FOXTROT

AI Adoption That Scales. Value That Lasts.

EXECUTIVE BRIEF

IF YOUR AI PROVIDER CHANGED TOMORROW, WHAT WOULD YOU HAVE TO REBUILD?

Real control means keeping the context, workflows and governance that make your AI valuable, even when the technology underneath it changes.

AI & Data Solutions | www.foxtrot-services.com

October 2026

AT A GLANCE

Most leaders expect to be able to change AI providers within four weeks. Most who have tried found it harder than expected.

89%

of enterprise leaders say they could move to a new AI provider within four weeks

ZAPIER, 2026

58%

of enterprises that have attempted a migration between AI platforms say it failed outright or took significantly more effort than expected

ZAPIER, 2026

→ **In a separate study of 1,000 senior executives, 71% said switching their primary AI vendor or model would be difficult.**

Sources: Zapier AI vendor survey, April 2026, comprising 542 U.S. executives and decision-makers; IBM Institute for Business Value and Oxford Economics, *The Calculus of AI Sovereignty*, June 2026, comprising 1,000 senior executives globally. The 58% figure applies only to respondents who had attempted a migration.

KEY POINTS

- **The model is the visible part of an AI system.** Much of the switching cost sits in the business context, logic and evaluations around it, and in whether the enterprise controls them.
- **The leading model changes often.** A company tied to one model inherits its tradeoffs; one with a model-agnostic harness can evaluate and adopt a better-fit model with less rework when one appears.
- **Sovereignty is a gradient, not a switch.** More control, up to self-hosting, brings more cost, more expertise and sometimes a capability gap, so the right level depends on the workload.
- **Governance should outlast the model.** Permissions, evaluations and business rules that sit outside the model carry over when the technology underneath them changes.
- **Change is already on the calendar.** Providers retire older models on published schedules, so an enterprise running on a hosted model should expect to move at some point.

CONTENTS

01 The useful question is who keeps the decision rights

02 Sovereignty is bigger than the model

03 Build for model agility

04 Privacy is where the conversation usually starts

05 Protect how your company works

06 Sovereignty is a gradient of decision rights

07 Governance becomes part of sovereignty when AI can act

08 Sovereignty is easier to promise than it is to engineer

09 Sovereignty gets tested when something changes

10 The durable asset is the company's ability to choose

01 The useful question is who keeps the decision rights

Most companies don't think about AI sovereignty until something changes.

Often, the first question is about data. Where does our information go? Who can access it? How long is it retained? What happens when proprietary data becomes part of an AI workflow?

Those questions matter. They're also only one piece of the sovereignty puzzle. Once AI starts creating value inside a business, control diffuses across the system. It reaches the data the AI uses, the logic behind decisions, the models doing the work, the compute underneath them and the rules that determine what the system is allowed to do. That's why we think the idea of **institutional sovereignty** is useful.

The market often talks about model sovereignty or AI sovereignty. We think the more practical question is who keeps the decision rights. Who decides how company data is used? How easily can you change the model or provider when requirements change? Where does the model run? Who owns the business logic? What happens when the company needs a different architecture? Each of those is a decision about the AI system itself. The enterprise should be the one making them.

Most companies haven't worked out where those rights currently sit. In a 2026 study of 1,000 senior executives by the IBM Institute for Business Value and Oxford Economics, 91% said they don't fully understand their organization's dependencies across AI vendors, models and infrastructure. In the same study, 73% described their AI environments as intentionally multi-vendor, which isn't the same thing as being able to move between them.

> **The model is the visible part. The bigger stakes often sit underneath it.**

EXHIBIT 1 WHERE CONTROL SITS IN A PRODUCTION AI SYSTEM

LAYER	WHAT SITS THERE
Model THE VISIBLE PART	Produces responses, predictions and tool calls. The most visible layer, and the layer the architecture should make easiest to swap.
	WHAT SITS UNDERNEATH
Business logic	The rules, decision criteria and workflow definitions that encode how the company operates.
Governance	Permissions, evaluations, audit trails and human-review rules that define, assess and document what the system may do.
Compute	Where the model runs: a provider's cloud, a private cloud or on-premise hardware.
Data	The company information the system can reach, and the boundaries that limit its access.

