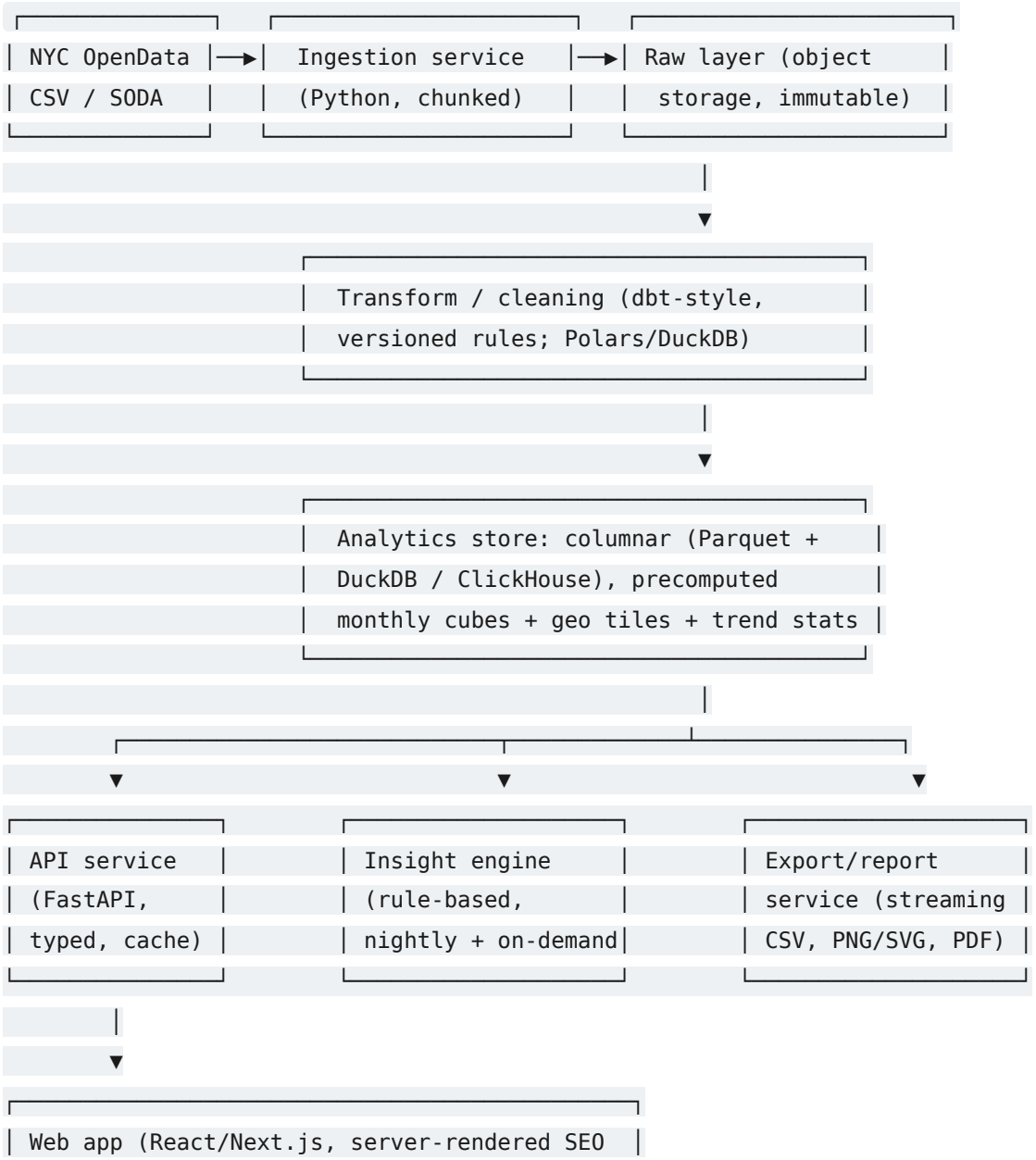
AC-10. Graceful degradation: If ingestion fails or the dataset is stale, the app continues serving the last good data with a clear "as of" banner; empty states and error states render correctly (no blank screens).

