



Changing Your AI Model: The Risks Institutions Underestimate

Can institutions replace an AI model without
surrendering control of the platform around it?

For asset managers and asset owners

Model migration, institutional knowledge, and control of the AI platform

Author Christopher Thomas (C.T.) Rusert

Date October 2026

Published by ZTA Labs

CONTROL. GOVERN. BUILD. **EVOLVE.**



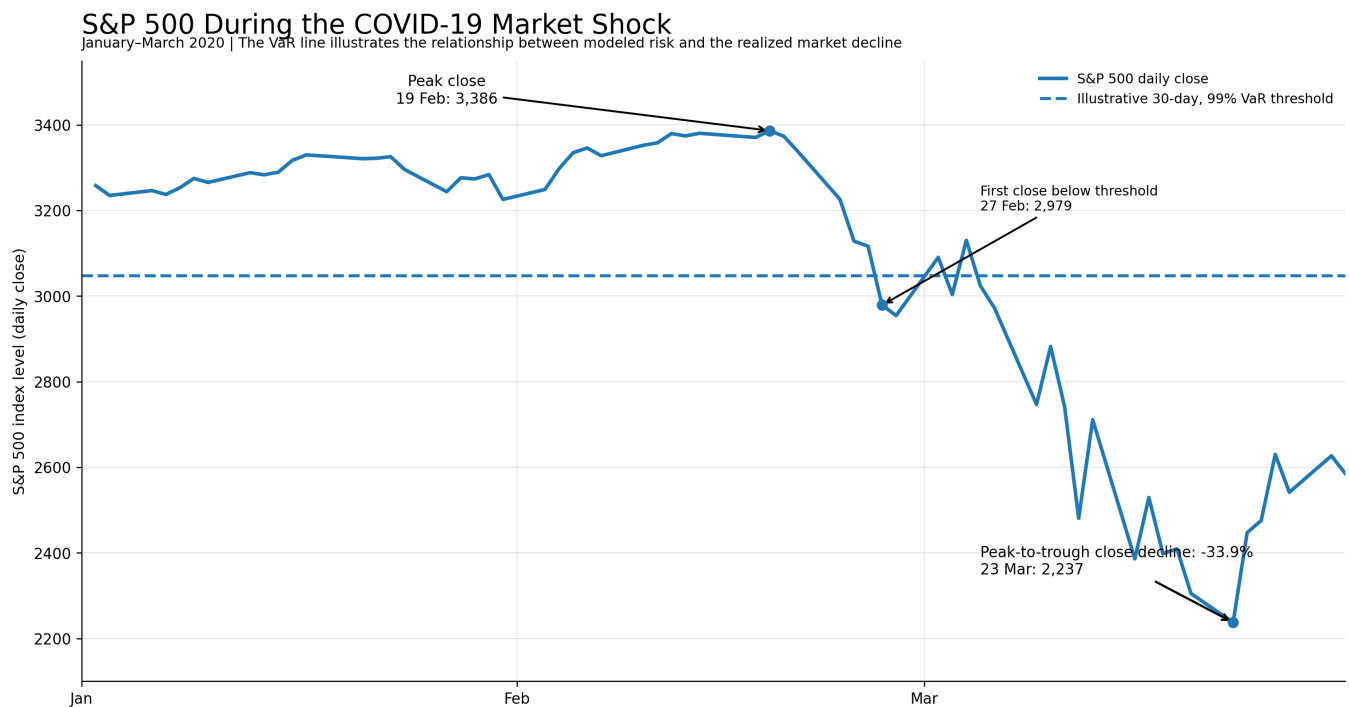
ZTA LABS | RESEARCH & EDITORIAL | OCTOBER 2026

OPENING PERSPECTIVE

When the Model No Longer Fits the Market

In March 2020, amid the market dislocations triggered by the COVID-19 pandemic, asset managers faced a critical question: did their risk models accurately reflect portfolio exposures?

The S&P 500 fell approximately 34% from its February peak to its March 23 low. Volatility surged, asset correlations shifted sharply, and assumptions derived from historical market behavior came under intense scrutiny. For firms relying on historical Value at Risk (VaR), the crisis exposed the methodological vulnerability of using past observations to estimate risk during an unprecedented market dislocation.



Historical market data: S&P 500 daily closing levels, January–March 2020. The dashed line illustrates a 30-day, 99% VaR threshold. Actual VaR estimates varied by portfolio, model configuration, and assumptions.

