

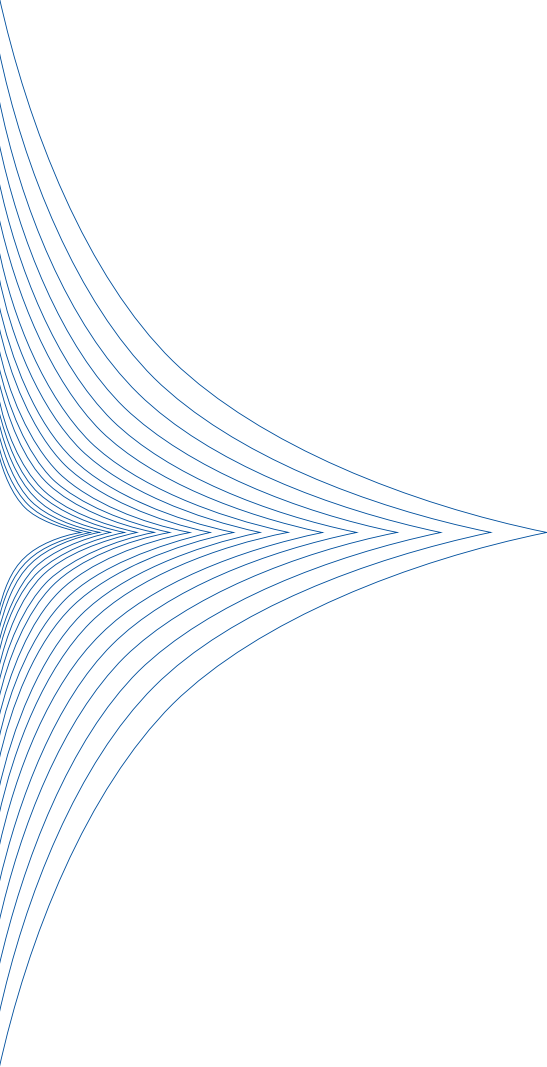


WHITE PAPER

Building Enterprise-Grade Agentic AI Systems

A control-stack approach for safe autonomy, auditable action, and operational scale





Executive overview

AI agents become relevant to enterprise strategy when they can plan across steps, consult internal knowledge, select skills at runtime, and execute actions through approved tools. And perform all these tasks while working with the uncertainty inherent in AI systems.

This means failure modes change. A wrong response can be corrected. A wrong action executes successfully while creating a downstream incident. Example: An agent approves a refund outside policy limits, or updates a customer record with stale data from yesterday's batch run.

That is why "production-grade" is not something to add after a pilot. You must engineer it into the system through explicit governance mechanisms. These controls cover data authority, orchestration discipline, safe tool execution, security boundaries, and day-two operations.

Contents

Executive overview	2
Market trends and why enterprises are moving to autonomous AI	4
Quick orientation for business leaders	4
Where agentic AI is being applied first, by function and industry	5
Why agents create new risk (and what fails first)	5
Common pilot entry points inside organizations	6
The six layer control stack for production readiness	7
Our control stack consists of six layers:	7
Layer A: Data and knowledge foundation	7
Layer B: Agent orchestration and runtime	9
Layer C: Intelligence and decision quality	10
Layer D: Integration and enterprise readiness	11
Layer E: Security, compliance, and trust	13
What changes for regulated decision and transaction paths	14
Layer F: Operations, monitoring, and scale	14
A practical view of technology stack options	15
How our implementation approach reduces risk	15
25 critical areas to assess your autonomous AI readiness	16
Leveraging AI agents in your organization	17
QKTech capabilities	17
Ready to assess your AI readiness?	18
About QKTech	19

Market trends and why enterprises are moving to autonomous AI

Enterprise interest in AI agents is rising because organizations are moving from copilots that assist single tasks to systems that complete multi-step work across tools and data—where scale value appears.

Gartner estimates that by 2028, 33% of enterprise software applications will include agentic AI, up from less than 1% in 2024, and that 15% of day-to-day work decisions will be made autonomously through these systems.

GenAI adoption is also broadening across functions. McKinsey reports 65% of respondents say their organizations are regularly using genAI, with frequent use across marketing and sales, product and service development, and IT. If you are not piloting agents now, you are 18 months behind your competitors.

