

BEYOND PLATFORM CHOICE

Enterprise AI Is Not a Build-or-Buy Decision — It Is an Orchestration Strategy

Evidence from 2024 points to a selective platform model: buy commodity capability, build differentiating control and own the seams

Giovanni Leonardi

September 2024



Enterprise AI Is Not a Build-or-Buy Decision — It Is an Orchestration Strategy

Transformation · No. 327

Originally written September 2024 | Re-edited and re-rendered for the WhereBeyond Edition, 2025

“The enterprise should own the evidence and authority that make AI governable, not every component that makes AI possible.”

1. The Platform Decision That Is Really Several Decisions

An executive committee asks for a recommendation on the enterprise AI platform. The paper presents three options: build, buy or partner. Each has a five-year cost, a delivery timetable and a risk score. A preferred vendor is emerging. The organisation expects one decision to settle architecture, investment and accountability.

It will not.

The proposed “platform” includes access to foundation models, document retrieval, prompt management, security controls, evaluation, data connections, agent orchestration, user interfaces, monitoring and support. Some of these capabilities are commodities. Some contain the organisation’s most sensitive data and decisions. Some are changing too quickly for a five-year commitment to remain meaningful.

Treating all of them as one platform purchase creates a false choice. Building everything is slow and expensive. Buying everything transfers too much control to a supplier. A vague multi-vendor strategy leaves teams to integrate the seams repeatedly.

The evidence from enterprise AI programmes in 2024 points to a more disciplined prescription: orchestrate a modular platform, buy replaceable capability, build only where the organisation’s decisions or controls are distinctive, and own the layer that makes components governable as one service.

2. Why the Old Platform Logic Breaks

Traditional platform decisions seek standardisation. An organisation selects a strategic product, reduces duplication and concentrates skills. The logic works best when requirements are stable enough to compare, integration patterns are understood and switching costs can be estimated.

Enterprise AI violates all three conditions.

Model capability and price are moving rapidly. Different workloads favour different models. Retrieval quality depends on data structure and access. Agentic workflows introduce tool permissions and decision paths that no model provider governs on the organisation's behalf. Evaluation remains use-case-specific. Regulation, contractual terms and internal risk appetite continue to develop.

A single suite may simplify procurement and initial integration, but the apparent simplicity can conceal dependency. If prompts, evaluation data, model calls, retrieval logic, user feedback and monitoring are embedded in proprietary services, changing provider becomes a programme rather than a configuration decision.

The opposite response—assemble every component independently—creates its own failure. Teams choose different gateways, logging, identity patterns and evaluation methods. Security review is repeated. Costs cannot be compared. Successful pilots reach production with bespoke support needs.

The problem is not the number of suppliers. It is the absence of an owned architecture for substitution, evidence and control.

3. What the Early Evidence Shows

Patterns across enterprise programmes are becoming clear.

3.1 Full internal build

Internal teams can create strong control over data, interfaces and workflow. This is valuable in highly regulated or genuinely differentiating applications. But many build programmes underestimate the continuing work: model changes, evaluation, abuse testing, observability, cost management, user support and new integration requests.

The organisation celebrates avoiding licence cost while creating a permanent engineering obligation.

3.2 End-to-end suite purchase

Suites accelerate access and provide coherent administration. They work well for common employee tasks and bounded productivity use cases. Problems appear when the organisation assumes that common controls make every embedded AI use responsible by default. The supplier governs the product; the enterprise still owns purpose, data, user behaviour and operational consequence.

Commercial convenience does not transfer accountability.

3.3 Unmanaged orchestration

Teams combine model services, retrieval tools, workflow products and custom code. This can produce rapid, high-quality pilots. Without shared standards, it also produces duplicated gateways, inconsistent logging and credentials scattered across projects. “Best of breed” becomes “each team owns a different production risk”.

Orchestration works only when the organisation deliberately owns the seams.

4. A Composite Portfolio in Numbers

Consider a composite enterprise with twelve AI initiatives.

Four are employee-assistance tools: drafting, summarising, translation and knowledge search. Three support customer operations. Two analyse technical documents. Two automate finance exceptions. One supports a high-value pricing decision.

The initial strategy selects one enterprise suite for all twelve. The three-year commercial estimate is £6.8 million. The suite covers common assistance well, but two document-analysis cases require specialised models, finance needs controlled tool execution, and pricing requires independent evaluation and strict data separation.

Teams begin adding custom services around the suite. After nine months, the portfolio contains:

- one strategic suite
- three external model services
- two retrieval implementations
- four evaluation approaches
- seven separate data connectors
- five monitoring dashboards
- no common record of prompt, model, evidence and action

The organisation has neither bought a platform nor built one. It has bought a centre and improvised the edges.

Support cost reaches £1.4 million annually, but the larger cost is change. A model substitution requires separate testing in eight workflows. A security control is implemented five times. Usage cost cannot be traced consistently to business outcomes.

The programme then introduces a thin owned orchestration layer: common identity, model gateway, approved connectors, event logging, evaluation records and policy enforcement. Employee assistance remains in the suite. Specialised models stay external. Pricing retains a dedicated environment. Within two releases, connector duplication falls from seven to three and all production workflows generate the same minimum evidence.

The value did not come from replacing suppliers. It came from deciding which integration and control responsibilities the enterprise would own.

5. The Four Layers of the Decision

A useful platform assessment separates four layers.

