Konstera

GUIDE

AI Agent Platform *Buyer's Guide.*

Ten evaluation criteria for picking an AI agent platform in 2026, written by a vendor and designed to be used against that vendor. Scorecards for five real competitors, when to pick each, and when Konstera isn't the right answer.

25-minute read · 10 criteria · 5 vendors · Konstera · 2026

Vendor-neutral · includes Konstera's honest self-assessment

GUIDE

Contents

– Why a vendor wrote this
01 Platform or framework
02 LLM lock-in posture
03 Human-in-the-loop product quality
04 Multi-tenancy at the data layer
05 Open source stance
06 Cost attribution and ceilings
07 Audit and replay
08 Compliance roadmap with dates
09 Deployment flexibility
10 Vendor longevity
– Vendor matrix
– When NOT to pick Konstera

The vendor-neutrality paradox.

Three good reasons to be sceptical of a buyer's guide written by a vendor. We're one of the vendors; of course we're going to lean toward our strengths. The criteria will quietly privilege the shape of product we built. The vendor matrix will conveniently show us winning.

We've tried to counter all three. The criteria are written as questions, not features — so they apply even to platforms that look nothing like Konstera. The vendor matrix includes four competitors we respect, each with a clear "pick this when" line. And section 11 ("when NOT to pick Konstera") is explicit about where we're the wrong answer.

Even with those honest moves, you should use this guide as one input, not a verdict. The goal is to help you ask the ten questions that matter — not to tell you the answer.

01 Platform or framework

Is this a platform that runs your agents, or a library you ship with?

A framework (LangChain, Semantic Kernel, Mastra) is a library you deploy as part of your agent code. A platform (Konstera, Palantir Foundry) runs the agent you built. Both are legitimate; mixing them up produces budget and governance surprises.

GOOD SIGNAL

Vendor can describe their deployment surface, multi-tenancy posture, and durable workflow story in one call.

RED FLAG

Vendor pitch is a GitHub README with import statements.

02 LLM lock-in posture

Can we use any LLM, or just the ones they resell?

BYOK (bring your own key) is the right posture for mid-market buyers. You contract with Anthropic / OpenAI / Google directly, control the invoice, and the vendor meters usage without touching the money.

GOOD SIGNAL

Multi-provider router (Claude + GPT + Gemini + DeepSeek at minimum), BYOK keys per tenant, no vendor markup.

RED FLAG

Vendor wraps the LLM, bills you per-token with a markup, discourages direct provider contracts.

03 Human-in-the-loop product quality

Is HITL a product with its own UX, or a feature nobody built?

Reviewer inbox UX is where most platforms quietly fail. A good HITL product reads like a thread; a bad one reads like a Zapier dashboard. The difference compounds across every reviewer, every day.

GOOD SIGNAL

Thread-first inbox, four actions per message (send / edit / redraft / reject), autonomy as scope-stacked policy, full replay.

RED FLAG

A "review queue" page with approve/reject buttons and no context panel.

04 Multi-tenancy at the data layer

How is cross-tenant data isolation enforced?

Application-code filtering (if-tenant-id-equals) is one missed query from a breach. Database-layer isolation (PostgreSQL RLS, dedicated per-tenant databases, separate schemas) is the only answer worth signing a contract against.

GOOD SIGNAL

RLS on every tenant table, NOBYPASSRLS platform role, per-connection GUC, independently verified in schema.

RED FLAG

"We filter by tenant_id in our ORM." If that's the whole answer, don't buy it.

05 Open source stance

What's open source, what's source-available, what's proprietary?

The honest answer is a spectrum. SDK as Apache-2.0 is table stakes. Platform as source-available (BSL 1.1) is defensible. Pure proprietary with no inspection rights is a 2027 switching-cost trap.

GOOD SIGNAL

SDK and reference agents under Apache-2.0 on public GitHub, platform source-available with time-limited conversion.

RED FLAG

Zero source access, agent code locked in vendor's proprietary runtime, no self-host option.

06 Cost attribution and ceilings

Can we see — and cap — AI spend per agent, per team, per model?

Attribution is where ROI arguments live or die. Enforced ceilings are where governance happens. Vendors who can't deliver attribution per-agent will deliver surprise invoices.

GOOD SIGNAL

Per-agent, per-team, per-model, per-tenant attribution surfaced in portal. Monthly caps enforced at invocation time.

RED FLAG

Blended monthly bill, no mid-month visibility, no cap enforcement.

07 Audit and replay

Can we reconstruct any past invocation and re-run it against a new agent version?

Audit without replay is a compliance checkbox. Replay is how you validate fixes against failing cases before deploying. Replay against a different agent version is the serious operational tooling.

GOOD SIGNAL

Every invocation reconstructable (inputs, model, prompt, tool calls, reviewer actions). Replay button in the portal.

RED FLAG

Logs. Just logs. No replay, no reconstruction tooling.

08 Compliance roadmap with dates

What's attested now; what's on the roadmap with a date?

Honesty about compliance posture is a product-maturity signal. Vendors who claim "SOC 2 compliant" when they mean "starting the Type II audit next quarter" are telling you about the rest of their product discipline.