Quick orientation for business leaders

Here is how to translate intent to architects:

What leaders expect from agents	What the platform must guarantee in production
Faster resolution and measurable productivity lift	Authoritative data foundations, controlled independence, and safe tool execution that remains stable under variance
Scale across channels and functions	Orchestration with routing, state, validation gates, and replay-safe actions
Broader adoption without governance friction	Role-aware controls, traceability, continuous evaluation, and controlled rollouts

Where agentic AI is being applied first, by function and industry

Early adoption concentrates in functions with high-volume, repeatable knowledge work and clear outcome measures: customer operations, IT service management, engineering enablement, and enterprise knowledge workflows.

Across industries, regulated sectors move deliberately, investing early in stronger governance because auditability, access boundaries, and evidence traceability are non-negotiable.

Why agents create new risk (and what fails first)

Conventional enterprise software is predictable because its control flow is deterministic and its contracts are explicit. Traditional ML systems often remain advisory because a separate layer decides whether actions are executed. Autonomous agents change both assumptions. They combine probabilistic reasoning, persistent state across interactions, and tool invocation against live systems of record. They also hallucinate, generating plausible but false information when evidence is weak or context is ambiguous.

This creates a new class of failure. The system appears healthy and behaves incorrectly in business terms, because a tool call succeeds technically while being wrong in context, entitlement scope, or current data. When this happens, the organization discovers it cannot reconstruct what evidence the agent used, which policy version was applied, and why the system believed it was safe to proceed.

At QKTech, we start by treating autonomy as a supervised runtime mode that must be engineered, monitored, and reversible.

The risks to plan for are consistent across enterprises: Unclear data authority, stale knowledge, and overbroad tool permissions produce plausible but incorrect actions. Weak state management compounds errors across retries. Missing trace logs make incident reconstruction slow and legally risky. Cost runaway happens when reasoning loops are not budgeted or monitored, as a single agent loop can burn \$50-500 in API calls.

Common pilot entry points inside organizations

Enterprises most often start with pilot projects that are measurable, bounded, and integration-light at the beginning, such as policy and knowledge search; summarization and drafting; contact center quality workflows and routing; and internal analytics assistance. These pilots then mature into transactional workflows once tool governance, approvals, and traceability are established.

