VerbaFlo Intelligence Copilot - Final Product Requirements Document (PRD)

0) Vision

A single, trustworthy intelligence layer for the business. A unified LLM-orchestrated brain that can answer any reasonable business question and proactively surface what matters across VerbaFlo's structured and unstructured data. It reasons over the canonical metrics, knows their definitions, runs analyses, renders visualisations, and explains *why* changes happen, with full provenance and governance.

Outcomes

- **Answerability**: >90% of ad-hoc questions answered accurately with traceable SQL and cited sources.
- **Intelligence**: Automatic surfacing of anomalies, emerging trends, drivers, and forecasts.
- **Trust**: Every answer includes data lineage (views/tables), executed SQL, filters, and data staleness.
- **Speed**: P50 < 5s for cached/canonical questions; P95 < 12s for novel complex queries.

1) Objectives, Non-Goals, Success Metrics

Objectives

1. Provide a natural-language interface to verified business metrics and dimensions.
2. Enable diagnosis (root-cause, contribution) and forward-look (forecast) analyses.
3. Support unstructured evidence (calls, WhatsApp, email, chat) linked to metrics.
4. Guarantee reproducibility via provenance, SQL transparency, and stable semantic layer.

Non-Goals (initially)

- Write operations back to source systems; bulk record edits; PII export without approvals.
- Fully automated decisioning without human accountability.

Success Metrics

- Plan Success %, Exec Success %, First-Answer Correctness, Insight Utility rating, % sessions resolved without escalation, p50/p95 latency, Alert precision/recall, % queries with downloadable artefacts.

2) Users & Jobs-To-Be-Done

Personas

- **Executive**: Portfolio health, weekly deltas, risks and opportunities.
- **City/Property Manager**: Lead mix and quality, loss reasons, SLA breaches, occupancy drivers.
- **Sales Ops**: Funnel throughput, qualification time, channel effectiveness.
- **Support/QA**: Outside-working-hours load, channel mix, transcript themes and sentiment.
- **Data Analyst**: Verified SQL generation, metric catalogue stewardship, shareable permalinks.

JTBD (examples)

- “Calls, WhatsApp and chatbot sessions this week vs last; inside vs outside hours.”
- “Lead distribution by city and property; 4-week trend and top loss reasons.”
- “Average lead-qualification time by channel and tenant type.”
- “Keywords and topics associated with ‘Property not listed’ losses last month.”

3) Scope & Phasing (Delivery still 1-month plan; capability roadmap below)

Phase 1 (Launch)

- LLM Orchestrator + SQL Tooling on allow-listed Canonical Views.
- Semantic Layer (metrics YAML + entities + grain + filters + joins); glossary.
- Answer Card (KPI + visual + table) with **Show SQL & Sources**, **Assumptions & Filters**, **Staleness**.
- Clarifying question engine; safe execution; semantic/result caching.
- Unstructured foundation: ingestion + indexing (STT/OCR + PII redaction + vector search) with cross-modal Q&A and linkage to leads.

Phase 2

- Insight tools: anomaly detection, contribution/driver analysis, cohorts, seasonality/forecast.
- Auto-Notebook (reproducible Python steps); scheduled digests/alerts.
- Conversational quality metrics (talk/listen ratio, hold time, interruption rate).

Phase 3

- Decision support (counterfactual “what-if”), multi-source joins, action hooks (create Jira/Slack/HubSpot tasks), proactive watchlists.

4) Data Sources & Ingestion

Structured (initial)

- `analytics.leads_daily_v`, `analytics.funnel_v`, `ops.occupancy_daily_v`, `support.contacts_v`, `support.sessions_v`, `web.analytics_v`.

Unstructured (initial)

- Call recordings (audio → STT), WhatsApp messages/media, email (EML/HTML), chat transcripts.

Connectors

- HubSpot/CRM, Telephony (Exotel/Twilio), WhatsApp BSP, Gmail/IMAP, Web chat, Google Sheets (CSV drops).

