KAIZEN AI

KAIZEN AI · GOVERNANCE BLUEPRINT

The Oracle Agentic AI Governance Blueprint

Policy-as-Code, Audit Trails, Rollback Architecture, and Human Control for EBS, Fusion, JDE, PeopleSoft, and NetSuite Environments.

Method. Measure. Momentum.

Governed agentic AI for Oracle-powered enterprises.

kaizen-orbit.lovable.app · info@kaizenai.ai

EXECUTIVE SUMMARY

Governance Is the Wedge — Not the Obstacle

Most AI failures in Oracle-heavy enterprises are not model failures. They are governance failures. The pilot is technically sound. The agent produces accurate recommendations. And then it touches a record in a SOX-covered Oracle module — without a documented entitlement model, without policy-as-code, without a pre-engaged audit team — and the entire program is suspended pending IT control remediation.

Kaizen AI's position is the inverse of the conventional framing: governance is not the obstacle to agentic AI deployment. It is the wedge. The organization that builds the governance architecture first deploys agents faster, sustains them longer, and scales them wider — because it has built the infrastructure that every subsequent agent can reuse.

This blueprint covers all eight layers of the Kaizen AI governance architecture: identity and entitlement, policy-as-code, pre-action validation, human-in-the-loop design, execution telemetry, rollback and reversibility, drift monitoring, and audit export. It also covers the operating model — the tripartite ownership structure that keeps agentic AI safely in production after go-live.

8	3	100%	On-demand
Governance layers	Named operating model owners	Write-backs through policy-as-code	SOX evidence package generation

The Business Problem: Why Governance Failures End AI Programs

Enterprise AI governance failures have a consistent pattern. An AI pilot is built outside the formal IT control framework — fast, effective, impressive. It is then proposed for production deployment in an Oracle-covered workflow. At that point, the CISO reviews the access model and finds a shared admin account. Internal audit reviews the evidence model and finds no structured audit trail. The Oracle application owner reviews the write-back mechanism and finds no rollback procedure. The program is suspended.

The failure is not the AI. The failure is that governance was treated as a post-deployment problem. In Oracle environments under SOX, ITIL, or regulatory scrutiny, governance must be designed before the first line of agent code is written — not retrofitted after the pilot succeeds.

Oracle Governance Challenges by Platform

Oracle Platform	Key Governance Challenges	SOX / Audit Implications	Kaizen AI Approach
Oracle EBS R12	Customized responsibilities; row-level security complexity; BPEL integration auditability	Agent activity must be separable from human activity in Oracle audit trail	Named service account with custom responsibility; Business Events for trigger; all writes via Web Services with audit ID
Oracle Fusion Cloud	RBAC complexity; REST API authentication; BPM Worklist approval integration	Fusion audit trail must capture agent identity distinctly from end-user identity	Dedicated application user; OAuth 2.0 scoped tokens; BPM Worklist integration for HITL approvals
JD Edwards EnterpriseOne	Orchestrator security model; AIS token management; Business Function auditability	JDE security roles for orchestrator must be scoped; all actions logged in JDE Work Center	Orchestrator-native deployment; AIS Server with scoped tokens; Work Center integration for audit
PeopleSoft	Integration Broker security; Component Interface restrictions; complex role inheritance	PeopleTools audit logging must capture agent activity; SoD analysis of Component Interface access	Integration Broker service operations with dedicated permissions; Component Interface access scoped by business process
NetSuite	REST API token management; SuiteScript execution context; sandbox vs. production parity	NetSuite SuiteAudit must capture agent writes; token management must satisfy SOC 2 requirements	Token-Based Authentication with scoped roles; SuiteAudit integration; sandbox-first deployment model

The Eight Governance Layers

The Kaizen AI governance architecture has eight layers, each with a specific function, design requirement, and Oracle touchpoint. Together they constitute the governance scaffold that every subsequent agent reuses.

Layer 1: Identity and Entitlement

Domain: IDENTITY

Principle: Every agent action carries a service-account identity bound to a scoped Oracle responsibility. The agent never operates under a shared account. Its activity is distinctly attributable in the Oracle audit trail.

Design requirements: Oracle DBA provisions a dedicated service account (EBS: named responsibility; Fusion: named application user + RBAC roles; JDE: named user + business unit access; PeopleSoft: Integration Broker service account). The scoped permission set is documented before agent development begins. SOD analysis is conducted before the service account is provisioned. The service account is rotated on the same schedule as privileged accounts.

> **Why This Layer Matters**
>
> A shared account makes audit attribution impossible. An overprivileged account expands the blast radius of any error.

