

Axiom Quant Trading System

Client Handover Report (Book 1)

Version: 1.0
Repository: `/Users/madhuram/trading_bot`
Prepared for: Client Handover
Prepared by: Engineering Team
Date: 2026-02-19

1) Executive Summary

This project delivers a production-oriented algorithmic trading platform focused on:

1. Deterministic and auditable decision-making.
2. Strict fail-closed runtime controls.
3. Operational observability for every critical gate.
4. Controlled separation between `LIVE` and `PAPER/SIM`.

The system now runs with:

1. Structured feed and gate telemetry.
2. Explicit blocker diagnostics in dashboard and logs.
3. Safer expiry selection and token/session handling.
4. Reduced ambiguity around “why no trade” outcomes.

2) Project Scope and Deliverables

Delivered Scope

1. Live market data ingestion and freshness tracking.
2. Indicator + regime + strategy-gate pipeline.
3. Trade candidate generation with strict filter chain.
4. Readiness and governance fail-closed choke points.
5. Trade and audit persistence in SQLite + JSONL logs.
6. Streamlit operational dashboard.
7. Daily audit/reporting and diagnostics scripts.
8. Extensive automated tests for critical paths.

Out of Scope

1. Exchange-side guaranteed order execution quality.
2. Market alpha guarantees.
3. Broker availability guarantees.

3) Current Architecture Summary

Runtime Flow

1. `main.py` starts orchestrator and safety controls.
2. `core/orchestrator.py` runs symbol cycles and stage gates.
3. `core/market_data.py` builds immutable market snapshots.
4. `core/strategy_gatekeeper.py` evaluates strategy family eligibility.
5. `strategies/trade_builder.py` attempts candidate construction.
6. `core/readiness_gate.py` and `core/governance_gate.py` enforce runtime permissions.

7. `core/trade_store.py`, `core/tick_store.py`, `core/depth_store.py` persist state.
8. `dashboard/streamlit_app.py` provides operations and debugging views.

Key Safety Invariants

1. `LIVE` mode remains fail-closed by default.
2. Missing critical dependencies block trades, not just warnings.
3. Every rejection path should emit an observable reason.

4) Major Features Delivered

4.1 Data and Feed Reliability

1. WebSocket feed lifecycle controls and stale detection.
2. Depth and tick freshness telemetry.
3. Future-skew tolerance for timestamp classification.
4. Feed diagnostics logs for payload schema and ingest errors.

4.2 Trading Decision and Risk Pipeline

1. Regime-aware strategy family routing.
2. Indicator freshness and warm-up gating.
3. Candidate filters: quote quality, spread, premium, OI/IV, score, lifecycle.
4. Controlled fallback for `PAPER/SIM` without weakening `LIVE`.

4.3 Governance and Auditability

1. Readiness gate + governance gate as explicit execution choke points.
2. Structured gate logs (`gate_status.jsonl`) per cycle/stage.
3. Blocked candidate logs (`blocked_candidates.jsonl`) with reason codes.
4. Decision and audit event logging pathways.

4.4 Operations and Dashboard

1. Unified operational dashboard sections:
 - Home readiness
 - Data & SLA
 - Execution
 - Reconciliation
 - ML/RL
2. “What Blocked Trades Today” now reads active reject logs, not only legacy files.
3. Day-type history reader compatibility improvements.

5) Problems Encountered and Resolution Register

This section records major production-impact defects addressed during delivery.

5.1 Lock and Process Ownership Issues

Problem:

1. Repeated `RUN_LOCK_ACTIVE` due to duplicate process ownership and respawn scripts.

Resolution:

1. Consolidated single-owner feed behavior.
2. Strengthened run-lock handling and diagnostics.

3. Added tests around stale/dead lock scenarios.

5.2 Feed Restart and Token Mismatch

Problem:

1. Auth validation passed while WS used stale/mismatched token source.
2. Restart storms on close/error created unstable loops.

Resolution:

1. Unified resolved token usage for WS startup.
2. Improved close/restart guard paths and policy.
3. Added tests for token path and restart behavior.

5.3 DB Path Drift and SQLite Reliability

Problem:

1. Inconsistent DB path defaults (`data/trades.db` vs desk DB) created split-brain behavior.
2. Relative path and CWD drift caused “unable to open database file”.
3. Hot-path DB usage risked file descriptor exhaustion.

Resolution:

1. Canonicalized DB path resolution behavior.
2. Added fail-fast split-brain checks.
3. Hardened connect/open diagnostics and write path handling.