Processing Pipeline

1. **Ingest** (stream/ETL)
2. **Normalise** (UTC, org/property IDs, channel enums)
3. **PII Redaction** (names, emails, phones)
4. **STT/OCR** (multi-accent; diarisation)
5. **Chunking** (semantic windows + overlap)
6. **Embedding** (vector representations)
7. **Indexing** (vector DB + metadata store)
8. **Entity Linkage** (lead_id/session_id)
9. **Quality Checks** (coverage, WER, redaction rate, language detection)

Storage

- Warehouse tables (columnar), Parquet for logs, Vector DB (e.g., pgvector/FAISS), Object store for media, with lifecycle rules.

5) Semantic Layer (Truth Canon)

A governed catalogue of metric and dimension definitions used by the Orchestrator and exposed to analysts.

Metrics YAML (illustrative)

```
metrics:
  calls_received:
    grain: [day, org_id]
    definition: "count(*) from support.contacts_v where channel = 'call'"
    filters: ["created_at between {{start}} and {{end}}"]
    pii: false
  emails_received:
    grain: [day, org_id]
    definition: "count(*) from support.contacts_v where channel = 'email'"
  whatsapp_sessions:
```

```
    grain: [day, org_id]
    definition: "count(*) from support.sessions_v where channel = 'whatsapp'"
  chatbot_sessions:
    grain: [day, org_id]
    definition: "count(*) from support.sessions_v where is_bot = true"
  avg_call_duration_sec:
    grain: [day, org_id]
    definition: "avg(duration_sec) from support.contacts_v where channel = 'call'"
  chats_per_visitors_pct:
    grain: [day, org_id]
    definition: "(chats::float / visitors) * 100 from web.analytics_v"
  outside_hours_count:
    grain: [day, org_id, channel]
    definition: "count(*) from support.contacts_v where is_outside_hours = true"
  outside_hours_ratio_pct:
    grain: [day, org_id, channel]
    definition: "(outside_hours_count::float / total_contacts) * 100"
  enquiries:
    grain: [day, org_id, city]
    definition: "count(*) from analytics.leads_daily_v where stage = 'Enquiry'"
  engaged:
    grain: [day, org_id, city]
    definition: "count(*) from analytics.leads_daily_v where stage = 'Engaged'"
  contacted:
    grain: [day, org_id, city]
    definition: "count(*) from analytics.leads_daily_v where stage = 'Contacted'"
  qualified:
    grain: [day, org_id, property_id]
    definition: "count(*) from analytics.leads_daily_v where stage = 'Qualified'"
  lost:
    grain: [day, org_id, property_id]
    definition: "count(*) from analytics.leads_daily_v where stage = 'Lost'"
  lead_qualification_time_min:
    grain: [day, org_id]
    definition: "avg(extract(epoch from (qualified_at - first_contact_at))/60.0)"
  tenant_type_distribution_pct:
    grain: [day, org_id, tenant_type]
    definition: "100 * count(*) / sum(count(*)) over (partition by day, org_id) from
analytics.leads_daily_v"
  lead_distribution:
    grain: [day, org_id, property_id, city, source_country]
    definition: "count(*) from analytics.leads_daily_v"
  lost_reason_distribution_pct:
    grain: [day, org_id, lost_reason]
    definition: "100 * count(*) / sum(count(*)) over (partition by day, org_id) from
analytics.leads_daily_v"
```

Entities: org, city, property, channel, source_country, tenant_type, room_type.
Joins:

- `analytics.leads_daily_v.property_id → ops.properties.id`
- `support.contacts_v.lead_id → analytics.leads.id`

Glossary

- **Outside Working Hours**: `contact_time` not in `[org.work_start, org.work_end]` in the org's timezone (default **Asia/Kolkata** unless tenant overrides).
- **Session**: Contiguous messages with ≤30 minutes inactivity.