Layer 2: Policy-as-Code

Domain: POLICY

Principle: Every decision rule the agent follows — approval thresholds, dollar limits, SoD constraints, blackout windows, rollback conditions — is expressed in a structured, version-controlled format that can be audited, tested, and changed through formal change control.

Design requirements: Policy is expressed in JSON schema or YAML with defined fields for: action type, approval authority, dollar threshold, SoD constraints, rollback window, blackout conditions, and escalation path. Policy is stored in version control (Git). Every change to policy goes through a pull request with named reviewer approval. The policy version active at the moment of any agent action is logged with that action.

> **Why This Layer Matters**
>
> Policy drift — the gap between what the policy says and what the agent does — is the primary source of agentic AI compliance failures.

Layer 3: Pre-Action Validation

Domain: VALIDATION

Principle: Before executing any write-back to Oracle, the agent simulates the action against current ledger, inventory, or master data to confirm that the action is valid at the moment of execution — not just at the moment of proposal.

Design requirements: Pre-action validation checks: current state of the Oracle record (has it changed since the agent last read it?), policy compliance of the proposed action at current values, SoD constraint satisfaction, rollback feasibility. If any validation fails, the agent escalates rather than proceeding. The validation result is logged as part of the action's audit record.

> **Why This Layer Matters**
>
> A proposal generated on stale data can produce incorrect actions even when the policy is correct. Pre-action validation catches state changes between proposal and execution.

Layer 4: Human-in-the-Loop

Domain: HITL

Principle: Every agent action above a defined risk threshold routes to a named human approver role with a structured approval request, risk factors, and recommended decision. The approver makes a decision; they do not check the agent's work.

Design requirements: Approval request structure: triggering event (with Oracle record reference), data correlated (with sources), proposed action (with policy basis), risk factors (with mitigations), recommended decision, and rejection consequence. Approval SLAs are defined by action type. SLA breaches auto-escalate to the next approver role. Approver identity, timestamp, and decision are captured in the audit record.

> **Why This Layer Matters**
>
> An ill-designed HITL creates either latency that negates cycle-time benefit or rubber-stamping that creates the illusion of oversight without the reality.

Layer 5: Execution Telemetry

Domain: TELEMETRY

Principle: Every Oracle object touched, every record changed, and every system called by the agent is captured in an immutable execution telemetry record with a stable audit ID.

Design requirements: Telemetry record contains: stable audit ID (UUID), triggering event, all Oracle records read (with timestamps), all external systems queried, policy version applied, approval record (if required), Oracle records written (with before and after values), execution timestamp, agent version. Telemetry is written to an append-only log. Kaizen AI audit log is accessible to internal and external auditors using the same access controls as Oracle standard reports.

> **Why This Layer Matters**
>
> Without complete execution telemetry, the SOX evidence package cannot be assembled. Partial telemetry creates evidence gaps that auditors will find.

Layer 6: Rollback Architecture

Domain: RESILIENCE

Principle: Every write-back action ships with an explicit undo path, a defined rollback window, and a named Oracle responsibility with the permissions to execute the rollback.

Design requirements: For each agent write-back action: document the standard Oracle reversal transaction, the rollback window (time from action to point of irreversibility), the named Oracle responsibility required to execute the rollback, and the escalation path if the rollback window has closed. Test rollback in non-production before go-live. Rollback procedures are part of the operating charter and the agent's policy-as-code.

> **Why This Layer Matters**
>
> An agent that cannot be reversed cannot be trusted. Rollback capability is the primary risk control that enables executive authorization of autonomous action.

Layer 7: Drift Monitoring

Domain: MONITORING

Principle: Continuous evaluation of the agent's behavior against KPI thresholds and policy boundaries, with auto-pause if the agent's behavior regresses against baseline.

Design requirements: Drift monitoring tracks: action distribution (is the agent handling the right types of exceptions?), approval rate (is the HITL being bypassed more often than the policy intends?), cycle-time trend (is the performance improvement holding?), error rate (is the agent encountering unhandled edge cases more frequently?). Auto-pause conditions are defined in policy-as-code. Monthly drift reports are reviewed by the AI Ops Lead and business sponsor.

> **Why This Layer Matters**
>
> Agents that are not monitored drift from their intended behavior over time as Oracle data patterns change. Drift detection is the operating model's primary quality control.

Layer 8: Audit Export

Domain: AUDIT

Principle: SOX- and ITIL-aligned evidence packages are generated on demand for any covered control, for any date range, without manual assembly.