21. Suggested Technical Architecture

21.1 Principles

- **Precompute everything user-facing:** dashboards never scan raw rows.
- **Immutable raw, derived analytics:** raw CSV stored unchanged; every transform is reproducible from source.
- **Columnar, compressed storage** for the analytics layer (Parquet + DuckDB or ClickHouse-class engine).
- **Stateless web tier** with all state in URL/query params + lightweight user prefs store.
- **Everything versioned:** schema, rename maps, geo maps, metric definitions, insight rules.

21.2 Components



```
| pages, Vega-Lite/ECharts, MapLibre maps,     |
| accessible components, responsive)           |
└──────────────────────────────────────────────┘
```

21.3 Technology choices (suggested, with rationale)

Layer	Choice	Rationale
Ingest/transform	Python + Polars + DuckDB	Fast columnar processing of multi-GB CSV on modest hardware; simple ops
Analytics store	DuckDB (embedded) for v1; ClickHouse if multi-user load demands	Zero-ops start; columnar; SQL for reproducible queries; migration path documented
Aggregates	Materialized monthly cubes + geo grids + trend-stat precompute	Meets §14 targets
API	FastAPI, typed, cached (Redis optional)	Simple, documented, testable
Frontend	React/Next.js; Vega-Lite + ECharts; MapLibre GL	Server-rendered for shareability; proven charting; open-source maps
Stats	scipy / statsmodels (Mann-Kendall via pymannkendall, Theil-Sen)	Standard, auditable implementations
Ops	Docker Compose (single-node) → optional K8s; S3-compatible storage; scheduled jobs via cron/Airflow-lite	Matches “small team, single dataset” scale
Testing	Fixture datasets with injected trends (AC-2/AC-3); golden-file aggregate tests; property tests on cleaning rules	Correctness is the product

21.4 Deployment

- Single-node MVP (Docker Compose): web + API + worker + storage. Scale path: separate worker, managed columnar DB, CDN for static assets.
- Environment: production + staging; ingestion runs in staging first, promoted to production after validation gates pass.

---

22. Assumptions Log (all documented assumptions, one place)

ID	Assumption	Status
A-1	Dataset: 2010–present, ~30M rows, growing ~4–5M/yr	Verify at first ingest; pipeline size-agnostic
A-2	Raw CSV multi-GB; processing designed for that scale	Design constraint
A-3	Updates lag 1–2 weeks; trailing partial period excluded from current trends	Implemented via C-10
A-4	Population = DCP community-district estimates, latest vintage, documented	Default normalization
A-5	Activity proxies (tourism, permits, traffic) are opt-in advanced mode	Advanced normalization

ID	Assumption	Status
A-6	Control-basket comparison is the primary defense against “everything is up”	Core analytical feature
A-7	Monthly delta ~400k rows; weekly incremental ingest ≤ 10 min	Performance target
A-8	`closed_date` is the resolution signal; `resolution_action_updated_date` is the last-action signal	Dual metric (§R-11)
A-9	Response-time analysis uses closed cohorts; open-share caveat displayed	Documented limitation
A-10	Insight engine is rule-based in v1; LLM narrative is v2 and never the source of truth	Trust & auditability
A-11	Users are U.S.-English; dates displayed as MM/DD/YYYY; numbers in U.S. format	UX default
A-12	Privacy: aggregated views only; block-level only at high zoom; no PII fields exposed (311 data is public; still avoid incidental PII in exports)	Responsible data practice

---

Appendix A — Metric definition quick reference (for engineering)

- `response_hours` = hours between `created_date` and `closed_date` (or `resolution_action_updated_date` for the “last action” variant), America/New_York, anomaly-clamped (0 < x ≤ 87,600).
- `V` = count(created_date in period).
- `V_N` = V / population(district) × 1,000. City-level: V / 8.3M × 1,000 (A-4).
- `MRT` = median(response_hours) over closed-in-period cohort.
- `p90` = 90th percentile of same cohort.
- `open_share` = open complaints / all complaints created in period (evaluated at period end).
- `YoY_pct` = (V_period / V_same_month_prior_year − 1) × 100.
- Trend tests: Mann-Kendall (p) + Theil-Sen slope on seasonally adjusted monthly V_N (S2) and on monthly MRT/p90 (S3).
- Classification thresholds: S2: p < 0.05, slope ≥ 5%/yr, n ≥ 12. S3: p < 0.05, MRT ≥ +10% YoY or p90 ≥ +15% YoY, n ≥ 12. All configurable.

End of PRD. Ready for engineering estimation and architecture review.