6) Product UX

Ask: One input with context (tenant, date range, grain).
Answer Card

- Topline: one-sentence insight.
- Visual: auto-selected chart (time series, bar/stacked, funnel, cohort heatmap, Pareto, KPI tiles, maps).
- Table: preview (LIMIT 50), sortable, filterable.
- Collapsible: **SQL & Sources**, **Assumptions & Filters**, **Data Staleness**.
- Chips: **Why?**, **Forecast**, **Drill-downs** (suggested dimensions: city, property, channel, tenant type, room type, source country).
- Actions: **Create Alert**, **Export** (CSV/XLSX/PNG/PDF), **Share Permalink**.

Exploration

- Guided filters and saved views; session context preserved across follow-ups.
- Drill-through to redacted transcripts from metric spikes.

Accessibility & Internationalisation

- UK English defaults; date/number localisation; keyboard navigation.

7) Intelligence (Analytical Tools)

- **Anomaly Detection**: rolling z-score/ESD; highlight spikes/dips with confidence bands; alert thresholds.
- **Contribution/Driver Analysis**: greedy Shapley-style attribution across dimensions; shows positive/negative contributions to deltas.
- **Cohorts**: conversion/retention by signup month/property/tenant type.
- **Forecasts**: seasonal naïve + ETS/ARIMA with 80/95% intervals; report MAE/MAPE.
- **Counterfactuals (later)**: simulate impact of changing a driver using historical elasticities; present caveats.

8) Unstructured Analytics

- **Cross-modal search** over calls, WhatsApp, email, chat.
- **Topic modelling** (e.g., BERTopic/LDA) for emerging themes; keyword trending.
- **Entity extraction** (properties, universities, budgets, dates) and timelines.
- **Quality metrics**: sentiment, talk/listen ratio, interruption rate, hold time.
- **Traceability**: snippet previews (redacted), link to source artefacts (audio/message) with audit.

Privacy

- PII detection/redaction at ingest; role-gated de-redaction with approvals and immutable audit log.

9) Architecture

Gateway/API (FastAPI): Auth (JWT→RBAC), rate limit, tenancy context.
Orchestrator (LLM): Router → Planner → Critic → Tool calls (SQL/Chart/Stats/Catalog/Notebook).
Data Plane: Read replicas for OLTP freshness; materialised canonical views; vector DB for unstructured; object store for media.
Caching: Semantic cache (question+context→SQL) and result cache (SQL+bindings→table JSON) with TTL, invalidation on fresh loads.
Observability: OpenTelemetry traces; structured logs; lineage graph.

10) Tool Specifications (Function-calling)

- `fetch_catalog({metric?, tables?, columns?}) → {metrics_yaml, ddl_snippets, samples}`
- `sql_query({dialect:"postgres", sql:string, params?:object}) → {rows, rowcount, runtime_ms}`
- `stats_analyse({kind:"anomaly"|"contribution"|"cohort"|"forecast", input:{…}}) → {insights}`
- `chart({intent:{x,y,series,mark}, data_preview}) → {vega_spec}`
- `notebook_export({plan, steps}) → {signed_url}`

Validator & Guardrails

- **SQL Policy**: `SET TRANSACTION READ ONLY`, statement timeout 5s, forbid DDL/DML, table/view allow-list, enforced date filters on large facts, row limits on detail queries, cartesian join detection.
- **Static Checks**: parse with `sqlglot`; lint with `sqlfluff`.
- **Critic Pass**: the LLM re-reads plan + result to catch obvious mistakes/PII leaks before responding.

11) Security, Privacy & Compliance

- **RBAC & RLS**: per org/role; view-level masking of emails/phones; privileged de-redaction workflow.

- **Secrets**: Vault/SSM; rotation schedule; least-privilege credentials.
- **Audit**: immutable logs of prompts, SQL text, bindings, row counts, runtime, user identity.
- **Data Residency**: region-aware storage; configurable retention (e.g., audio 180 days by default).

12) Performance & SLOs

- p50 < 5s cached; p95 < 12s cold; timeout 15s.
- Default max result rows 5k (role-gated overrides). Streaming for large tables.
- Concurrency limits per tenant; graceful degradation (summary first, details on demand).

