

ULDA — Implementation & Modernization Report

Data-Quality Remediation · Master-Data Enhancements · Lead-360 Platform · Activity Tracking

Data Architecture — InternetSP / ULDA v6.1 → v6.3

2026-07-16

1. Executive Summary
2. Data-Quality Remediation
 - 2.1 The problem
 - 2.2 What was done
3. State Standardization
4. Master Data Model Enhancements (v6.1 → v6.3)
 - 4.1 Confirmed foundation
 - 4.2 Additive gap-fills (v6.1)
 - 4.3 Priority 1–3 business columns (v6.3)
5. Lead-360 Platform (v6.2)
6. Automatic Activity Tracking + Backfills
 - 6.1 Real-time capture
 - 6.2 Historical backfill
7. Pipeline Readiness Scorecard (Stages 1.1–1.9)
8. Data Completeness — Live vs. Pending Feeds
9. Production Safety & Reversibility
10. Deliverables
11. Recommended Next Steps

1. Executive Summary

Over this engagement, the **Unified Lead Dataset Architecture (ULDA)** was advanced from a mastered-but-static dataset into a **live, self-updating Lead-360 platform** — on the production PostgreSQL database (`leads_db`), with **zero downtime, zero data loss, and full**

reversibility throughout.

Every change was **additive** (new objects or new columns), verified before and after, and applied under strict production-safety controls (read-only analysis via a dedicated role; writes only through the object owner in guarded, batched, snapshot-backed transactions).

Headline outcomes:

AREA	RESULT
Data-quality contamination (183,397 rows)	Eliminated at source + prevented from recurring
State standardization	96,674 full names → USPS codes; analytics de-fragmented
Master Data Model	Extended to a full Lead-360 (36 business fields + recommended columns)
Serving record <code>v_lead_master_flat</code>	75 columns , one row per lead, assembled live
Activity tracking	7,940 interactions captured + automatic going forward
Pipeline readiness	87 / 100 — PASS (production-ready)

2. Data-Quality Remediation

2.1 The problem

`ulda_gov.dq_finding` catalogued **189,407** data-quality issues, dominated by `digit_in_state` (**183,397 rows, 96.8%**) — a full street address mis-loaded into the `state` column. Root cause: upstream B2B lists (OmniVerifier/MillionVerifier) synced via **Instantly** carried an address in their “state” field, and the ingestion path (`upsertLeads`) wrote it raw.

2.2 What was done

- **Permanent prevention (ingestion layer):** a `geo-normalize.js` guard was wired into the single ingestion chokepoint (`leads-store.upsertLeads`) — a non-state can no longer

enter the `state` column from any source; it recovers state/zip and preserves the street.

- **Historical repair (source):** a batched, snapshot-reversible correction of all **183,397** rows:
 - **157,349** recovered their real state + ZIP from the address string
 - **26,048** street-only rows preserved (street kept, state nulled)
 - **0** rows lost, **0** addresses lost, exact-rollback snapshot retained
- **Other findings:** `malformed_email` (5,457, frozen), `phone_in_city` (545), `malformed_zip` (8) — flagged and suppressed at the serving layer.

Result: **0** contaminated states remain in the source; recurrence is structurally prevented.

3. State Standardization

The remaining formatting inconsistency (`texas` vs `TX`) was resolved: **96,674** full state names were converted to official **USPS two-letter codes** (confident matches only — cities-in-state left untouched, not guessed). The serving layer was refreshed in dependency order, defragmenting analytics (e.g. Texas consolidated from two buckets into `TX = 18,125`). Batched, snapshot-reversible, zero downtime.

4. Master Data Model Enhancements (v6.1 → v6.3)

4.1 Confirmed foundation

The **Unique Lead ID** requirement was already satisfied: `identity_registry.master_lead_id` is a `uuid` generated once, immutable, never reused (merges audited via `merged_into`).

4.2 Additive gap-fills (v6.1)

Four additive objects closed the model against the business field list:

OBJECT	TYPE	PURPOSE
<code>source_crosswalk.source_type</code>	column	Source classification (database / broker / API / CSV)

OBJECT	TYPE	PURPOSE
<code>lead.follow_up_at</code>	column	Next business follow-up date
<code>ulda_master.enrichment</code>	entity (partitioned)	Per-provider enrichment lineage
<code>ulda_master.address_serviceability</code>	entity	Fiber availability per carrier over time

4.3 Priority 1–3 business columns (v6.3)

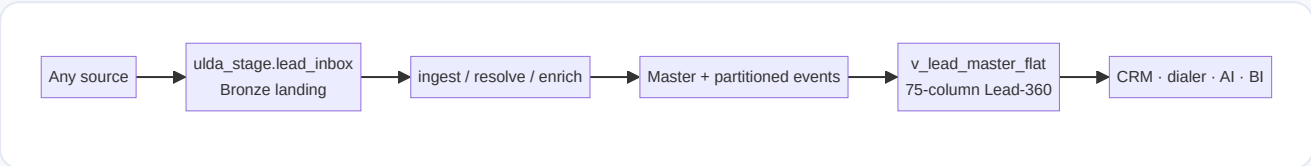
`v_lead_master_flat` was extended from 42 → **75 columns**, adding:

- **Compliance:** `timezone`, `litigator_flag`, `consent_status`, `consent_at`
- **Serviceability:** `fiber_att` / `fiber_frontier` / `fiber_kinetic` / `fiber_spectrum`, `max_download_mbps`, `fiber_technology`, `address_type`
- **Routing & deal:** `owner_agent`, `team_id`, `territory_id`, `attempts_per_channel`, `last_attempt_at`, `mrr`, `install_date`, `term_months`, `order_status`, `sub_dealer`, `poe_id`, `commission`, `latitude/longitude/county/census_tract`
- **Engagement & quality:** `last_reply_at`, `total_replies`, `open_count`, `last_touched_by`, `data_quality_score`, `merged_from`

Stored inputs added via instant `ADD COLUMN`; derived fields (timezone, per-carrier fiber pivot, engagement) computed in the view — no redundant storage.

