Evaluation Guide

How to Determine Whether Your AI is “Enterprise-Grade”

Introduction

“Agentic AI” is quickly becoming table stakes. What’s rarer—and harder—is enterprise-grade agentic AI: systems that don’t just reason and act, but do so safely, audibly, repeatably, and at scale inside real enterprises.

Enterprise-grade is not a UX flourish. It’s an architectural and operational commitment. The gap between a demo that works and a system that ships is measured in operational reality. For agentic AI, "enterprise grade" is a specific set of capabilities, methods, and mindsets that separate production systems from prototypes.

Deep, Bi-Directional Data Integration

Most discussions of agentic AI start with the model. They should start with data.

Enterprise-grade systems require comprehensive bidirectional integration across four data domains:

- Ecosystem partners: CRM platforms, marketing automation, support systems
- Internal on-premise systems: ERP, legacy databases, proprietary applications
- Data lakes: Historical analytics, training datasets, behavioral logs
- Cloud storage: Document repositories, collaboration tools, knowledge bases

"Integration" means more than read access, what we like to call ‘Data in.’ Agents must write back—update records, trigger workflows, and create tasks. Read-only access produces insights. ‘Data out’ produces outcomes.

The critical distinction: ingest both the system of record (the data itself) and system of engagement (the activities and context that created it).

An invoice number matters. So does the email thread, approval chain, and timeline that generated it. Agents trained only on outcomes miss the reasoning that produced them.

Knowledge Architecture: The Compounding Layer

Raw data integration feeds what matters most: your knowledge graph.

Not a vector database. Not a knowledge base. A knowledge graph maps entities, relationships, and context—and compounds value as the data feeding it grows over time.

This is where proprietary data becomes strategic. Every customer interaction, every edge case handled, every correction compounds into competitive advantage. Your context graph should capture:

- Domain-specific entity relationships (not generic ontologies)
- Temporal context (how relationships evolve)
- Decision provenance (why actions were taken)
- Failure patterns (what didn't work and why)

Generic foundation models know the world. Your knowledge graph knows your business. This context compounds and the system gets smarter, not because the model changes, but because enterprise context grows steadily over time.

Security, Access, and Governance

Enterprise buyers don't ask "can your agent do X?" They ask "who can make your agent do X, and how do we prove they should?"

Policy-driven access control means:

- Role-based permissions that mirror organizational hierarchy
- Data-level access controls (agents can't see what humans can't)
- Action authorization (reading a file ≠ deleting a file)
- Audit inheritance (agent actions carry user identity)

Agent security extends beyond access. It includes:

- Prompt injection defenses
- Output validation and sanitization
- Guardrails that prevent unintended actions
- Isolation between agent instances (customer A's agent can't access customer B's data)

Every enterprise has governance requirements. SOC2, GDPR, HIPAA, industry-specific regulations. Your agentic system must enforce these before the agent acts, not after you discover the violation.

The Audit Imperative

"The agent made a mistake" is not an acceptable incident report.

Enterprise-grade systems maintain complete audit trails:

- Every decision (with a reasoning trace)
- Every interaction (input, output, tools used)
- Every response (including rejected alternatives)
- Every failure (with error classification and recovery path)

Audit trails serve three purposes:

1. Compliance: Regulators want explanations, not black boxes
2. Debugging: You can't fix what you can't trace
3. Training: Failed interactions become evaluation datasets

The trail must be immutable, searchable, and retained in accordance with your data governance policies.

(continued on following page)

Evaluation: Domain-Specific, Always On

Generic benchmarks (MMLU, HumanEval) don't predict performance on *your* workflows with *your* data.

Enterprise systems require:

Batteries-included evals for common patterns:

- Factual accuracy against internal knowledge bases
- Tool use correctness (right API, right parameters)
- Safety and compliance checks
- Latency and cost budgets

Custom evals built from proprietary data:

- Customer-specific edge cases
- Domain vocabulary and jargon
- Business logic validation
- Historical outcome alignment

Evaluation isn't pre-deployment testing. It's continuous monitoring. Every production interaction generates evaluation data. Performance degradation should trigger alerts, not customer complaints.

DID YOU KNOW?

We've produced a checklist of questions to ask any vendor you're evaluating.

Access that checklist here.

Model Operations: The Invisible Infrastructure

Users don't care which model answered their question. They care about accuracy, speed, and cost, and they expect consistent, reliable answers.

Determinism and Reliability: Enterprise systems cannot tolerate the same question producing different answers across successive calls. Non-deterministic behavior breaks:

- Compliance workflows: Regulators expect repeatable decisions
- Testing and validation: You can't debug what you can't reproduce
- User trust: Inconsistent answers signal unreliability

Hallucinations: Non-determinism's dangerous cousin: confidently wrong answers. Hallucinations aren't eliminated through prompting alone. They're managed through architectural defense-in-depth:

- Prevention: RAG grounding in verified sources, context graph constraints limiting generation to known entities, input validation rejecting out-of-scope queries.
- Validation: Citation verification, consistency checks against system of record, confidence scoring, contradiction detection against business rules.
- Operational safeguards: Human review for high-stakes decisions, audit trails showing information sources, feedback loops capturing failures for eval improvement.

Zero hallucinations are unachievable, and that’s not the goal. The goal is bounded risk: hallucinations caught before they cause damage, with clear accountability when they occur.

Model routing handles the tradeoff space:

- Route simple queries to fast, cheap models
- Escalate complex reasoning to frontier models
- Cascade through fallbacks when primary models fail
- Track cost per interaction against value delivered

Guardrails prevent expensive mistakes:

- Output validation before action execution
- Confidence thresholds for autonomous decisions
- Human-in-the-loop triggers for high-stakes actions
- Graceful degradation when uncertainty is high

Orchestration isn't about using every model. It's about using the *right* model for each task, automatically, while staying within operational constraints.

Augmented Intelligence: AI Proposes, Humans Decide

The hardest problems aren't technical. They're organizational.

Enterprise-grade agentic systems embed into existing workflows:

- AI generates recommendations in tools teams already use
- Purpose-built review interfaces for decision approval
- Clear handoff protocols (when humans override agents)
- Feedback loops that improve future recommendations

"Augmented AI" means the agent works *for* the human, not instead of them. Especially early in deployment, humans should approve high-stakes decisions. As trust builds and evals prove reliability, you expand the autonomous envelope.

Outcomes must activate in production systems. A recommendation in a dashboard is a memo. A recommendation that updates Salesforce, triggers an email sequence, and schedules follow-up is work done.

Built-With, Not Built-On

Here's the uncomfortable truth: most "enterprise AI platforms" are configuration layers over generic models.

True enterprise-grade systems are purpose-built for specific domains while leveraging platform infrastructure:

- Custom data pipelines for your integration requirements
- Domain-specific context graphs (not generic knowledge bases)
- Evals trained on your historical data
- Workflows that match your organizational processes

You get platform benefits—security hardening, regulatory compliance, operational monitoring—without platform constraints. The system should feel purpose-built because it *is*, while benefiting from lessons learned across dozens of enterprise deployments.

Enterprise-grade agentic AI is not defined by model intelligence. It is defined by operational rigor. Coding up an agent that executes and successfully completes a task once is easy. To get an agent to do so repeatedly, within policy constraints, with full auditability, bounded risk, and at scale, is hard.

When agentic systems fail in production, the failures are rarely due to the model. Yes, sometimes you have to retry the LLM call. But more often than not, the failures are due to missing context, weak governance, shallow evaluation, or fragile integrations. Enterprise systems assume regulated data, real financial impact, and organizational complexity. They require continuous evaluation and controlled, accountable autonomy.