

— PLATFORM BRIEF · FOR ANALYSTS, TECHNICAL EVALUATORS, AND DESIGN PARTNERS

The agentic operating layer for Consumer Goods supply chains — and the kernel it becomes.

This brief explains what ZeroMan.ai is being designed to coordinate, why the decision layer matters, how governance, optimization, telemetry, and integration fit together — and how the architecture compounds from a wedge application into an enterprise operating platform.

PURPOSE & AUDIENCE

Written for people evaluating architecture, not marketing: analysts, technical buyers, design partners, and investors performing technical diligence. The right evaluation is architectural — does the proposed decision layer clarify operating scope, governance, integration, telemetry, and value paths?

Sense Decide Optimize Govern Execute Learn

Plan Source Make Deliver Return Finance Govern

1 · Operating thesis

Supply chains are coordination systems, not only data systems. Existing platforms record transactions, support planning workflows, and expose dashboards. The operational gap is the space between a signal and a governed action — today that space is filled by people, spreadsheets, and meetings.

ZeroMan.ai is being designed to turn that space into a continuous, governed decision loop:

THE DECISION LOOP

signal -> state -> reasoning -> optimization -> governance -> execution -> learning

Two consequences follow. First, the unit of product is the **decision loop**, not the feature: a loop either closes — measurably, under policy — or it does not. Second, the layer must live **above** the systems of record, because no single system sees the whole decision.

2 · Scope — the wedge and the operating model

ZeroMan.ai enters with the highest-frequency, lowest-blast-radius loops in mid-market Consumer Goods (roughly \$20M–\$1B revenue): **inventory replenishment exceptions** and **MRP exception resolution**, followed by allocation and order promising. The architecture, however, is scoped to the full operating model from day one, so loops can be added without re-platforming:

DOMAIN	SCOPE
Plan	Demand sensing, supply planning, MRP exceptions, inventory policy, scenario planning.
Source	Procurement coordination, supplier follow-up, inbound risk, alternate supply response.
Make	MPS, production planning, scheduling, capacity, changeovers, material feasibility.
Deliver	Allocation, order promising, outbound logistics, delivery exceptions, service impact.
Return	Returns, reverse logistics, claims, quality issues, recovery actions.
Finance	Cost, margin, working capital, cash impact, trade-off analysis.
Govern	Decision rights, autonomy thresholds, audit, approval, explainability, override.

3 · The operating stack (L1–L8)

Agents are only the visible layer. The platform is the governed decision architecture connecting data, a live twin, optimization, execution, and learning:

LAYER	ROLE	ELEMENTS
L8 · Learning	Compounds decision quality	Outcomes, assumption updates, policy refinement, model updates.
L7 · Execution	Acts through existing systems	ERP updates, POs, plan changes, allocations, workflow tasks.
L6 · Governance	Bounds autonomy	Decision rights, thresholds, audit, explainability, override.
L5 · Optimization	Generates feasible options	MILP/LP, heuristics, scenarios, sensitivity, trade-offs.
L4 · Role agents	Reason per function	Demand, supply, MRP, production, inventory, logistics, finance, risk.
L3 · Digital twin	Holds decision-relevant state	State, constraints, risks, available decisions, scenario impact.
L2 · Semantic core	One governed truth	Items, BOMs, orders, forecasts, capacity, cost, policies, contracts.
L1 · Enterprise systems	Systems of record	ERP, APS, WMS, TMS, MES, SRM, CRM, finance, portals.

4 · A worked decision run (illustrative)

ILLUSTRATIVE SCENARIO

All figures below are illustrative design targets used to specify the system's behavior. They are not measured results from any deployment.

Signal. Monday 06:40 — sell-through on a priority SKU family runs +38% above its rolling forecast band for the third consecutive day. The Demand Agent classifies the anomaly and opens decision **DR-1042**.

State. The twin snapshots the operating state: 11 days of cover at DC-North against a 14-day policy floor; Line 2 at 81% utilization this week; one inbound packaging shipment at supplier risk; strategic customer orders due Thursday.

Scenarios. The optimization core returns three feasible responses:

SCENARIO	SERVICE	INCREMENTAL COST	INVENTORY	RISK
A — Protect service	99.1%	€41k (expedite + overtime)	Tight, within floor	Low
B — Minimize cost	94.6% (at risk)	€6k	Higher downstream	Elevated
C — Balanced (recommended)	98.2%	€18k	Controlled	Managed

Governance. Scenario C includes a freight expedite above the €15k threshold, so the run routes to the supply chain director with a one-page decision memo: decision, rationale, trade-offs considered, and required approval. On approval, actions are prepared in the ERP — a production priority change, a stock reallocation, and one expedited inbound line.

Learning. The loop stays open until outcomes land: realized service, actual expedite cost, and forecast error feed assumption updates and the decision-quality score for this loop class.

DECISION-RUN TELEMETRY (EXCERPT)

```

decision_id: DR-1042           loop: replenishment.exception
signal: demand.band_breach    severity: P2
state_snapshot: twin@06:41:12 constraints: 14 binding: 3
scenarios: 3                  recommended: C (balanced)
governance: expedite>EUR 15k -> approval: SC_Director
actions_prepared: 3           execution: ERP.prepare_action
learning: open                 outcome_due: T+14d

```

5 • Optimization core

Optimization is not an add-on; it is the engine that makes agent reasoning **feasible**. The core generates scenarios under capacity, material, inventory, service, cost, margin, and policy constraints; balances local against global objectives; and explains **why** a scenario is recommended, which assumptions it depends on, and which constraints bind the decision.

- Methods as appropriate per loop: MILP/LP, heuristic scenario generation, sensitivity analysis, simulation.
- Every recommendation ships with its binding constraints and assumptions — explainability by construction.
- Model selection itself is recorded in telemetry, so model performance is auditable over time.