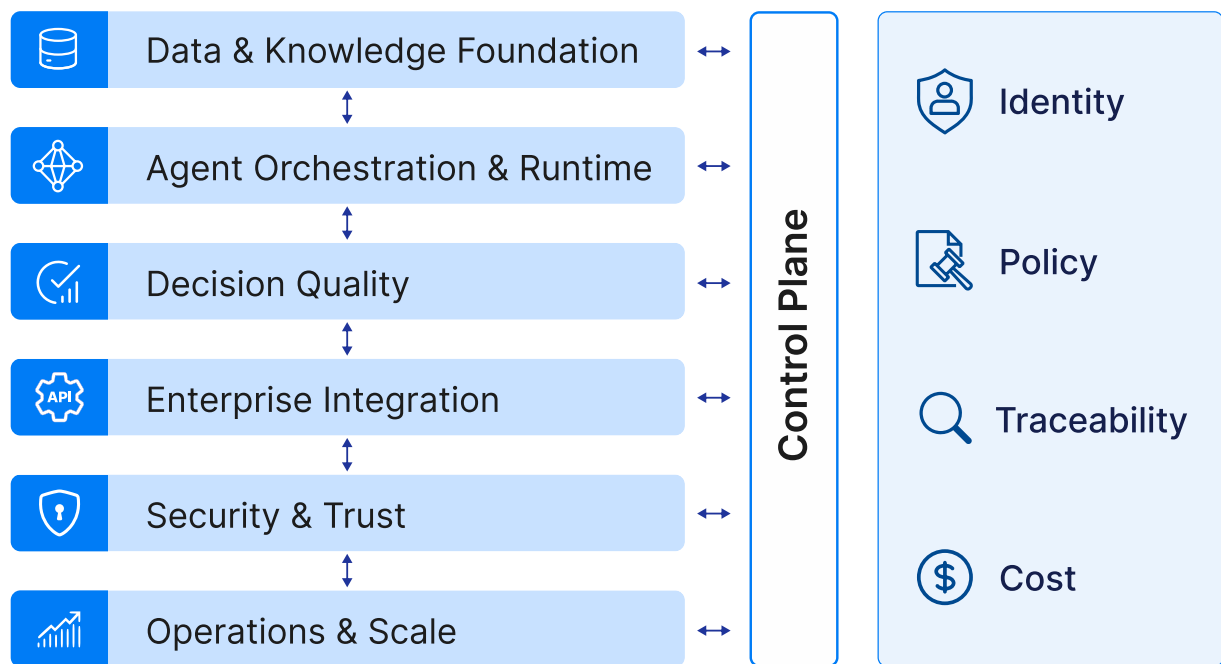


Figure 1: The Enterprise Agentic AI Control Stack: Six layers from data foundations to operational monitoring

The six layer control stack for production readiness

Our control stack consists of six layers:

Layer A: Data and knowledge foundation

Enterprise agents need more than access to data. They need authority, freshness, and traceability.

→ Golden record creation

- Production agents need a single authoritative view of critical entities, whether the entity is a customer, vendor, asset, or ticket. You must know how that "single view" gets created. Some enterprises centralize it. Others federate it across systems. Either way, the agent must consume a consistent representation built from defined resolution rules. Otherwise, the model will attempt to rationalize conflicts and will do so differently across prompts and contexts.
- For example, applying it for a telecom company means one customer ID that reconciles billing, support tickets, and device history; and not three conflicting versions across CRM, billing, and network systems.

→ Stale data detection and refresh strategy

- In production, facts decay. You must engineer for this. TTLs, re-fetch policies, and confidence decay on retrieved facts prevent agents from acting on information that was correct at the start of a workflow but became wrong before execution. This is critical in every industry, and it becomes non-negotiable in regulated workflows where "current at time of action" is part of defensibility.

→ Multi-source data federation

- Most enterprise work blends real-time and batch signals. A service workflow may need live ticket state plus policy documents, while a finance workflow might require ledger-consistent snapshots plus exceptions from an operational system. An enterprise-ready platform supports real-time and batch access across ERP, CRM, data lakes, and APIs. It also annotates the latency characteristics, so the system does not silently treat a batch extract as live truth.

→ Contextual data injection engine

- Agents should not guess business context. Inject context consistently: user role, entitlements, interaction history, system state, and workflow stage. When implemented properly, this reduces ambiguity and lowers security risk, because the system isn't asking the model to infer boundaries it should never infer.

→ RAG with source traceability

- Treat retrieval as evidence, not influence. Page-level citations, document lineage, and version control turn RAG outputs into auditable artifacts. If a decision is questioned later, the platform must be able to show which document version was used and where the supporting statement came from.

→ Knowledge freshness governance

- Knowledge bases drift. Policies update. Standards change. Scheduled re-indexing, semantic drift detection, and lifecycle rules make "knowledge" a governed asset, rather than an ever-growing pile of documents that quietly changes system behavior.