Each layer carries its own decision rights, and many of the harder dependencies sit below the model.

Our perspective comes from our Forward Deployed Engineers, embedded in real customer workflows, where they see firsthand how enterprise knowledge, business logic, permissions and evaluations become connected to the model layer. Foxtrot's sovereignty work is built around preserving those enterprise-owned assets even when

the underlying model changes. That gives companies more freedom to evaluate, switch or combine models as performance, security, latency, regulatory requirements and cost change. We call that **model agility**. With the best available model changing from one release to the next, it's one of the most practical tests of whether a company actually has sovereignty.

02 Sovereignty is bigger than the model

Swapping your LLM may sound simple.

In practice, the model is only one piece of a production AI system. The workflow around it may contain business rules, enterprise context, evaluations, permissions, tool access and connections to other systems. Those elements are what turn a capable general model into something that can actually work inside a particular company.

Over time, they can become deeply embedded, and the companies that have tried a move tend to find that out partway through it. In a 2026 Zapier survey of 542 U.S. executives and decision-makers, two-thirds said they had already attempted to migrate between AI platforms, and 58% of those said the move either failed outright or required significantly more effort than expected.

Portability depends on keeping the enterprise-owned layers around the model as durable and separable as possible. That can include a shared data layer with single sources of truth, enterprise-owned knowledge and workflow definitions, clear interfaces to models and tools, permissions enforced independently of the model, reusable evaluation datasets and execution records the company can retain or export.

That doesn't make changing models plug-and-play. A new model still needs to be revalidated for accuracy, tool use, latency, behavior and the specific requirements of the workflow, and some changes may require engineering work. The advantage is that the enterprise doesn't have to rebuild its business context, workflow definitions, permissions and evaluation criteria from scratch.

EXHIBIT 2 WHAT SHOULD SURVIVE A MODEL CHANGE, AND WHAT STILL NEEDS CHECKING

PART OF THE WORKFLOW	WHAT SHOULD REMAIN USABLE	WHAT STILL NEEDS CHECKING
Business knowledge	Definitions, relationships, operating context and single sources of truth	Whether the new model retrieves that context and reads it the way the business does
Business logic	Rules, workflow definitions and action limits held outside the model	Whether the replacement calls tools correctly and respects the limits
Governance	Access policies, approval requirements and retained records	Whether those controls cover the new deployment and its data paths
Evaluations	Representative test cases and the business criteria for accepting a result	Performance itself, which has to be rerun and compared
Integrations	Documented interfaces and downstream data requirements	Provider-specific adapters, prompts, output handling and error behavior

What carries over depends on the implementation, the contractual rights and the export mechanisms available. Changing the model is usually the most contained of these changes, which is a reason to choose the layers beneath it deliberately.

Ownership, export and reuse are three different questions

Strong contractual ownership and a complete export both matter, and both are worth establishing explicitly rather than assumed. What neither settles on its own is whether the definitions, workflows and controls that make the data useful would run somewhere else. That depends on how the system was built rather than on the terms it was bought under.

Model agility is also not the same as having no commitments. A company that can change models still runs on a cloud, on a platform that coordinates its AI workflows, on a shared model of how its business objects relate to one another, and with the team that built the system. Most of those commitments are deliberate, and they are usually where the value sits. What matters is that the enterprise chose each one knowingly, knows what it buys the business, and could say what changing it would involve.

That's where sovereignty becomes more than portability.

For an enterprise, sovereignty means retaining enough control over the system that a technology decision made today doesn't quietly determine how the business has to operate years from now. The company should be able to preserve the knowledge and operating logic it has built while continuing to make new choices about the technology underneath it. That's the difference between having access to another model and having meaningful freedom to use one.

03 Build for model agility

The model leaderboard doesn't sit still.

One week the strongest model comes from one provider, and the next week another provider's release overtakes it. The differences aren't binary, either. The model that leads on a coding benchmark may be the weaker writer, and the gaps between the leaders are often narrow enough that the ranking depends on which task you care about.