Figure 1. S&P 500 daily closing levels, January–March 2020. The dashed line illustrates a 30-day, 99% VaR threshold to show the relationship between modeled risk and the realized market decline. Actual VaR estimates varied by portfolio, model configuration, and assumptions.

This vulnerability is acutely pronounced in historical simulation frameworks that rely on rigid 250- or 500-day lookback windows. Prior to the onset of the COVID-19 contraction, these discrete lookback windows lacked any comparable sequence of extreme tail risk, volatility spikes, or systemic correlation breakdowns. Consequently, relying on these unadjusted historical observations understated portfolio exposure to a market regime entirely unrepresented within the sample baseline. While trailing observations eventually entered the model as the crisis intensified, the resulting risk estimates severely lagged the real-time deterioration of market conditions. The systemic issue was not an inherent lack of utility in historical modeling, but rather that its predictive capacity was structurally constrained by the finite parameters of the active sample window.

In contrast, Monte Carlo VaR provides risk managers with an alternative framework for evaluating portfolio exposures. Rather than relying exclusively on the sequential replay of observed market movements, Monte Carlo methodologies generate thousands of hypothetical market trajectories derived from specified statistical distributions and systemic risk-factor relationships. This capability enables risk teams to simulate stress scenarios well beyond



the constraints of the empirical sample—including elevated volatility regimes, fat-tailed return distributions, stressed correlations, and non-linear portfolio sensitivities. The efficacy of this approach, however, remains fundamentally contingent upon the calibration of model assumptions and scenario design. Monte Carlo simulations do not intrinsically anticipate systemic shocks or capture fat-tail events; those characteristics must be deliberately engineered into the underlying distributions, parameter calibrations, or prescribed stress parameters. Ultimately, while the methodology broadens the spectrum of quantifiable risks, it shifts rather than eliminates structural model risk.

This operational distinction was directly relevant to portfolio-risk practice during the 2020 crisis. Because Monte Carlo engines provided the structural capacity to simulate unobserved market shocks—even if those shocks required deliberate parameterization—asset managers using Bloomberg PORT were advised to consider replacing historical VaR with Monte Carlo frameworks where the alternative better addressed their portfolio characteristics and risk-analysis requirements. The decision involved more than substituting one mathematical calculation for another. It required assessing the empirical boundaries of the existing methodology, the user-defined assumptions underlying the alternative, the explicit scenarios each approach could represent, and the resulting implications for institutional risk governance. The experience illustrates a definitive principle that extends far beyond investment risk: replacing a core analytical model is fundamentally a decision about the broader operational system that relies on it.

Although investment risk engines and large language models (LLMs) serve different purposes, the governance discipline required to transition between them follows a comparable principle. Asset managers are embedding LLMs into investment research, quantitative analysis, and operational workflows. Replacing an underlying LLM can alter how evidence is synthesized, how tools are invoked, and how structured outputs feed enterprise applications. An LLM that achieves superior results on general-purpose LLM benchmarks may still introduce material failures within an institution's specific workflow.

The pace of AI development compounds the challenge. LLMs improve each quarter, creating opportunities to increase analytical performance, reduce latency, and improve cost efficiency. Remaining tied to a legacy LLM because of integration friction can prevent an institution from realizing those gains and create technical debt. Conversely, migrating without evidence of a meaningful improvement introduces operational risk and expense without a demonstrated return. LLM replacement should therefore be driven by rigorous comparative testing against institutional workflows, with integration dependencies, governance requirements, and operating costs included in the assessment.

This paper presents a five-step framework for managing LLM transitions while preserving institutional control over the technology stack. It also examines the longer-term risk of becoming so dependent on a particular LLM or ecosystem that replacement becomes impractical.

1. Structural Disruption in Working Systems

Replacing an embedded LLM changes the behavior of the system that depends on it and can expose hidden dependencies after deployment. In investment research, an LLM replacement may achieve higher aggregate scores on standard evaluation suites while degrading source-citation accuracy, breaking downstream API schemas, or misinterpreting contradictory qualitative evidence. The overall benchmark result improves, but the production workflow no longer satisfies institutional requirements. LLM performance must therefore be evaluated in the context of the operational ecosystem in which the LLM will run.