Layer	Default stance	Build when	Buy when
Experience	Buy or configure	Workflow is strategically distinctive	Task is common and low consequence
Orchestration and control	Own a thin enterprise layer	Authority, evidence or substitution matters	Use case is fully bounded inside one suite
Models and specialised capability	Buy access, preserve choice	Proprietary model creates proven advantage	Market capability is sufficient and replaceable
Data and knowledge	Own governance and access	Meaning, lineage and rights are distinctive	External data is licensed and controlled

This separation prevents one decision from being mistaken for four.

The experience layer can change without moving the data. A model can be replaced without rewriting the workflow. A common control layer can enforce identity, budgets and evidence across suppliers. Sensitive use cases can remain isolated without forcing the entire enterprise into a specialised architecture.

Modularity is not an aesthetic preference. It is the mechanism by which commercial choice remains real.

6. The Strongest Case for One Strategic Supplier

There is a serious argument for consolidation. Enterprise technology estates are already fragmented. AI adds scarce skills, new security concerns and rapidly rising demand. A strategic suite can provide identity integration, administration, support and contractual clarity. Concentrating usage may improve commercial terms. Standard tools can reduce uncontrolled experimentation.

For many low-consequence use cases, this is the right answer.

The error is extending that conclusion from common productivity to every AI-enabled decision. A supplier can provide a consistent environment, but it cannot make all enterprise data equally suitable, all models equally capable or all workflows equally safe. Nor can it guarantee that today's leading component remains appropriate for every workload.

The second serious objection concerns orchestration itself. An owned control layer can become an expensive internal product. If designed too broadly, it recreates a platform team that delays delivery and offers fewer features than the market.

That risk is real. The answer is a thin control plane, not a new universal AI stack.

Own only the functions required across material use cases:

- identity and permission enforcement

- approved model and tool access
- usage and cost attribution
- minimum event and decision logging
- evaluation records and release evidence
- policy enforcement and stop controls
- standard connectors for priority data
- portability of prompts, tests and workflow definitions where practical

Everything else should justify itself through repeated demand.

The enterprise should own the evidence and authority that make AI governable, not every component that makes AI possible.

7. A Decision Method for Build, Buy and Orchestrate

Apply the decision at component and use-case level, not once for the whole enterprise.

1. Classify the consequence. Is the capability assisting a person, influencing a material decision or acting in a system? Higher consequence increases the need for owned controls and evidence.
2. Test distinctiveness. Does the capability embody proprietary data, workflow or judgement that creates real advantage? If not, buying is usually more rational.
3. Measure substitution cost. Can the model, retrieval service or user experience be replaced without moving sensitive data or rebuilding controls? If not, dependency is already accumulating.
4. Separate control from capability. Identify what the supplier controls and what the enterprise remains accountable for.
5. Price continuing operation. Include evaluation, monitoring, model change, data maintenance, incident response and user support.
6. Choose the smallest owned layer. Build only enough shared capability to govern, integrate and substitute the components the portfolio actually uses.

The decision should be revisited as evidence changes. A component that is commodity today may become strategically important through scale or data. A custom component may become unnecessary when market capability improves.

Platform strategy must therefore be governed as a portfolio of dependencies, not frozen as a product selection.

8. The Operating Model Matters More Than the Diagram

A modular architecture fails if accountabilities remain fragmented.

The platform product owner should own shared services and developer experience. Use-case owners should remain accountable for outcomes, data and decision

consequences. Security, privacy and risk functions should define reusable controls rather than review every implementation from the beginning. Procurement should negotiate for portability, auditability, data rights and service-change notice—not only unit price.

Architecture needs explicit authority to refuse designs that bypass common evidence or identity controls. It also needs service commitments so teams are not forced around the platform by delay.

The minimum governance should track:

- production use cases and owners
- models, tools and data sources used
- supplier and component dependencies
- evaluation status and material limitations
- run cost by use case
- incidents and control exceptions
- substitution and exit readiness
- concentration of sensitive data or critical decisions

This is platform governance as an operating discipline, not a reference architecture.

9. What to Recommend in 2024

A credible enterprise prescription is selective.

Use a strategic suite for common, low-consequence employee use where its controls and economics are adequate. Preserve access to more than one model capability for workloads where performance, cost or data terms differ materially. Isolate high-consequence decisions when shared services cannot provide sufficient evidence or control. Build differentiated workflows only where operational ownership and value are clear.

Across these choices, establish a thin enterprise orchestration and control layer that the organisation owns. It should make approved components easy to use, unapproved paths harder to use and every material workflow observable.

This is not a compromise between build and buy. It is a clearer allocation of responsibility.

Buyers should buy capability without buying avoidable captivity. Builders should build differentiation without rebuilding commodities. Platform teams should orchestrate without becoming the new bottleneck. Leaders should judge success by reusable control, faster responsible deployment and the ability to change a component without changing the whole service.

10. The Prescription

The question “Which enterprise AI platform should we choose?” invites an answer that will age quickly.

The more durable questions are:

- Which capabilities are common enough to buy?
- Which decisions are distinctive enough to build around?
- Which evidence, permissions and data must remain under enterprise control?
- Which components must be replaceable?
- Which seams will the organisation own as a product?

Answer those questions, and platform selection becomes manageable. Ignore them, and every apparent choice produces a different form of lock-in: to internal engineering, to a strategic supplier or to a collection of ungoverned integrations.

The recommended strategy is to buy broadly, build selectively and orchestrate deliberately.

Enterprise AI will continue to change faster than ordinary platform cycles. The architecture must therefore preserve the right to learn. An organisation that owns its decision logic, evidence and control boundaries can adopt better capability as it appears. One that embeds them inside a single product will discover that the platform decision was also a decision about how difficult future change would become.