> **A company that can only use one model**
> **has to accept every tradeoff that model makes.**

Change doesn't always arrive as a better option, either. Model lifecycles are public, and providers publish retirement dates ahead of time with migration guidance and a named successor. That still means a company running on a hosted model should plan for a move it did not initiate, and the revalidation work lands on the enterprise either way. An architecture built around a single model makes that work larger than it needs to be.

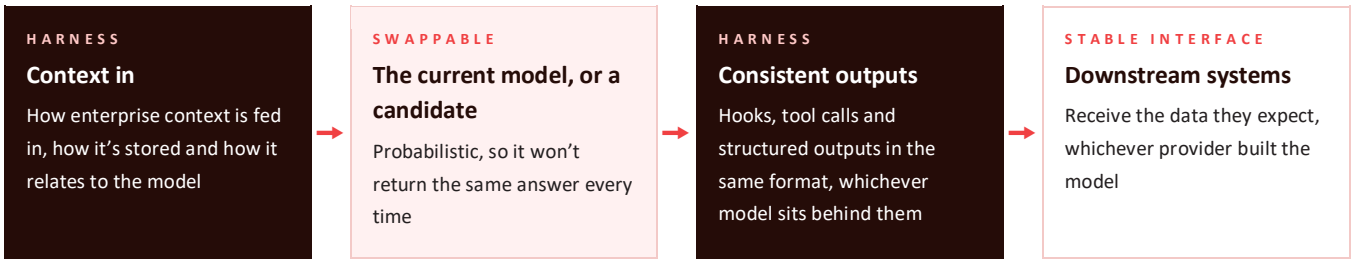
Retirement is not the only kind of change, either. Pinning a specific model version reduces exposure to model updates, but does not eliminate every source of behavioral change. Using an automatically updated alias can bring changes without an explicit migration, depending on the provider's versioning policy. Both approaches can be reasonable, and both require repeatable, enterprise-owned evaluations to confirm that a workflow is still performing.

Benchmarks alone won't tell you which model fits a workflow. Sometimes the only way to know is to get hands-on with a model the day it's released, before any consensus has formed about what it's good at. That takes people who understand which models are good at what, and an architecture that lets the company act on their judgment quickly.

That architecture is the harness around the model: how context is fed in, how it's stored and how it relates to the model. Models are probabilistic and won't return the same answer every time, so the systems around them have to create determinism. Hooks, tool calls and structured outputs should return data in the same format no matter which model sits behind them or which provider built it. That consistency is what makes a model hot-swappable.

Consistent interfaces help systems connect, but they don't prove equivalent behavior. A new model can return exactly the expected data format and still make a different decision, which is why testing has to cover the real work, including the hard cases and the failures.

EXHIBIT 3 WHAT MAKES A MODEL HOT-SWAPPABLE



A probabilistic model can't be consistent on its own, so the harness around it has to be. That consistency is what lets the model in the middle change.

Gartner's guidance points the same way. Its January 2026 recommendations to CIOs on region-specific AI platforms included: "Design model agnostic workflows using orchestration layers that enable switching between LLMs across regions and different vendors."

Switching will always carry some cost. The closer a company gets to minimizing it, the faster it can adopt a better model when one appears, and the less any single provider shapes how the business operates.

04 Privacy is where the conversation usually starts

Enterprises are putting increasingly sensitive information into AI workflows.

The right protections will depend on the product, provider, enterprise agreement and deployment. Some workloads may be well served by managed models with strong enterprise privacy, security and retention controls. Other workflows carry different stakes. A highly sensitive transaction, a regulated process or a mission-critical operation may require tighter boundaries around where data moves, where inference happens and who ultimately controls the infrastructure.

Partial trust is easy to state and hard to enforce

Deciding that a provider can be trusted with some data but not the rest is only half the problem. Enforcing it is harder. A local model can screen requests before they leave, or a routing layer can decide which data goes to which model, but model-based screening and routing are not fully deterministic. Deterministic enforcement can

mean inspecting outbound requests for restricted content, which adds more overhead than most organizations expect. And once employees have an open conversational interface, what goes into the model is up to their discretion, because it's free text. That's why the starting point has to be least privilege: decide what each model and interface is trusted with before anyone starts typing.

The sovereignty question is therefore bigger than whether a provider can be trusted. It's about deciding which rights the enterprise needs to keep for the work in front of it.