13) Product Requirements & Acceptance Criteria

R1: Ask → Answer with provenance
Given a valid metric, *when* asked in natural language, *then* return KPI, chart, table (≤50 rows), SQL, sources, filters, staleness.
AC: Human can re-run SQL and reproduce within ±1% on fixture data.

R2: Visualisations
Auto-select chart type; allow manual override; export PNG/PDF.
AC: “Lead Distribution by City (last 30 days)” renders stacked bar with tenant_type by default.

R3: Unstructured Q&A
Search across calls/WhatsApp/email/chat with snippet previews; link back to artefacts.
AC: Query “guarantor issue mentions last 14d” achieves ≥90% precision on labelled set.

R4: Outside-hours metrics
Compute per channel using org timezone (default Asia/Kolkata; tenant overrides).
AC: Contact at 22:15 IST counts as outside-hours when org hours are 10:00–20:30 IST.

R5: Funnel
Stages Enquiry→Engaged→Contacted→Qualified→Lost with counts and %; by city/property/channel.
AC: Matches ground-truth fixture within ±1 lead per stage for test period.

R6: Lost Reasons
Pareto + 100% stacked bar; click-through to transcripts.
AC: Users can filter by keyword/topic and channel.

R7: Exports & Downloads
CSV/XLSX tables; PNG/PDF charts; JSON via API; bundle (chart+table+SQL+metadata).
AC: From any answer card, Export → format → signed link valid 24h.

R8: Alerts & Digests
Create alerts on metrics or slices; deliver to Slack/Email; daily/weekly schedules.

R9: Permissions
Roles: Analyst (full explore), Manager (summary+exports), Executive (summary only), QA (transcripts with redaction).
AC: Exec cannot view raw PII; QA can request temporary de-redaction; all actions audited.

R10: Reliability
If SQL validation fails or times out, provide fallback summary and suggest a narrower query.

14) Canonical Views (Schemas - sketch)

analytics.leads_daily_v
Keys: date, org_id, lead_id, property_id, city, source_country, stage, tenant_type, room_type, lost_reason, created_at, first_contact_at, qualified_at.

support.contacts_v
Keys: contact_id, org_id, lead_id, channel, direction, created_at, duration_sec, is_outside_hours.

support.sessions_v
Keys: session_id, org_id, lead_id, channel, is_bot, started_at, ended_at.

nlp.docs_v
Keys: doc_id, org_id, lead_id, channel, ts, text, lang, sentiment, entities, topics, pii_masked_text, media_ref.

15) Metric Dictionary (mapped to your KPIs)

Operational

- *Calls Received* → `calls_received`
- *Emails Received* → `emails_received`
- *WhatsApp Sessions* → `whatsapp_sessions`
- *Average Call Duration* → `avg_call_duration_sec` (display in mm:ss)
- *Chats / Visitors* → `chats_per_visitors_pct`
- *Outside Working Hours – Calls/Emails/WhatsApp* → `outside_hours_count`, `outside_hours_ratio_pct` per channel
- *Chatbot Sessions* → `chatbot_sessions`

Funnel

- *Enquiries, Engaged, Contacted, Qualified, Lost* → `enquiries`, `engaged`, `contacted`, `qualified`, `lost` (+ stage rates)

Lead Quality & Speed

- *Avg. Lead Qualification Time* → `lead_qualification_time_min`

- *Tenant Type Distribution* → `tenant_type_distribution_pct` (Student / Working professional / Nil)

Distribution & Segments

- *Lead Distribution by City / Property* → `lead_distribution` sliced by `city` / `property_id`
- *Lead Generation Trend* → `lead_distribution` over rolling 4-day buckets (configurable)
- *Enquiries by Room Type* → `lead_distribution` filtered `stage='Enquiry'` grouped by `room_type`
- *Lead Distribution by Source Country* → `lead_distribution` grouped by `source_country`

Performance & Outcomes

- *Qualified Leads by Property* → `qualified` grouped by `property_id`
- *Qualified & Lost Totals* → `qualified`, `lost`
- *Lost Reason Analysis* → `lost_reason_distribution_pct` with topic/keyword overlays from unstructured data

Search & Keywords (Unstructured)

- Top keyword lists and trends derived from `nlp.docs_v.topics/keywords` linked to leads and outcomes.