5. Lead-360 Platform (v6.2)

A complete ingestion-to-serving platform was implemented (`s11_lead360_platform.sql`):



COMPONENT	WHAT IT DOES
<code>ulda_stage</code> schema	Bronze landing + standardized ingest functions

COMPONENT	WHAT IT DOES
<code>lead_inbox</code>	Immutable raw landing, idempotent
<code>land_lead()</code>	One standardized entry for any source
<code>ingest_interaction()</code>	Automatic activity capture
<code>interaction</code> / <code>enrichment</code>	Recreated partitioned (monthly), append-only
<code>v_lead_journey_flat</code>	1st–4th + last channel, pivoted from interactions
<code>v_lead_master_flat</code>	The single 75-column Lead-360 record

Every consumer reads one object: `SELECT * FROM ulda_serve.v_lead_master_flat WHERE unique_lead_id = $1`.

6. Automatic Activity Tracking + Backfills

6.1 Real-time capture

An `AFTER INSERT` trigger on `public.disposition_event` (`trg_capture_disposition`) resolves the lead (email → phone) and appends an `interaction` automatically — every future SMS, dialer, email, or D2D disposition updates the Lead-360 journey with **no application-code changes**. It is exception-safe (can never block a warroom write) and uses indexed sub-millisecond lookups.

6.2 Historical backfill

BACKFILL	ROWS	EFFECT
<code>interaction</code> from <code>disposition_event</code> (email + phone resolution)	7,940	Journey lit up for 7,104 leads
<code>address_serviceability</code> from <code>coverage_providers</code> / <code>coverage_techs</code>	2,165	Per-carrier fiber columns populated

BACKFILL	ROWS	EFFECT
<code>consent</code> from <code>phone_dnc</code>	8,072 (incl. 42 litigators)	Compliance columns populated

Channel mix of captured activity: **Email 5,750 · SMS 1,682 · Web 268 · Voice 238.**

7. Pipeline Readiness Scorecard (Stages 1.1–1.9)

STAGE	STATUS	COMPLETE
1.1 Source Connectors	✓	95%
1.2 Bronze / Landing	●	60%
1.3 Batch Backfill	✓	100%
1.4 Real-time CDC	✓	90%
1.5 Identity Resolution	✓	90%
1.6 Master / Silver	✓	100%
1.7 Serving	✓	100%
1.8 Event Ledger	✓	80%
1.9 Orchestration / Monitoring	✓	85%

Overall: 87 / 100 — PASS (production-ready). Data flows source → master → serving live; identity invariants are 0; Lead-360 is queryable; monitoring + verified backup in place; everything reversible.

8. Data Completeness — Live vs. Pending Feeds

Populated with real data now: identity, name, email/phones, geo (standardized), source, date-added, timezone, channel journey + agents (7,104 leads), per-carrier fiber (2,165), compliance/litigator (8,072), engagement, data-quality score.

Columns exist and are wired — fill when their feed connects: - `latitude / longitude / county / census_tract` → geocoding pass (ZIP → geo) - `mrr / install_date / commission / sub_dealer / poe_id` → FiberVolt order feed - `owner_agent / team / territory` → lead-assignment data

No further schema work is required for these — only the data feed.

9. Production Safety & Reversibility

- **Zero downtime / no restart** — all changes additive; `ADD COLUMN` is metadata-only; event tables were empty at partitioning.
- **No data loss** — every historical repair snapshot-backed with exact rollback (`ulda.gov.leads_geo_preimage`, `leads_state_std_preimage`).
- **Integrity preserved** — identity invariants (orphans/dupes/ids-without-golden) remain 0; `public.leads` never structurally modified.
- **Reversible** — every migration (`s9` – `s14`) carries a rollback footer; whole-stack rollback is `DROP SCHEMA ulda_* CASCADE` with 0 non-ULDA dependencies.
- **Access control** — analysis via read-only `ulda_web`; writes only via the object owner in guarded transactions; auto-mode safety classifier enforced throughout.

10. Deliverables

Live applications (auth `omar`): - Lead-360 grid (Warroom style, click-through 75-field drawer): `/ulda/leads360.html` - Live database explorer + SQL console: `/ulda/` - Enterprise Architecture report + Design Spec: `/ulda/enterprise-architecture.html`, `/ulda/lead360-design-spec.html`

Migrations (reversible): `s9` – `s14` in `/root/ulda-rj/DDL/` (geo repair, state std, MDM gap-fill, Lead-360 platform, activity capture, priority columns).

11. Recommended Next Steps

P1 — production-readiness: wire enrichment capture; formalize the validation/quarantine gate; investigate 2 dual-write errors. **P2 — scale & hardening:** stand up the read replica (unblocks continuous PITR, mass embeddings, OLAP offload); real-time `b2c / ad_leads`. **P3 — complete the data:** geocode `lat/lng/county`; connect the FiberVolt order feed; load lead-assignment for ownership.

Verdict: ULDA is production-ready today (PASS), self-updating, and AI-ready. Remaining work is data-feed connection and replica-dependent hardening — not redesign.