These risks extend directly into investment research and portfolio management. An LLM used to monitor credit positions may be required to identify changes in an issuer's financial condition, reconcile new disclosures with the existing investment thesis, and flag developments that warrant a portfolio manager's attention. A replacement LLM may perform better on general-purpose LLM benchmarks yet omit a material covenant qualification, misattribute a statement to the wrong source, or fail to distinguish a confirmed deterioration from an emerging risk. Each failure can undermine the research workflow, even if the replacement produces stronger results across the broader



evaluation suite. Qualification must therefore establish that the LLM meets the institution's specific evidentiary, analytical, and escalation requirements.

Four dimensions of risk require institutional mitigation:

- **Performance divergence.** Higher aggregate LLM benchmark scores can conceal regressions in critical tasks, including structured data extraction, calculation accuracy, evidence attribution, and appropriate abstention when the available evidence is insufficient.
- **Integration incompatibility.** Differences in tokenization, tool-calling syntax, context-window utilization, and output formats can require application changes, code refactoring, and additional orchestration testing.
- **Governance and control failure.** Changes in LLM behavior can expose weaknesses in data-entitlement enforcement, compliance boundaries, tool permissions, or mandatory human-approval gates.
- **Transition risk.** Pre-deployment testing may not reproduce production edge cases. Inadequate monitoring, unclear release authority, and insufficient rollback procedures can turn a manageable defect into an operational incident.

The degree of assurance should reflect the consequences of failure. An LLM used to summarize documents presents a different risk profile from an agent connected to systems that can initiate consequential actions. In either case, the institution must identify which dependencies can change, which controls must remain effective, and what evidence is required before the replacement is approved.

2. Five Steps to Manage an LLM Change

Step 1: Establish the Dependency Baseline

Before evaluating replacement LLMs, document how the existing LLM is used and which components depend on its behavior. The inventory should cover LLM versions and configurations, inference parameters, prompts, orchestration, retrieval pipelines, data connectors, tool interfaces, output schemas, evaluation methods, monitoring, licensing, and infrastructure requirements. The objective is to distinguish LLM-specific dependencies from components the institution controls independently. An application that relies on a particular LLM's ability to return structured output, for example, may require substantial engineering work if that assumption is embedded across multiple applications.

The assessment should establish which components can remain unchanged, which integrations and workflows require modification, which permissions and controls need revalidation, and which licensing, hardware, or infrastructure constraints affect candidate selection. It should also identify the operational procedures required for deployment and rollback. This baseline provides a more realistic estimate of migration effort and exposes dependencies that may constrain subsequent LLM changes.

Step 2: Decouple Institutional Knowledge from the LLM

Institutional knowledge should remain available when the LLM that uses it is replaced. Investment firms accumulate knowledge through research, analyst corrections, documented investment theses, decisions, and subsequent outcomes. Some resides in approved research and structured records; other knowledge develops through repeated work, including corrections to recurring analytical errors and lessons from previous investment decisions. When this knowledge is embedded primarily in LLM-specific fine-tuning, prompts, or proprietary memory mechanisms, replacing the LLM can require the institution to reconstruct what it has already learned.

ZTA Labs' approach is to maintain an institution-controlled Knowledge Platform independently of the LLMs that use it. Validated knowledge can be retrieved through retrieval-augmented generation (RAG) or incorporated into curated fine-tuning datasets when appropriate. These mechanisms serve different purposes: RAG makes relevant knowledge available during inference, while fine-tuning can adapt LLM behavior or improve performance on defined tasks. Neither replaces the need to maintain authoritative institutional knowledge outside the LLM.

The Knowledge Platform must also manage how knowledge changes. New evidence may supersede an earlier fact, qualify an existing conclusion, or have little bearing on established knowledge. Recency alone is insufficient to



determine which information should influence current analysis. A governed Knowledge Flywheel provides a mechanism for managing this process:

- **Capture:** Retain relevant evidence, analyst corrections, decisions, and subsequent outcomes.
- **Validate:** Assess provenance, reliability, corroboration, and relevance before promoting information into approved institutional knowledge.
- **Maintain:** Update conclusions, resolve contradictions, mark superseded information, and reduce the operational weight of knowledge that has become stale.
- **Reuse:** Make validated knowledge available to research workflows through retrieval and, where appropriate, curated fine-tuning.
- **Learn:** Evaluate how that knowledge contributes to subsequent analysis and decisions, using validated findings to improve the platform.