05 Protect how your company works

A general-purpose model doesn't arrive knowing how your business makes decisions.

It doesn't know which exception an experienced operator recognizes immediately. It doesn't know why one signal matters more than another, which unwritten rule changes a decision or how people resolve the cases that never made it into the process manual.

That knowledge already exists inside the enterprise. Some of it lives in structured data. Some of it is buried in documents, systems and conversations. A lot of it still lives in people's brains. The people who carry that knowledge are often the least likely to notice it. After years of working under the same constraints and assumptions, tribal knowledge stops feeling like knowledge and starts feeling like the way things are done. It often takes someone new to the organization to see it.

A model starts where a new hire does, without any of that tribal context, and it can't absorb it informally. It won't pick up the unspoken cues around a project or what was decided in meetings nobody transcribed.

> **If it isn't written down, an agent can't use it.**

This is where forward deployed engineering becomes particularly important. An FDE brings the outsider's eye to work that has long since stopped looking unusual to the people doing it. Embedded in the workflow, our FDEs work with operators to uncover the edge cases, decision rules and operating patterns that make the business work in practice, then translate them into structures the AI system can actually use.

In an ontology-led architecture, that can mean turning institutional knowledge into business objects and relationships, workflow definitions, permissions and evaluation criteria. The goal is to make the knowledge that drives the business explicit enough to use, govern and evaluate in production. Writing it down is only the first step. The architecture also has to make that knowledge discoverable. And because an agent works within a limited context window, the system has to surface the most valuable information for the problem at hand, so the answer holds up against what a senior operator with years of context would say.

THE RESULT

Specialist expertise at scale

while preserving distinct information boundaries

IN PRACTICE | GLOBAL RISK FIRM

A global-risk firm needed to identify relevant intelligence among thousands of daily signals while delivering tailored insights across separate engagements. Foxtrot trained AI on the firm's analytical methods, consolidated previously separate environments onto a shared intelligence platform, and implemented granular permissions so each engagement's data, workflows and outputs remained accessible only to authorized users. The result was a reusable capability for applying specialist expertise at scale while preserving distinct information boundaries.

The decision loop can become an asset too

There's another layer to this that we think will matter more over time. A company can begin capturing the decision loop itself.

That operating history has the potential to become an AI asset in its own right. Over time, capturing those loops can make institutional knowledge more durable and reusable.

Once a company has done the hard work of encoding that context and logic, it shouldn't have to rediscover how its own business works every time the technology changes. The same holds for the partners who help build it. What we build is documented and handed over, so the enterprise can operate it with us, with another partner, or with its own team.

06 Sovereignty is a gradient of decision rights

This is where the conversation often becomes too binary.

Real enterprise environments are more complicated. Different workloads need different levels of control.

EXHIBIT 4 CONTROL SCALES WITH THE STAKES

WORKLOAD	APPROPRIATE LEVEL OF CONTROL
Low-risk internal productivity workflow	May be well served by a managed model under strong enterprise controls
Workflow touching more sensitive information	May call for tighter boundaries around data, routing or infrastructure
Highly regulated or mission-critical process	May justify an accredited or customer-isolated provider deployment, dedicated infrastructure, a self-hosted model or other forms of structural separation

The right level of control rises with the sensitivity of the data and the criticality of the process.

We think of this as a gradient, with the appropriate level of control depending on the sensitivity of the data and the criticality of the process. That's why we find **decision rights** more useful than a simple yes-or-no definition of sovereignty. As the stakes rise, an enterprise may choose to retain more authority over the data, logic, model, compute, governance and outcomes surrounding the workflow.

In practice, one way the gradient is enforced is through access. A model should only be able to reach the subset of data its workflow is authorized to use, with markings, guardrails and governance holding that boundary in

place. It works the way security clearances do on a classified network: someone cleared for Secret sees Secret information and nothing above it. The gradient is real, and each level on it can function as a gated knowledge silo.

> **The goal isn't maximum control everywhere. It's the right control for the work.**

More control comes with more responsibility

Self-hosting sits toward one end of that spectrum. Running an open-weight or custom model inside infrastructure the enterprise controls can provide tighter boundaries around data, inference and model operations. Foxtrot's work includes these patterns for situations where private cloud, customer-controlled infrastructure or on-premise deployment may be appropriate.

