GOVERNANCE & RESPONSIBLE AI MODEL · EDITION 2026.02

Autonomy you can govern is the only autonomy worth deploying.

This document specifies how ZeroMan.ai bounds, audits, and earns autonomy — for its own agents today, and for third-party modules in the marketplace era. It is written to be held against us: every commitment here is testable in the product.

> POSITION
> Full autonomy without governance is a liability. Recommendations without execution are theater. The only credible enterprise AI is a system that acts inside an explicit decision-rights envelope — and can prove it.

1 · Principles

01 **Bounded autonomy**
No agent or module acts outside declared decision rights, thresholds, and data scopes.

02 **Human ownership of strategy**
People own objectives, policies, thresholds, and exceptions. The system owns coordination.

03 **Explainability by construction**
Every recommendation carries its assumptions, binding constraints, and trade-offs — generated with the decision, not reconstructed after it.

04 **Auditability**
Every decision run produces an immutable, replayable record. If it isn't logged, it didn't happen.

05 **Reversibility & override**
Humans can pause, override, or roll back any automated action class at any time — and overrides are themselves logged and learned from.

06 **Proportionality**
The autonomy granted to a decision class is proportional to its measured reliability and bounded blast radius — never to its convenience.

2 · Autonomy levels

LEVEL	NAME	BEHAVIOR	TYPICAL USE
L1	Recommendation	Analyzes, explains, proposes. No action.	New loops; low data confidence.

L2	Approve to act	System prepares the action; a named human approves before execution.	Default posture for all new deployments.
L3	Bounded autonomy	Executes automatically within policy, thresholds, and blast-radius limits.	Proven, low-risk action classes.
L4	Enterprise autonomy	Governed loops run end to end; humans own strategy, policy, and exceptions.	Long-term destination, earned per domain.

HOW AUTONOMY IS EARNED — AND LOST

Promotion: a decision class moves up one level only after N consecutive runs approved without material correction, within defined error bounds, over a minimum calendar period — all three parameters set per class with the customer. Demotion: any threshold breach, material correction, or anomaly automatically demotes the class one level and opens a review. Blast-radius limits (value caps, scope caps, rate caps) apply at every level above L1.

3 · Decision rights & approval matrix

Decision rights define who may approve which action type under which conditions. The matrix below is the reference pattern; every deployment instantiates its own version during the design partnership.

DECISION TYPE	APPROVAL LOGIC	CONTROL CLASS
Replenishment adjustment	Allowed automatically below value threshold	Autonomous (L3-eligible)
Purchase order change	Requires approval above cost limit	Cost threshold
Production schedule change	Requires approval if customer impact exists	Customer impact
Inventory reallocation	Allowed within policy; flagged if cross-region	Policy bounded
Freight expedite	Requires approval above expedite-cost threshold	Cost threshold
Customer allocation	Requires approval if strategic customers affected	Customer impact
Supplier substitution	Requires approval if contract or compliance risk exists	Policy bounded
Financial trade-off	Executive approval if margin/service trade-off exceeds policy	Executive

4 · The audit envelope

Every decision run — recommendation or execution, human-approved or autonomous — writes a complete, immutable telemetry record:

```
TELEMETRY RECORD — REQUIRED FIELDS

decision_id · loop_class · trigger_signal · severity
state_snapshot_ref · constraints[] · binding_constraints[]
agents_invoked[] · models_used[] · assumptions[]
scenarios[] · tradeoffs · recommendation · rationale
governance: rights_applied · thresholds_checked · approver · timestamp
actions_prepared[] · execution_channel · execution_status
outcome: realized_metrics · variance · decision_quality_score
overrides[] · corrections[] · review_flags[]
```

Records are retained per customer policy, exportable on request, and designed to satisfy internal audit, risk, and compliance review without reconstruction.

5 · Override, escalation, and the stop switch

- **Pause:** any decision class, loop, or the entire platform can be paused instantly by authorized roles; in-flight prepared actions are held, never half-executed.
- **Override:** approvers can modify or reject any prepared action; the override, its reason, and its outcome enter the learning loop.
- **Escalation:** unresolved approvals escalate on a timer through named tiers (owner → director → executive) — silence never equals consent.
- **Stop switch:** a hard kill per integration channel guarantees no writes reach any system of record while engaged.

6 · Marketplace-era governance

As ZeroMan.ai opens to third-party modules, the same discipline extends to the ecosystem:

- **Data-scope manifests.** Every module declares exactly what it reads and writes — app-style permissions, enforced by the core, visible to the customer. No scope, no data.
- **Certification.** Modules pass conformance (contracts honored, telemetry emitted, failure modes handled) before listing; certification is renewable, not permanent — and is administered through Loop Zero, ZeroMan's agent-led onboarding.
- **Cross-provider isolation.** No module ever sees another provider's data, prompts, models, or telemetry. Isolation is enforced by the platform, not by policy documents.
- **Measured ratings, published methodology.** Provider scorecards are computed from orchestration telemetry against ground-truth outcomes; the methodology is public, scores are not for sale, and providers have a documented appeal process.
- **The neutrality rule.** ZeroMan's own modules are built exclusively on the public APIs available to every third party — no private hooks, no privileged data, no preferential ranking. The referee does not play with house advantages.

7 · Responsible AI commitments

- Customer data is not used to train shared or foundation models; learning is per-tenant unless explicitly agreed otherwise.
- Model and assumption traceability for every recommendation; no unattributable decisions.
- Human-impacting decisions (e.g., labor scheduling, supplier termination) default to L1/L2 regardless of measured performance.
- Bias and drift monitoring on forecasting and allocation models, with documented review cadence.
- Incident disclosure: material automation incidents are disclosed to affected customers with root cause and remediation.
- We publish this document and invite customers, partners, and researchers to hold us to it.

8 · Scope

ZeroMan.ai is early-stage; the platform is being designed and built. This document specifies the governance model the platform is built against — it does not claim certifications, customers, or production deployments. Questions, critique, and red-team interest are welcome: zeroman.ai/contact.