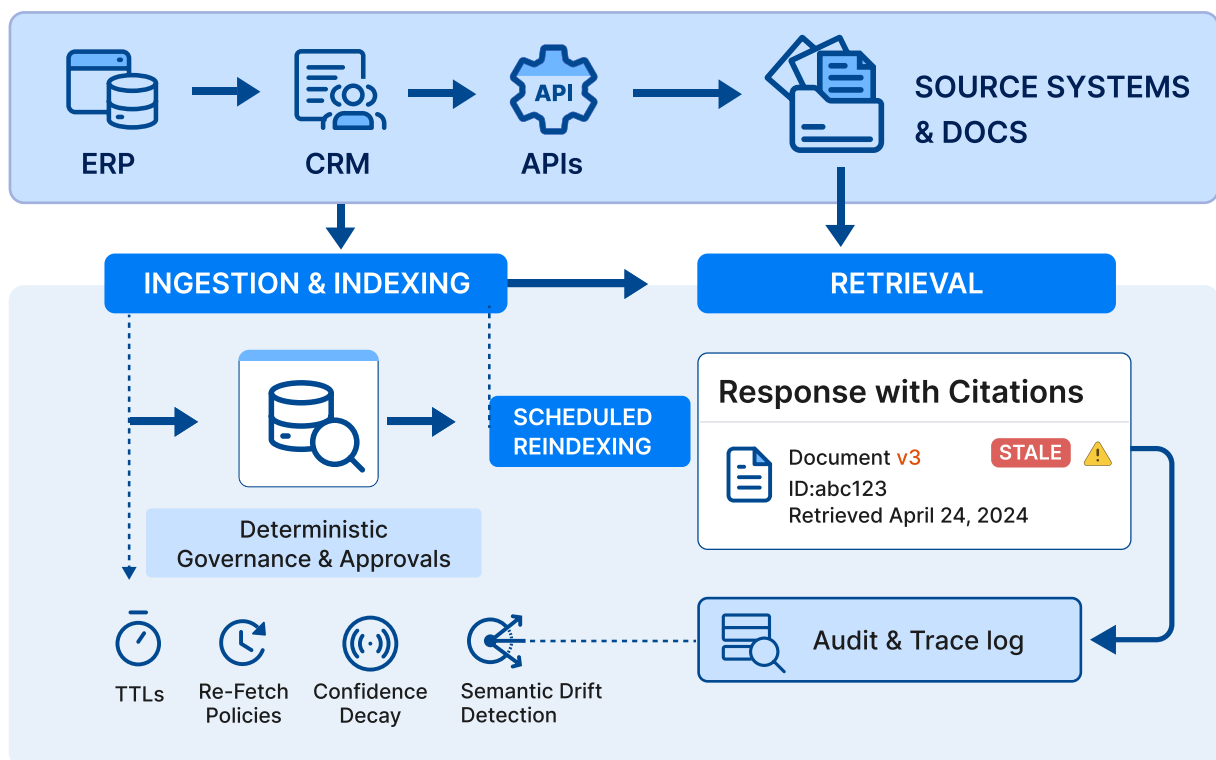


Figure 2: Knowledge lineage and freshness

Layer B: Agent orchestration and runtime

Orchestration determines whether autonomy stays controlled and safe, when requests become ambiguous, workloads spike, or downstream services fail partially.

→ **Multi-agent collaboration framework**

- Stable implementations separate responsibilities. A planner decomposes intent into steps. An executor performs defined actions. A critic or validator checks outcomes and triggers fallback before side effects are committed. This division improves reliability, and it also makes failures diagnosable.

→ **Conversation and task state management**

- Enterprise workflows span time, retries, and escalations. Conversation history helps. Task state is mandatory, and it should persist across sessions, support resumption, and remain structured enough to avoid memory becoming a dumping ground that quietly increases risk.

→ **Agent routing and skill selection engine**

- Routing should be dynamic and policy-aware. The system selects agents based on intent, confidence, and cost. It also respects the risk tier of the workflow, so high-impact actions are routed through stricter paths.

→ **Deterministic to autonomous control switch**

- The platform needs an explicit control switch between autonomous execution and deterministic constraints. When risk rises, the system shifts toward rules, validations, and approvals. When risk is low and bounded, autonomy can be wider without being reckless.

→ **Tool invocation and function governance**

- Tool calls are where incidents are born. Governance requires whitelisting, schema validation for inputs and outputs, retry and timeout behavior that avoids duplication, and logging that supports replay protection. If you cannot safely retry a tool call, do not use it in an autonomous loop.

"Agentic AI delivers durable value when it can move from intent to action without turning uncertainty into operational risk; and when its decisions can be reconstructed with the same discipline expected of any enterprise system that touches customers, money, or compliance."

Layer C: Intelligence and decision quality

Decision quality is what turns model capability into enterprise trust.

→ **Confidence scoring and answer reliability**

- Agents must self-score and expose uncertainty. Confidence then becomes a runtime input. It decides whether the system proceeds, escalates, or refuses to act when evidence is weak.

→ **Ranking and prioritization engine**

- When several actions are plausible, the system should rank them, expose rationale, and follow a consistent escalation path. This improves operational predictability in triage environments.

→ **Fallback and graceful degradation logic**

- Failures will occur, but the question is whether they degrade safely. A deployment-worthy platform switches to alternate agents, deterministic rules, or human escalation when confidence drops, tools fail, or evidence is stale. Design and test these paths. Do not improvise them in production.

→ **Cost-aware reasoning controls**

- Cost management is part of operational control. Token budgets, model switching by task criticality, and inference throttling protect the system from runaway loops and enable FinOps discipline for GenAI workloads.