5.4 Missing Decision and Audit Data Crashes

Problem:

1. Daily audit failed when decision events or truth dataset were absent.

Resolution:

1. Implemented skip contracts with explicit artifacts and reasons.
2. Preserved fatal behavior for unexpected/real errors.

5.5 Import and Script Execution Fragility

Problem:

1. Script execution failed from varying working directories.

Resolution:

1. Package markers and module-run conventions.
2. Script runtime path hardening.
3. Daily ops subprocess invocation stabilization.

5.6 Trade Starvation from Missing Index Bid/Ask

Problem:

1. Indices often have missing depth bid/ask; strict quote checks rejected all candidates.

Resolution:

1. In `PAPER/SIM`, synthetic index bid/ask from LTP with explicit markers.
2. In `LIVE`, still fail-closed on missing true quote fields.
3. Added unit tests for both branches.

5.7 Observability Gap: Blocked Candidates Not Visible

Problem:

1. Dashboard showed “No blocked candidates” despite trade-builder rejections.

Resolution:

1. Added lightweight structured logger in trade builder for rejection events.
2. Dashboard now reads desk-scoped blocked candidates log and supports legacy fallback.
3. Added tests for `missing_index_bid_ask` and `no_signal`.

5.8 Expiry Selection Mismatch

Problem:

1. Config weekday hints conflicted with actual available exchange expiries.

Resolution:

1. Selection now prefers actual available expiries.
2. Weekday mapping used only as fallback.
3. Defaults aligned to current weekly expiries (NSE Tuesday, Sensex Thursday).

6) Strengths and Improvements Achieved

1. Higher observability: root-cause logs at gate and candidate levels.
2. Better mode safety: strict `LIVE`, controlled `PAPER/SIM`.
3. Better data integrity: timestamp, DB path, and ingestion diagnostics.
4. Improved audit continuity and reduced false operational failures.
5. Better client/operator explainability: explicit reason codes in UI and logs.

7) Known Constraints and Residual Risks

1. Broker/exchange dependencies remain external runtime risk.
2. Large script surface area still requires disciplined release control.
3. Strategy performance depends on market conditions and model quality, not software correctness alone.
4. Runtime tuning requires controlled rollout with regression tests.

8) Validation and QA Coverage

The project includes broad test coverage in `/Users/madhuram/trading_bot/tests`, including:

1. Feed health and restart behavior.
2. Readiness and governance gates.
3. Strategy gatekeeper and trade-builder behavior.
4. Risk and exposure controls.
5. Auth and token consistency.
6. Data/audit/reporting contracts.

Recent additions include:

1. `tests/test_trade_builder_blocked_candidates.py`
2. `tests/test_trade_builder_trend_vwap_fallback.py`
3. `tests/test_expiry_selection.py`

9) Deployment and Operations Handover

Recommended Operational Commands

1. Validate environment:

- `bash /Users/madhuram/trading_bot/scripts/ci_sanity.sh`
- 2. Validate auth/session:
 - `PYTHONPATH=. python /Users/madhuram/trading_bot/scripts/validate_kite_session.py`
- 3. Start stack:
 - `bash /Users/madhuram/trading_bot/scripts/start_live_stack.sh`
- 4. Start dashboard:
 - `PYTHONPATH=. streamlit run /Users/madhuram/trading_bot/dashboard/streamlit_app.py`
- 5. Run tests:
 - `PYTHONPATH=. pytest -q`

Runtime Artifacts to Monitor

- 1. `logs/desks/<DESK_ID>/gate_status.jsonl`
- 2. `logs/desks/<DESK_ID>/blocked_candidates.jsonl`
- 3. `logs/depth_ws_watchdog.log`
- 4. `logs/sla_check.json`
- 5. `logs/incidents.jsonl`

10) Handover Acceptance Checklist

- 1. Auth validation succeeds.
- 2. Feed and SLA fresh during market-open windows.
- 3. Dashboard reflects gate reasons and blocked candidates.
- 4. Daily audit runs without crash.
- 5. Regression tests pass on release candidate.

11) Client Recommendations for Next Phase

- 1. Keep `LIVE` strictness unchanged; tune only in `PAPER/SIM` first.
- 2. Add weekly blocker reason analytics review.
- 3. Maintain release checklists with mandatory regression runs.
- 4. Version formal data contracts for UI and reporting artifacts.

12) Appendix Reference

Full repository file inventory is provided in:

`/Users/madhuram/trading_bot/docs/APPENDIX_FILE_INDEX.md`