The cycle is continuous. Its purpose is to improve the quality and relevance of institutional knowledge, not simply to accumulate more information. Historical knowledge may remain valuable even after it loses relevance to current decisions; the platform should preserve that distinction rather than erase older information whenever a new conclusion is recorded. For LLM replacement, the institution can connect a new LLM to its existing knowledge assets while still testing whether the replacement retrieves, interprets, and applies that knowledge appropriately.

Step 3: Preserve Control Over Permissions and Consequential Actions

The institution's authority model must remain consistent when the underlying LLM changes. A research agent may be permitted to retrieve approved evidence, prepare a recommendation, and assemble a proposed order while requiring human authorization before the order can be submitted. That boundary must be enforced by the system regardless of which LLM performs the analysis. A replacement may interpret instructions differently or interact with tools in unexpected ways; better analytical performance does not establish that it will respect the same operational constraints.

ZTA Labs' AI Control Plane is designed to enforce institutional policy independently of LLM behavior. Its responsibilities include identity and entitlement checks, bounded access to data and tools, action-specific permissions, mandatory human-authorization gates, and records of consequential events. An LLM change should trigger verification that data access remains consistent with user and workflow entitlements, tool permissions remain within the approved scope, required human approvals cannot be bypassed, and expired or revoked authorization is rejected. Authorization failures must not silently grant permission, and denied actions and consequential events must remain auditable.

These controls must be enforced at the interfaces through which actions are performed. Instructions telling an LLM to respect a policy are useful, but they do not replace an enforceable access-control mechanism. The Knowledge Platform and AI Control Plane serve different purposes: the former maintains institutional knowledge, while the latter governs access and action. Keeping these responsibilities separate reduces the risk that an LLM change also changes the institution's knowledge-management or authorization arrangements.

NIST's AI Risk Management Framework provides a useful foundation for this lifecycle approach through its four functions: Govern, Map, Measure, and Manage. It supports the need to identify risks, establish accountability, evaluate system behavior, and manage changes. It does not prescribe this particular architecture or a specific LLM-migration procedure.

Step 4: Qualify the Replacement Against Institutional Requirements

The replacement decision should be based on evidence from the institution's own workflows. General-purpose LLM benchmarks help identify promising candidates, but they cannot establish that an LLM meets the requirements of a particular investment process. A firm may need to evaluate source attribution, calculation accuracy, treatment of conflicting evidence, adherence to a defined analytical framework, and the ability to recognize when evidence is insufficient. A qualification suite should use representative institutional tasks, validated answer keys, source requirements, and known failure cases, with acceptance criteria defined before testing begins.



The evaluation should address four dimensions:

- **Analytical performance:** Accuracy, calculation quality, evidence use, and appropriate handling of uncertainty.
- **Workflow compatibility:** Compliance with output schemas, tool interfaces, and downstream application requirements.
- **Governance:** Entitlement enforcement, restricted access, authorization gates, and correct escalation.
- **Operating requirements:** Latency, resource consumption, cost, licensing, and deployment constraints.

Testing should cover the complete configuration, including the LLM version, inference settings, prompts, retrieval, orchestration, and tools. Results from an isolated LLM test do not necessarily predict behavior after integration. The replacement should be compared with the existing configuration using the same evaluation set and scoring methodology. Material regressions should be investigated individually rather than obscured by an aggregate score, and failures that violate mandatory requirements should block release.

The evaluation process should also be repeatable. Retaining test cases, acceptance criteria, results, and known limitations allows future candidates to be assessed against the same institutional baseline. ZTA Labs applies this principle through Model Assurance, evaluating LLMs and configurations against defined institutional tasks and evidence requirements rather than assuming that general LLM rankings determine production readiness.

Step 5: Govern Deployment and Ongoing Operation

Qualification establishes whether a candidate meets defined requirements under test conditions; deployment introduces additional operational considerations. Before release, the institution should specify the approvals required, the conditions that prevent deployment, the controls that need revalidation, and the criteria for restoring the previous configuration. Where practical, the replacement can run alongside the existing LLM on representative tasks, allowing the institution to compare output quality, tool behavior, latency, and resource consumption before making the new configuration authoritative. Shadow testing must not permit an unapproved configuration to initiate consequential actions.

The change record should identify the previous and replacement configurations, evaluation results, known limitations, approvals, and material modifications. Historical decision records should remain intact: generating a new answer to an old question does not reproduce the evidence and judgment that informed the original decision. After deployment, the institution should monitor performance and control behavior against the established baseline, investigate unexpected tool calls and changes in evidence quality, and use escalation procedures proportionate to the consequences of failure.