Case study: Emails operations agent

→ **Industry:**

- High-volume customer support

→ **Volume:**

- 50,000+ emails/month

→ **Challenge:**

- Rigid RPA-based classification could not handle variance. QKTech replaced the existing system with a custom-trained AI model that classifies email inflow across multiple categories, supported by audit and feedback loops and on-prem deployment.

Results:

- 70% reduction in classification costs
- 20% reduction in time to first response
- Real-time anomaly detection for misrouted emails.

“If your agent cannot name the authoritative source for an attribute, it should not be allowed to execute an update based on that attribute.”

Layer D: Integration and enterprise readiness

Agents create value when they transact, and transactional capability must be safe by design.

→ **API-first transaction enablement**

- Agents should not remain “chat only.” Enterprise-ready agents create, update, and validate through APIs with documented contracts. APIs provide enforceable schemas, consistent audit trails, and predictable error handling

→ **Idempotent and auditable actions**

- Safe retries prevent duplicate side effects. Replay protection avoids accidental re-execution. Transaction logs support investigations, audits, and controlled rollbacks when business context changes.

→ **Event-driven architecture compatibility**

- Enterprises run on events. Agents should react to events, not only prompts, for operations and compliance use cases. When agents react to events, idempotency and traceability become even more important, because speed increases blast radius when controls are weak.

What happens when an API is down? Implement circuit breakers that prevent retry storms, fallback to read-only mode, and queue actions for replay once connectivity is restored. Never let an agent loop indefinitely against a failed endpoint.

Case Study: Trade compliance reconciliation

Industry:

- Financial services

Challenge:

- Manual audio-to-tradebook reconciliation for phone-based orders We automated the complete workflow: transcription, structured extraction, matching, discrepancy flagging, and audit-ready evidence generation.

Results:

- 100% audit coverage with traceable digital logs
- 95% reduction in manual effort
- Near real-time alerts for mismatches
- Zero compliance violations post-deployment

Case study: AI-driven contact center transformation

Client:

- Ather Energy, a connected electric mobility company

Challenge:

- High volumes of incoming support calls, manual call disposition that consumed significant agent time, produced uneven quality, and left valuable customer insights under-captured

We implemented an automated conversation summarization workflow using Google's GenAI and Text-to-Speech capabilities. The solution captured key discussion points, customer sentiment, and actionable items. These insights were then automatically fed to Salesforce records as structured, standardized summaries.

Results:

- 90% reduction in call disposition time
- 80% reduction in mis-tagged calls
- 100,000+ agent hours saved
- Improved customer satisfaction scores through faster follow-up

Layer E: Security, compliance, and trust

In production, security is an enforced boundary before and after inference.

→ **Role-aware authorization, RBAC and ABAC**

- Agents must respect enterprise access boundaries. Authorization checks apply to retrieval and to tool execution. Do not rely on natural language context as a permission mechanism.

→ **Prompt, data, and response guardrails**

- Guardrails exist on both sides of model calls. Inputs and retrieved context are filtered and constrained. Outputs are checked for policy violations and data leakage, before any transactional step is allowed to proceed.

→ **Human-in-the-loop controls**

- Approvals are a design choice, not a maturity gap. High-risk actions should require documented approval, when the system touches finance, compliance, customer commitments, or regulated decision support.

→ **Explainability and decision trace logs**

- Trace logs must reconstruct the complete chain. Capture: context injection, retrieval evidence with versions, policy checks, confidence score, tool call parameters, approvals, and validation outcomes. Without this, audit readiness remains a promise rather than a capability.

→ **Data residency and cross-border compliance**

- For enterprises operating in EU and Asia: Ensure agent data flows respect GDPR, data localization laws, and regional AI regulations. Use regional model endpoints, segregate training data by jurisdiction, and maintain separate audit trails for cross-border transactions.

Case study: Secure enterprise adoption at scale

Client:

- Mahindra and Mahindra GenAI platform

Challenge:

- Enable broad access while protecting sensitive data across a diverse workforce

We implemented robust IAM controls while maintaining an intuitive interface accessible to employees with varied technical backgrounds and authorization requirements.

