WHITEPAPER · AGENTIC AI FOR PRODUCT LEADERS

What the Platform Gives You

And what you compose, what you write, and what nobody ships

BUY NOBODY SHIPS IT

How to read a vendor page without overestimating it or rebuilding what already exists. Five categories, four lines, and the eight capabilities this book asks you to own that no major platform ships as of mid-2026.

Yoram Friedman, MD
Physician and enterprise product leader · SAP, Walmart, Microsoft, regulated health technology
Drawn from the Agentic AI for Product Leaders series · agenticaiproductmanagement.com
Capability survey stamped mid-2026. Verify before quoting.

THE ARGUMENT

Two ways to get this wrong, and both of them are expensive.

The capabilities an agentic product requires do not arrive on your desk in a single product. They are distributed across what your platform ships, what your engineering team composes from platform events, what your product team writes as planning artifacts, what your organization enforces as policy, and what the industry does not ship at all.

A product manager who cannot read that distinction fails in one of two directions.

Overestimate the platform, assume the governance capabilities are in the box because the vendor page implied they were, and ship an agent with holes in it.

Underestimate the platform, rebuild capabilities that have been commoditized for a year, and lose months to work somebody else already finished.

> Both failures are common. Both are avoidable, and avoiding them is a reading skill rather than a technical one. You need a taxonomy for vendor pages, and a map of where each capability actually sits.

One caution before the map. This is the most perishable document in the series. Platform capability moves quarterly, and the gap line in particular is exactly the kind of thing that closes fast once a vendor decides to ship it. Treat every specific claim here as a dated snapshot and check it before you repeat it in a procurement meeting.

PART ONE

Five categories, and why vendors blur them

Almost every argument about what a platform should provide is really a disagreement about which of these five a capability belongs to.

A · PLATFORM PRIMITIVE

A first-class feature the platform exposes that you consume directly. Typed approval and interruption with pause and resume, agent tracing, judge evaluator libraries, versioned reference datasets.

B · DERIVED METRIC

A number you compute on top of events the platform already captures. Override frequency is a count of approval-interrupt events per unit time divided by total approvals. The platform emits the events; the metric is your composition. Almost every observation instrument lives here.

C · RELEASE CRITERION OR PLANNING ARTIFACT

Something your team produces during planning and tracks in a work-management system. A suitability declaration, a go or no-go memo, a coverage statement. Not a platform feature in any sense.

D · POLICY OR GOVERNANCE REQUIREMENT

An organizational standard the platform must support but does not define. Kill-switch obligations, retention and right-to-delete, authority delegation, bias mitigation.

E · BASE PLATFORM DEPENDENCY

Not agent-specific, but required. Access control and identity, general logging, secret management, data masking, rate limiting, tenant isolation. Vendors frequently repackage these as AI capabilities. They are platform hygiene with a new label.

Figure 1. Where each capability sits, as of mid-2026. Reading left to right is reading how much of the work is yours. The rightmost column is the one that decides whether your roadmap is realistic.

PART TWO

The commodity line: stop building these

These are Category A and available on every serious platform you will consider. Building them yourself is the second failure mode, and it costs months.

Tracing and observability. Every major platform emits, accepts, or standardizes on OpenTelemetry span capture. If a vendor claims AI observability as a differentiator, ask which events it emits that a base OpenTelemetry implementation does not. The answer is usually nothing, and the question is the fastest way to find out whether you are talking to an engineer or a marketer.

Judge evaluator libraries. Trajectory match, tool-call accuracy, groundedness, intent resolution. Available on every evaluation platform and most enterprise AI platforms. The methodology is commoditized. So is the bias literature on judge models, the longer-answer, position and same-family preferences. What is not commoditized is the calibration to your domain, which remains your work and is the part that decides whether the scores mean anything.

Typed approval and interruption with pause and resume. Every major agent framework has converged on the same pattern: a typed interruption object, durable run state, and an explicit approve or reject resume. If your framework does not expose this, you are on the wrong framework.

Agent identity bound to enterprise IAM. The major identity platforms now treat an agent as a first-class security principal. This is Category E hygiene relabeled, but it is available and you should consume it rather than invent it.

Per-trace token cost tracking, versioned reference datasets with lineage, date-stamped model version pinning and retirement notifications, and trajectory match at exact, in-order, any-order and superset granularity. All shipped. All boring. All yours for free.