6 • The semantic core — from twin to kernel

The platform needs one governed representation of operating truth. In the wedge, that is a decision-scoped digital twin: products, SKUs, BOMs, routings, plants, warehouses, suppliers, customers, orders, forecasts, inventory, capacity, cost, margin, lead times, policies, constraints — the state required to evaluate actions, not a visualization of everything.

That same layer is designed to compound into the **kernel** of an AI-native operating platform: master data management, identity, permission scopes, versioned data contracts, and an event bus — exposed through open protocols (MCP-native) so any approved tool or module connects through one governed interface. We adopt open standards for the wire; the differentiation is the semantic model, the permission scopes, and the accumulated outcome data — not a proprietary format.

MODULE DATA-SCOPE MANIFEST (CONCEPT)

```
module: "mrp.planner"           provider: "third_party"
reads: [items, boms, inventory.positions, orders.open, capacity]
writes: [plans.mrp.prepared]    # prepare-action only
forbidden: [finance.ledger, hr.*, customers.pii]
telemetry: required            rating: outcome-scored
```

7 · Governance gates

Enterprise autonomy requires policy controls **before** action: decision rights define who may approve which action type under which conditions; thresholds cover cost, customer impact, financial exposure, and compliance risk; human override, explainability, and immutable audit trails keep autonomy bounded and accountable. Autonomy levels run from L1 (recommend) through L4 (enterprise autonomy), and each decision class earns promotion through measured performance. The full model — including promotion and demotion criteria and marketplace-era neutrality rules — is specified in the companion document **Governance & Responsible AI Model (Edition 2026.02)**.

8 · Telemetry, learning, and measured ratings

Every decision run produces a traceable record: signal, state snapshot, agents invoked, assumptions, constraints, scenarios, trade-offs, approvals, prepared actions, execution channels, and outcomes. The learning layer turns outcomes into assumption updates, policy refinement, and decision-quality feedback.

Telemetry has a second job: it is the substrate of **measured provider ratings**. Because the platform routes the flows and holds ground truth, module performance — forecast fit, plan adherence, SLA, cost per decision — can be scored continuously and objectively. Reviews rate software; this architecture measures it.

9 · Integration footprint

ZeroMan.ai is designed to operate above, and through, existing systems — read paths bring state into the semantic core; write paths prepare actions only inside approved governance boundaries.

SYSTEM CATEGORY	READ PATHS (STATE IN)	PREPARE-ACTION PATHS (OUT)	TYPICAL INTERFACES
ERP	Items, BOMs, inventory, orders, costs	PO changes, plan updates, allocations	OData / BAPI / REST / files

APS / planning	Forecasts, plans, capacity	Plan adjustments	REST / files
WMS / TMS	Stock positions, shipments, ETAs	Reallocation, expedite tasks	REST / EDI / events
MES	Line status, schedules, changeovers	Priority changes (prepared)	REST / OPC-UA / files
Finance	Cost, margin, working capital	Trade-off inputs to approvals	REST / files
Portals & collaboration	Supplier confirmations, customer signals	Follow-ups, workflow tasks	Email / API / webhooks

10 · Security & data handling posture

DESIGN COMMITMENTS – NOT CERTIFICATIONS

ZeroMan.ai is early-stage and makes no certification claims. The commitments below are architectural principles the platform is being built against, stated so evaluators can hold us to them.

- Tenant isolation by design; least-privilege, manifest-declared data scopes for every module and agent.
- Encryption in transit and at rest; secrets management and key rotation as standard practice.
- Immutable audit logs for every decision run; model and assumption traceability end to end.
- Customer data is not used to train shared or foundation models.
- Regional data-residency options planned for EU deployments; deletion and export on request.
- Third-party modules never see data outside their declared scope; cross-provider isolation is enforced by the core.

11 · Implementation — Loop Zero and the 90-day design partnership

Onboarding is a product, not a services engagement. **Loop Zero** is ZeroMan's AI onboarding agent — and it is itself a governed decision loop: it connects systems with least-privilege scopes, runs schema mapping with confidence scores, walks the data-quality report card, conducts the conversational governance interview, configures read-only shadow mode and backtests, and tracks go-live readiness. It proposes; named humans approve; every step is logged. The first thing a customer experiences is the platform's own philosophy at work.

- **Small businesses:** near-autonomous setup, end to end.
- **Mid-market:** Loop Zero leads, your team confirms — first loop live in days to weeks, with no external consultancy.
- **Complex enterprises:** Loop Zero orchestrates your internal team — tasks, owners, runbook artifacts. External partners can accelerate; they are never required.

The paid 90-day design partnership is the high-touch edition of the same program, and every engagement feeds its artifacts — mappings, governance presets, archetype templates — back into Loop Zero:

01 Weeks 0–2 · Map

Operating context, systems, constraints, decision rights, priority loops.

02 Weeks 3–6 · Model

Twin entities, scenario logic, optimization requirements, data availability.

03 Weeks 7–10 · Govern

Governance gates, integration architecture, telemetry schema, user surfaces.

04 Weeks 11–13 · Pilot path

Pilot backlog, success measures, operational owners, scale plan.

In the store era, the same agent onboards providers: module registration, data-scope declaration, contract mapping, sandbox conformance, certification checks, pricing setup, and marketplace publication — with measured telemetry from the first run.

12 · Scope & trust posture

Honest scope is part of the operating model. ZeroMan.ai is early-stage; the platform is being designed and built. No certifications, customers, or production deployments are stated or implied. Reference models published by ZeroMan.ai are composite and illustrative unless explicitly stated otherwise.

NEXT STEP

Start a strategic conversation: zeroman.ai/contact · zeroman.ai/partner-program