Results:

- Role-based access for 10,000+ users
- Zero security incidents post-launch
- 30% productivity increase in knowledge-intensive workflows

What changes for regulated decision and transaction paths

In BFSI, autonomous systems must treat identity, authorization, evidence traceability, and approvals as first-class runtime controls. An incorrect action creates audit exposure and customer harm. The EU's AI Act reinforces a risk-based framing for AI governance and highlights why understanding and oversight matter, which maps directly to agent traceability and human override controls for consequential workflows.

Global regulators such as India SEBI and RBI, UK FCA, US OCC, EU EBA, among others, are actively monitoring the risks of AI agents through human-in-the-loop oversight, AI governance, and validation of data from multiple systems.

Layer F: Operations, monitoring, and scale

Autonomous systems change over time because their inputs change, and because prompts, tools, and models evolve. Operational discipline determines whether that change stays controlled.

→ **Agent observability and telemetry**

- Telemetry should capture latency, failure rate, cost per task, and quality signals that indicate groundedness and unsafe behavior. It becomes operationally useful when correlated to the precise agent, prompt, tool, and model version.
- Our PHI solution reflects this with as our operational inputs: usage analytics, token and cost consumption, logs, and feedback loops to update context or model behavior.

→ **Continuous evaluation and regression testing**

- Run evaluation continuously. Golden test sets and deterministic scenarios detect regressions when knowledge bases refresh or prompts change. This is how the system stays stable while improving.

→ **Versioning and safe rollouts**

- Version everything that shapes behavior: agents, prompts, tools, and models. Roll out through staged deployments with rollback. Production systems treat this release discipline as a risk control mechanism.

A practical view of technology stack options

Most operationally mature autonomous systems combine a model layer, an orchestration layer, a retrieval layer, a tool governance layer, an identity and policy layer, and an observability and evaluation layer.

Implementation choices vary, but the control surfaces remain consistent. Reference examples for orchestration and managed agents include OpenAI's practical agent guide and managed agent patterns on major cloud platforms.

How our implementation approach reduces risk

We start by scoping an initial workflow with documented action boundaries and success metrics. Then engineer the control stack early: data authority, tool governance, traceability, approvals, and evaluation. Run pilots while monitoring telemetry data and regression tests, then expand through versioned rollouts with fast rollback.

25 critical areas to assess your autonomous AI readiness

The table below maps the relation between architecture validation and the business requirements:

Domain	Details covered
Data and knowledge foundation	1. Golden record creation 2. Stale data detection and refresh strategy 3. Multi-source data federation 4. Contextual data injection engine 5. RAG with source traceability 6. Knowledge freshness governance
Data and knowledge foundation	7. Multi-agent collaboration framework 8. Conversation and task state management 9. Agent routing and skill selection engine 10. Deterministic to autonomous control switch 11. Tool invocation and function governance
Intelligence and decision quality	12. Confidence scoring and answer reliability 13. Ranking and prioritization engine 14. Fallback and graceful degradation logic 15. Cost-aware reasoning controls
Integration and enterprise readiness	16. API-first transaction enablement 17. Idempotent and auditable actions 18. Event-driven architecture compatibility
Security, compliance, and trust	19. Role-aware authorization 20. Prompt, data, and response guardrails 21. Human-in-the-loop controls 22. Explainability and decision trace logs
Operations, monitoring, and scale	23. Agent observability and telemetry 24. Continuous evaluation and regression testing 25. Versioning and safe rollouts