Design requirements: Audit export covers: complete action log for the selected control and period, policy version active during the period, all approval records (approver, timestamp, decision), all Oracle writes (before/after values, API endpoint, service account), drift monitoring reports for the period, rollback events (if any) with resolution records. Audit export format matches the internal audit team's standard evidence template. External auditor access is provisioned as read-only with access logging.

> **Why This Layer Matters**
>
> Audit evidence that must be manually assembled is evidence that will be incomplete, late, or wrong. Automated audit export is the single greatest efficiency gain from governed agentic AI for finance teams.

SOX and ITIL Compliance Architecture

Agentic AI in Oracle environments almost always touches SOX-covered processes. The Kaizen AI governance architecture is designed to satisfy the specific requirements of SOX 404 IT general controls, ITIL change management, and standard enterprise audit frameworks.

Compliance Framework	Relevant Agent Controls	Kaizen AI Governance Layer	Evidence Generated

SOX 404 ITGC	User access reviews; change management; automated controls	Identity & Entitlement; Policy-as-Code; Audit Export	Service account scoping documentation; policy version log; complete action audit trail
SOX 404 Process Controls	AP approval authority; journal entry review; reconciliation	Policy-as-Code; HITL; Execution Telemetry	Approval authority matrix; approver identity + timestamp; before/after values for all Oracle writes
ITIL Change Management	Agent deployment as IT change; policy updates as changes	Policy-as-Code version control; deployment change records	Git commit history for policy; change record for each agent deployment
Internal Audit	Control effectiveness testing; evidence completeness	Audit Export; Drift Monitoring	On-demand evidence packages; monthly drift reports

The Governance Operating Model

A governed agent has three named owners, each with distinct accountability. This tripartite model is installed in the first engagement and handed off as a permanent capability.

Owner Role	Accountability	Key Decisions	Meeting Cadence
Business Sponsor	Workflow outcome; cycle-time KPIs; adoption targets; escalation authorization	Which actions require HITL; what constitutes successful deployment; when to expand scope	Monthly with other owners; weekly with process team during first 90 days
Oracle Application Owner	System-of-record integrity; service account governance; entitlement model; Oracle audit trail	Service account scope; what Oracle objects the agent may read and write; patch and upgrade impact on agent	Monthly with other owners; immediate notification of Oracle environment changes
AI Operations Lead	Agent lifecycle; drift monitoring; policy-as-code maintenance; model performance; audit export	Policy updates; agent version updates; auto-pause decisions; drift remediation	Monthly with other owners; weekly drift report review; immediate response to auto-pause events

Where to Start: The High-Evidence First Agent

The governance scaffold is built once and reused across every subsequent agent. The highest-value starting point is a workflow with three characteristics: high frequency (the agent has many opportunities to

demonstrate value), high evidence weight (the governance requirements are demanding, so the scaffold built is comprehensive), and low irreversibility (the rollback window is long enough to catch and reverse errors).

Typical first agents that satisfy all three criteria:

- **AP exception handling:** High frequency (hundreds of exceptions per cycle); strong SOX evidence weight; reversible via standard invoice cancellation within payment run window.
- **Supplier delay response:** High frequency (daily disruption events); moderate audit weight; reversible via order amendment before fulfillment commitment.
- **ServiceNow ticket triage:** Very high frequency; low SOX weight (good for governance scaffold development without compliance pressure); reversible via ticket reassignment.

The governance scaffold built for the first agent becomes the template for every subsequent agent. The compounding effect of scaffold reuse is the single largest determinant of enterprise momentum.

Executive Checklist: Governance Architecture Validation

1. Named Oracle service account with scoped responsibilities — not a shared admin account.
2. Policy-as-code drafted, version-controlled, and reviewed by internal audit.
3. Pre-action validation logic designed and tested for every write-back action.
4. HITL approval workflow built, integrated with Oracle BPM or equivalent, and SLA-tested.
5. Execution telemetry schema defined and audit log provisioned with auditor access.
6. Rollback procedures documented and tested in non-production for every write-back action.
7. Drift monitoring thresholds defined and auto-pause conditions written in policy-as-code.
8. Audit export format agreed with internal audit and external auditors before go-live.
9. Tripartite operating model named, chartered, and committed — not theoretical.
10. SOX control classification agreed with internal audit — agent documented as a tested IT control.

Book the Kaizen AI 3M+ Assessment

The 3M+ Assessment builds and validates the complete governance architecture for your first agent: service account design, policy-as-code authoring, HITL workflow integration, audit export configuration, and operating model charter.

The governance scaffold built in the Assessment becomes the permanent infrastructure for your agentic AI program.

Contact: info@kaizenai.ai · kaizen-orbit.lovable.app/assessment

Method. Measure. Momentum.