That level of control is powerful, but it isn't free. The enterprise takes on more responsibility for model serving, infrastructure, performance, security and compute. Self-hosting models that approach frontier performance requires significant hardware and specialized technical expertise. In practice, specialized infrastructure expertise can be a meaningful constraint for companies pursuing self-hosted approaches outside established platforms.

There is still a ceiling on what self-hosting can reach. Some of the most capable proprietary frontier models are available only as hosted services, however much budget, hardware and expertise an enterprise brings to the problem. That gap has been closing, though, and open-weight models now sit behind a growing number of deployments built for classified, air-gapped and other controlled environments. For a given workload, the controlled end of the spectrum is increasingly a deliberate choice rather than a fallback, which is all the more reason to test a candidate against the actual work before committing to it.

So the question shouldn't be whether self-hosting is inherently better. The question is whether the additional control is worth the added responsibility (and any capability tradeoff) for that particular workload. For some, it will be. For others, a managed model with the right enterprise controls will be the smarter architecture.

Sovereignty is valuable because the company retains the ability to make that choice.

07 Governance becomes part of sovereignty when AI can act

The control question gets more serious as AI moves from producing information to taking action.

An agent may query sensitive systems, create records, initiate workflows, recommend decisions or act on someone's behalf. Now the enterprise has to control much more than the model response.

EXHIBIT 5 SIX QUESTIONS TO SETTLE ONCE AI CAN ACT

01	Who owns the agent?	04	How is its performance evaluated?
02	What can it access?	05	When does someone need to step in?
03	Which actions is it authorized to take?	06	Can the organization reconstruct what happened later?

Plenty of enterprises will learn this after the fact. In May 2026, Gartner predicted that by 2027, 40% of enterprises will demote or decommission autonomous AI agents due to governance gaps identified only after production incidents occur. “Enterprises are treating AI agent governance as binary, either locked down or fully trusted, and that is the root cause of failure,” said Shiva Varma, Senior Director Analyst at Gartner.

THE GOAL

Consistent institutional oversight

even when agents were built with different tools and frameworks

IN PRACTICE | FORTUNE 100 ENTERPRISE

For one Fortune 100 enterprise looking to scale agents across business lines and corporate functions, Foxtrot designed a common control layer to register, evaluate and certify agents before production, connect them to trusted enterprise knowledge with permissions and human approvals, and monitor behavior, performance and risk. The goal was consistent institutional oversight even when agents were built with different tools and frameworks.

These controls matter in every enterprise. They become particularly important in regulated and high-stakes environments, where the organization may need to explain what happened, who had authority and which policies were applied.

There’s a sovereignty implication here too.

Governance shouldn’t have to be reinvented every time the model changes.

Take a customer asking an AI agent for a refund. How much any model is allowed to refund should be capped by deterministic business logic that sits outside the model and is accountable for the outcome. Then it doesn’t matter which model handles the conversation, because the rule holds either way.

There is a second benefit to moving settled logic out of the prompt. Rules that live in code are cheaper to run, easier to test and simpler to explain than rules a model has to be asked to follow on every request.

If changing one component of the AI stack also means reconstructing the permissions, evaluations and human-review rules around the workflow, the enterprise has created another form of dependency. Durable governance gives the company more freedom to change the technology without giving up control of how the work gets done.

08 Sovereignty is easier to promise than it is to engineer

The concept sounds clean. Real enterprise environments rarely are.

Many companies don't yet have their data or operating knowledge organized well enough to make that context durable. Legacy systems create another challenge. Some were built long before modern APIs or newer agent connectivity standards existed, which makes connecting AI safely to existing data and tools far more complicated.

Self-hosted environments require specialized infrastructure expertise. Governance has to span systems. Evaluations need to remain meaningful as models change. Data boundaries have to be understood rather than assumed. In practice, data readiness, legacy integration and specialized infrastructure expertise are among the biggest barriers to sovereign AI architectures.

The practical goal is to make every part of a workflow that can be deterministic actually deterministic, and leave the model to handle the rest. Getting close is hard when systems and data change daily, but it's achievable. It takes sustained time and effort, and forward deployed engineers close enough to the business to keep up as it changes.