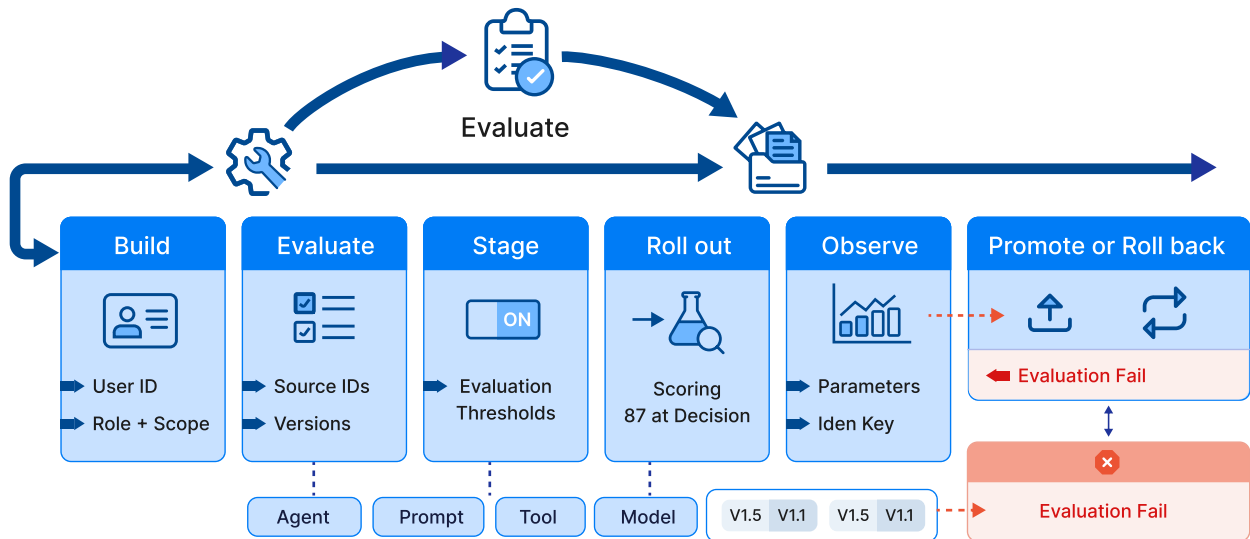


Figure 3: Release and evaluation control loop: Continuous feedback from telemetry to prompt refinement and model selection

Leveraging AI agents in your organization

Enterprise-grade autonomous AI is achieved when autonomy is controlled, when data authority is documented, and when every action can be reconstructed with evidence, approvals, and versioned artifacts. This discipline is what allows leaders to expand adoption confidently while giving architects a platform that remains operable as workflows evolve.

Download the extended implementation guide: [“Agentic AI Enterprise Checklist: From Blueprint to Production”](#) and use it to benchmark readiness across architecture, security, and operations.

QKTech capabilities

We help enterprises design and implement autonomous systems with:

- **Governed data and retrieval**
- **Orchestrated multi-agent runtimes**
- **Safe tool integration**
- **Role-aware controls**
- **Traceable decision logs**
- **Continuous evaluation**
- **Controlled rollouts**

Ready to assess your AI readiness?

Start with our 25-point checklist, then book a 30-minute architecture review with our team. We will map your autonomy boundaries, validation gates, and telemetry requirements to your existing systems.

We'll help you:

- Define autonomy boundaries for your first workflow
- Design validation gates that match your risk profile
- Establish telemetry requirements aligned with your architecture standards
- Plan rollout strategy using proven patterns from large-scale platforms and regulated workflows

About QKTech

QKTech, formerly QualityKiosk Technologies, helps businesses win in the AI era by engineering reliability. Backed by experience in delivering reliable software, products, and platforms for leading global enterprises, QKTech combines deep domain knowledge with strong technology skills to help customers deliver great experiences to their own customers in an AI-first world.

Founded in 2000, QKTech offers reliability, cloud, AI, and product engineering services that help organizations build and adopt AI effectively, spanning product development, quality and reliability engineering, digital assurance, CloudOps and automation. At the core of this is the AI Reliability OS, an integrated portfolio of services covering AI for Reliability, Reliability of AI, Shift Right testing, Reliability of Frontier Systems, and Agentic Engineering. It brings together context engineering, harness engineering, loop engineering, observability, and platform-based delivery.

Operating in more than 25 countries with offices in Princeton and San Jose in the US, as well as Dubai, Singapore, Manila, Hong Kong and Jakarta, QKTech helps leading brands in banking, capital markets, insurance, automotive & manufacturing, technology & digital natives, telecom, airlines, consumer goods, and healthcare to transform their digital capabilities. The company is headquartered in Mumbai, India, with engineering centers across Bengaluru, Hyderabad and Pune.

The company's expertise and achievements have garnered recognition from esteemed global analyst firms such as Gartner, Forrester, and The Everest Group for innovative, IP-led solutions and the significant value of those solutions delivered to clients. QKTech has partnered with the most innovative and disruptive technology companies to force multiply outcomes that can be delivered to the customers.

For more information, visit www.qktech.com