GOOD SIGNAL

SOC 2 Type II date, GDPR DPA ready, HIPAA BAA path, data residency pinned per tenant, subprocessor list current.

RED FLAG

Vague "enterprise-grade" claims. No DPA. No subprocessor list on request.

09 Deployment flexibility

SaaS, Helm, Compose, air-gapped — same images?

Feature parity across delivery shapes is the test. Vendors with a watered-down self-hosted tier are telling you that self-host is a Plan B for them. Mid-market retailers under regulatory pressure need Plan A on-prem to be real.

GOOD SIGNAL

Managed SaaS, Helm, Compose, air-gapped from one source. BSL 1.1 or similar source-available licence.

RED FLAG

SaaS only, or self-hosted "community edition" with the enterprise features gated.

10 Vendor longevity

If this vendor disappears in 2028, what's your exit?

Startup risk is real. The test isn't whether the vendor will exist in three years — it's whether their disappearance is survivable. OCI containers, MCP protocol, Apache SDK, BSL platform — all portable. Proprietary runtime, undocumented wire format — not.

GOOD SIGNAL

Open standards at every layer (MCP, OCI, OIDC, OTLP). Agents portable off the platform. Source-available platform you can run yourself.

RED FLAG

Agents written in vendor's proprietary SDK. Wire format undocumented. Runtime closed-source. Can't self-host as a last resort.

Five vendors against ten criteria.

No scoring. Scoring reduces useful nuance to a number for comparison tables; nuance is where buying decisions actually live. Instead: a short honest line per vendor, per criterion — and a "pick this when" verdict at the end.

CRITERION	PALANTIR FOUNDRY	SALESFORCE AGENTFORCE	MICROSOFT COPILOT STUDIO	LANGCHAIN / LANGGRAPH	KONSTERA
Platform or framework	Platform (data+agent)	Framework in Salesforce	Framework in M365	Framework (library)	Platform (agent)
LLM lock-in posture	Multi-provider	Vendor-chosen	Microsoft-chosen	Multi-provider	Multi-provider, BYOK

CRITERION	PALANTIR FOUNDRY	SALESFORCE AGENTFORCE	MICROSOFT COPILOT STUDIO	LANGCHAIN / LANGGRAPH	KONSTERA
Human-in-the-loop product quality	Workflow checkpoints	Feature, not product	Feature, not product	DIY	Product (ADR-017)
Multi-tenancy at the data layer	Enterprise-grade	Salesforce multi-tenant	M365 multi-tenant	DIY	PostgreSQL RLS
Open source stance	Proprietary	Proprietary SaaS	Proprietary SaaS	Open source	Apache SDK, BSL platform
Cost attribution and ceilings	Opaque at tier	Per-conversation pricing	Per-message credits	DIY	Per-agent / team / model
Audit and replay	Mature	Salesforce audit	M365 audit	DIY	Replay-first
Compliance roadmap with dates	SOC 2 Type II, FedRAMP	Salesforce stack	Microsoft stack	DIY	GDPR now, SOC 2 Type II 2027
Deployment flexibility	SaaS + FedRAMP cloud	SaaS only	SaaS only	DIY	SaaS + Helm + Compose + air-gapped
Vendor longevity	F500 moat	Salesforce bet	Microsoft bet	OSS	Open protocols, portable
Pick when	F100 with global ontology problem and 7-figure budget.	Salesforce-first businesses with agents inside the CRM.	Microsoft-first businesses with agents inside M365 / Teams.	Teams building their own platform from scratch. You still need everything on this guide — just build it yourself.	**Mid-market retail, wholesale and distribution ($50M–$500M). The honest answer to "a platform designed for us, not F500."**

When NOT to pick Konstera.

Four scenarios where another vendor is the right answer. Buying Konstera into any of these is unhappy for both sides.

You're a Fortune 100 with a global ontology problem.

Buy Palantir Foundry. It's not that we can't scale — it's that your use case is better served by the data-ops roots Foundry grew out of. Konstera is agent-native from day one for the mid-market; Foundry is enterprise-first with agents added. Different fits.

Your entire operation runs inside Salesforce or M365.

Buy Agentforce or Copilot Studio. If 95% of your agent interactions are inside a single vendor's UI, the agent should live where the UI lives. Konstera is the plane underneath multi-vendor stacks; a mono-vendor stack doesn't need it.

You have one agent and it's a chatbot.

Use a framework (LangChain, Semantic Kernel) and call it done. Don't buy a platform for a single workflow. Konstera is the right answer when you're deploying three or more agents across departments and want governance, cost control, and shared context from day one.

You can't name three enterprise systems the agents would touch.

The value of Konstera's connector framework and shared context compounds across systems. With fewer than three systems in play, you're paying for infrastructure you won't use. Come back when the integration graph is wider.

Konstera

One platform for every AI agent in your business.

Governed centrally, costed in one ledger, built on shared context — with people in control of every decision that matters.

konstera.ai · konstera.ai/book-a-call · konstera.ai/resources