16) Example End-to-End (diagnostic)

Question: "Paid/Qualified leads last 4 weeks by city; what drove the week-over-week drop?"
Plan: Use `qualified` metric from `analytics.leads_daily_v` → roll up weekly by city → compare last week vs prior → run `stats_analyse(kind="contribution")`.
Answer (sketch): "Qualified leads fell 12% WoW; biggest negatives: Mumbai (-6.2%) and Organic (-3.1%). Primary driver: lower weekend conversion post price update on 2 Aug. See SQL & Sources."
UX: Line chart + stacked bar; chips for *Drill into Mumbai*, *Show channel split*.

17) Engineering Blueprint

Services: gateway, orchestrator, sql-runner, stats, render, catalog, ingestion workers.
Languages/Libraries: Python (FastAPI, Pydantic, SQLAlchemy/asyncpg, sqlglot, sqlfluff, statsmodels, ruptures), Redis (cache), Postgres/DuckDB.
Infra: K8s with HPA on p95 latency; read-replica for OLTP; object store for query bundles; Vault/SSM for secrets.
Observability: OpenTelemetry, Prometheus/Grafana dashboards (latency, cache hit-rate, plan/execution success).

18) API Surface (externalising the brain)

- `POST /ask` → `{question, context?} → {summary, chartSpec (Vega-Lite), table, sql, provenance, followUps}`
- `POST /alerts` → create/update watches (metric, slice, threshold)
- `GET /runs/{id}` → reproducible execution details (tools called, timings)
- `GET /export/{run_id}` → signed bundle (CSV/XLSX/PNG/PDF/JSON)

Exports

- CSV/XLSX for tables, PNG/PDF for charts, JSON for programmatic use, "answer bundle" zip (chart + table + SQL + metadata + README).

19) Security & Privacy (detailed)

- **RLS** on facts; **view allow-list**; **PII masking** in canonical views.
- **Role-based scopes**: Analyst, Manager, Executive, QA; temporary privilege escalation requires approval.
- **Query budgets** per session; auto-cancel long-running scans; daily user/role limits.
- **Audit**: store prompt, SQL, bindings, rowcount, runtime, user, export actions.

20) Evaluation & QA

- **Gold Set**: 150+ real questions with ground-truth SQL and expected charts.
- **Red-Team**: ambiguous phrasing, timezone edge cases, PII leakage attempts.
- **Scorecard**: Plan/Exec Success, Correctness, Insight Utility, Latency.
- **Feedback Loop**: failed plans triaged → add/adjust few-shots, expand catalog, update canonical views.

21) Roll-out Plan (1 Month)

Week 1: Metrics YAML + 4 canonical views; ingest unstructured sample; "Answer Card" skeleton with SQL & Sources.
Week 2: SQL tool + validator; visualisations; exports (CSV/XLSX/PNG/PDF); provenance/staleness.
Week 3: Anomaly + contribution; alerts; topic extraction; drill-downs.
Week 4: Security hardening (RLS/PII), critic pass, gold-set eval, pilot with founders + QA team.

22) Open Questions

- Multilingual STT/OCR coverage at launch (TR, FR, ES)?
- Per-tenant working hours/holidays source of truth?
- Media retention SLAs and de-redaction approval workflow?

23) Backlog (prioritised)

1. Metric dictionary UI.
2. Watchlists and proactive digests.
3. Forecasts & cohorts.
4. Counterfactual “what-if”.
5. Auto-Notebook export.
6. Property/City hierarchy management.

24) Appendices

A. Prompting System & Few-shots (maintained in repo).
B. OpenAPI Schemas (drafts for `/ask`, `/alerts`, `/runs`, `/export`).
C. Chart Grammar (Vega-Lite templates).
D. SQL Validation Rules (sqlglot/sqlfluff configs).
E. Data Contracts & Fixtures (for correctness tests).