Lessons from the transition should feed back into the evaluation suite and Knowledge Platform. Validated failures can become regression tests, analyst corrections can inform future research, and documented integration problems can improve the dependency baseline. The result is a repeatable process for future changes, with clearer evidence about what works, what remains uncertain, and what needs to be tested again.

3. The Cost of Governance—and the Cost of Avoiding It

A governed LLM replacement requires engineering time, evaluation resources, integration testing, and operational oversight. The effort can be material, particularly where the existing platform has many LLM-specific dependencies or supports high-consequence workflows. Assurance should therefore be proportionate to the scope of the change and the consequences of failure. A change that leaves the LLM's role and interfaces largely unchanged may require a narrower assessment than one that introduces new tools, changes orchestration, or expands the system's authority.

The business case should compare the cost of qualification with the potential cost of an ungoverned transition: disrupted workflows, unreliable analysis, control failures, remediation, and unplanned engineering work. It should also account for the cost of remaining on an increasingly unsuitable LLM because the platform is too difficult to change. Not every replacement will justify its migration cost. A candidate may offer insufficient improvement to



warrant the work, or the existing configuration may remain the better option. A disciplined qualification process makes those decisions easier to defend. The objective is to understand the cost, control the risk, and make a deliberate decision based on the institution's requirements.

4. The Longer-Term Risk: An Architecture That Is Difficult to Change

Managing an individual LLM change is only part of the problem. Institutions also need to consider whether their architecture preserves the ability to make future changes. A firm may have access to multiple LLMs but still depend heavily on one of them because its applications rely on proprietary interfaces, its workflows are tightly coupled to a particular LLM's behavior, or its institutional knowledge is difficult to transfer. These dependencies can constrain future decisions when a provider changes pricing or licensing terms, an LLM is deprecated, another candidate offers better performance, or a different deployment arrangement better meets the institution's requirements. The firm may then face a choice between accepting an unfavorable dependency and undertaking a costly migration.

Open-weight LLMs can expand the available options, but access to model weights alone does not establish platform independence. The surrounding data architecture, integrations, orchestration, governance, and evaluation processes also need to remain under institutional control. The objective is not to eliminate all LLM-specific engineering: different LLMs have different capabilities, interfaces, hardware requirements, and licensing conditions, and a replacement may require adaptation and requalification. The objective is to prevent those differences from creating an avoidable dependency across the entire platform.

Three architectural principles are particularly important:

- **Separate institutional knowledge from LLM implementation.** Knowledge should remain accessible and governable independently of the LLM that consumes it.
- **Enforce authority outside LLM behavior.** Permissions and consequential-action controls should not depend solely on instructions or LLM compliance.
- **Standardize evaluation and qualification.** The institution should retain its own test cases, acceptance criteria, and evidence of approval.

These principles preserve the institution's ability to change LLMs while maintaining the platform's essential operating requirements.

Conclusion: Make LLM Replacement a Managed Capability

Changing an LLM introduces operational risks to performance, integrations, control behavior, and production reliability. Those risks require a structured migration process supported by institutional evaluation, explicit acceptance criteria, and ongoing monitoring. Avoiding LLM changes indefinitely introduces a different risk: dependence on an LLM or platform that may no longer serve the institution's interests.

The five steps provide a practical response: map dependencies, preserve institutional knowledge, maintain independent control of permissions, qualify replacements against real workflows, and govern deployment and monitoring. The result will not be a platform in which every LLM is interchangeable without effort. It will be a platform in which the institution understands the consequences of a change, retains control of its critical assets, and can make replacement decisions on the basis of evidence.

For ZTA Labs, this defines the operational meaning of LLM independence: an environment in which the institution retains control of the knowledge repositories, governance policies, operational workflows, and qualification protocols that make its AI systems valuable, irrespective of underlying LLM evolution. The systemic risks documented throughout this paper are not an inevitable byproduct of technological change. They emerge from deficient governance and become most acute when institutions underestimate the dependencies embedded in their AI architecture.



Individual LLMs will continue to evolve. An institution's control of its platform, however, must remain independent of any single LLM, vendor, or ecosystem.

Further Reading

NIST AI Risk Management Framework — AIRC Core Functions

The Seven Steps to AI Independence — ZTA Labs Research

The five-step framework presented in this article represents a proposed ZTA Labs methodology informed by lifecycle risk-management principles. It is not an official migration framework prescribed or endorsed by the National Institute of Standards and Technology (NIST).