That's why we think sovereignty is better understood as something companies build toward than something they simply turn on.

> **Promising sovereignty is the easy part.**
> **The architecture is where it has to hold.**

09 Sovereignty gets tested when something changes

An AI environment can feel flexible when everything keeps working exactly as planned.

The revealing moment comes when the business needs something different. A new model performs better. The economics change. A workload becomes subject to stricter regulation. Sensitive information needs to remain inside a controlled environment. An agent is given more autonomy.

That's when earlier architecture decisions become visible.

EXHIBIT 6 **WHEN SOMETHING CHANGES, ASK**

01	Can the company change the model while preserving the context the workflow depends on?
02	Will the evaluations still tell the team whether the system is performing well?
03	Do the hooks, tool calls and output formats around the model still return what downstream systems expect?
04	Do permissions, auditability and human-review requirements continue to work?
05	Could a sensitive workflow move into a more controlled environment if the business required it?

Those questions tell you a lot about where the real decision rights sit.

10 The durable asset is the company's ability to choose

A more durable investment is protecting the parts of the system that belong to the enterprise.

The context should remain usable. Governance should remain enforceable. Business logic shouldn't disappear because a model changes. The architecture should leave room for different choices when performance, security, regulation or economics require them, which is what **model agility** looks like in practice.

That's what institutional sovereignty means to us.

> **It's the ability to keep making decisions about your own business as the technology around it changes.**

SELF-ASSESSMENT

Where do your decision rights sit?

Pick the one AI workflow whose failure would materially affect the business, and answer it with the business owner and the technical team in the room, using evidence from the system rather than recollection.

01	**We know where the dependencies sit** We can name the model, platform, infrastructure and integration dependencies, and what the business would lose if each became unavailable.	☐ YES, WE'VE TESTED IT ☐ TO SOME EXTENT ☐ NOT YET
02	**We can retain and reuse our operating knowledge** Business definitions, workflow rules, evaluation cases and records are documented, accessible and reusable within our rights.	☐ YES, WE'VE TESTED IT ☐ TO SOME EXTENT ☐ NOT YET
03	**We have tested an alternative** A candidate replacement has been run against representative tasks, including the hard cases, tool use and failure handling.	☐ YES, WE'VE TESTED IT ☐ TO SOME EXTENT ☐ NOT YET
04	**Our controls hold independently of the model** Permissions, action limits, approvals and audit records can be verified in the system itself rather than inferred from how the model behaved.	☐ YES, WE'VE TESTED IT ☐ TO SOME EXTENT ☐ NOT YET
05	**We know what a change would take** The team can separate what stays usable from what needs retesting, adapting or rebuilding.	☐ YES, WE'VE TESTED IT ☐ TO SOME EXTENT ☐ NOT YET
06	**We have a workable fallback** There is an accountable owner, a controlled change process and a tested fallback suited to how critical the workflow is.	☐ YES, WE'VE TESTED IT ☐ TO SOME EXTENT ☐ NOT YET

This is a discussion tool rather than a maturity score. One untested assumption in a critical workflow can matter more than several confirmed answers elsewhere.

Mostly "Yes, we've tested it"

You have a stronger starting point for keeping your options open. The next test is whether that holds as more agents come online and more of the business depends on them.

Several not yet

The workflow may carry dependencies nobody chose. Start with the one the business relies on most and trace what a model change would touch, from context and logic through to permissions and evaluations.

If you can't say what a change of provider would touch, start with the workflow you would least want to rebuild.

Bring us the workflow your business can least afford to interrupt. Foxtrot helps enterprises identify critical AI dependencies, establish the right controls, and build or adapt production workflows that preserve enterprise knowledge, governance and flexibility as models and requirements change.

Talk to Foxtrot about AI sovereignty. hello@foxtrot-services.com | www.foxtrot-services.com

About Foxtrot. Foxtrot Professional Services is a Palantir partner founded by Palantir alumni and enterprise operators. Forged in Foundry, Foxtrot builds production-grade, governed AI systems for commercial and government organizations that are designed for adoption, flexible across models and accountable for what they cost.