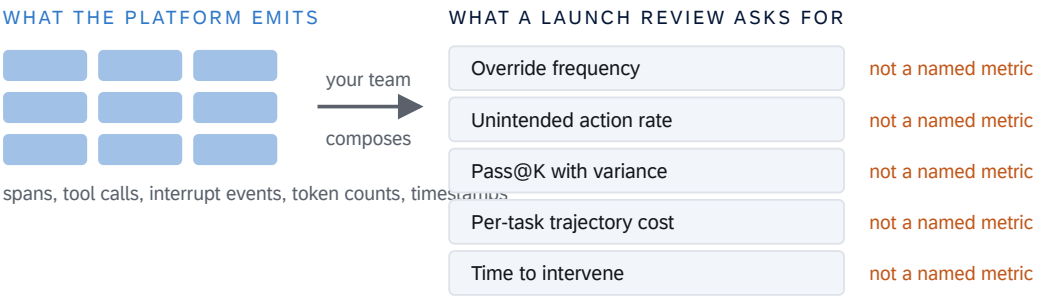
The question that reads a vendor page in one move

For any capability on the page, ask which of the five categories it is, and then ask what the platform does that a competent team could not compose from the events in the trace. A vendor selling a true Category A will answer immediately. A vendor selling Category B as though it were Category A will describe an outcome rather than a mechanism.

PART THREE

The derivation line: the platform emits, you compose

This is the line that causes the most damage, because the capability appears to exist. The data is there. The metric is not, and nobody tells you which is which.



The events exist on essentially every platform. The metrics exist on almost none.
An instrument you cannot produce is a finding about a surface nobody designed, not a gap in your data pipeline.

Figure 2. The gap between what a platform captures and what a launch review asks for. Every item on the right is derivable from the events on the left, and almost none of them ships as a named metric. Composition is your team's job.

The six observation instruments, task success rate, unintended action rate, override frequency, confidence calibration, rollback time and incident recovery time, are derivable from trace data on every platform and named as primitives on almost none. A small number of evaluation-specialist platforms ship a subset. They are still a subset.

Beyond those: **Pass@K with K and variance reported** is not a named metric on any major platform, though it is computable from repeated evaluation runs. **Compound probability across chained steps** is not shipped anywhere, though it is computable from per-step confidence scores if the platform emits them. **Per-task cost at the trajectory level**, including multi-agent coordination overhead and human review and rework time, is not shipped as a named metric, though it is derivable from token and span events plus integration with your human workflow system.

Time to intervene, response rate per interruption priority, and review depth sit in the same place. The trace data exists. The named metrics do not.

The practical consequence for a build plan is that every one of these is a small engineering task nobody scoped, because everyone assumed it came with the platform. Ask for them by name during evaluation, and treat any that cannot be produced at all as a finding about a surface nobody designed rather than as a reporting gap.

PART FOUR

The planning line and the policy line

Two lines that are not features at all, and where the mismarketing is most brazen.

What your team produces

The suitability assessment and the go or no-go memo. The coverage statement for the evaluation suite. The four-owner pre-launch readiness memo. Per-action consequence classification when produced as a design artifact. The retirement decision memo. The supervisory workflow composition document. The non-delegable list and the criteria that put a decision on it.

These live in your work-management system, not on a platform dashboard. The platform may help generate evidence for them. The artifacts are your team's output, and any vendor marketing them as primitives is mismarketing.

What your organization enforces

Right of access for agent decisions affecting named individuals. Source-document supersession and data freshness policies. Authority delegation and approval chains. Kill-switch obligations. Retention and right-to-delete for agent reasoning traces.

Security frameworks belong here too. The OWASP Top 10 for Agentic Applications and the Cloud Security Alliance's MAESTRO are policy frameworks the platform must support rather than features it provides. A platform may ship primitives that align with them, input trust classification, tool restriction by source. The policy is your organization's.

> A vendor that claims to deliver compliance is selling you a mechanism. The compliance obligation does not transfer with the purchase order, and no contract has ever made it transfer.

PART FIVE

The gap line: what nobody ships

Eight capabilities that the supervisory discipline asks you to own and that no major platform ships as a first-class feature as of mid-2026. You will demand them, build them, or work around them.

ONE

Interruption budget router with a priority taxonomy. Every platform ships an interruption mechanism. None ships the economics layer above it: a cap on how many approval events a supervisor receives per hour, a classification by priority, and routing against the cap.

TWO

Background failure detection. A primitive that correlates semantic success with an actual state change in the target system, so an agent cannot mark work complete that never happened, is not shipped as a named feature anywhere. This is the invisible-action failure, and the industry has no answer to it yet.

THREE

Source-document supersession detection. Retrieval is universal. Detection that fires re-validation when an underlying source changes is not, which is precisely the failure that put a care-coordination agent on a three-year-old clinical guideline while every checkmark stayed green.

FOUR

Context sufficiency validation as a typed pre-build gate. Retrieval that returns something is universal. An assertion that the relational and governance context is actually sufficient before the agent acts is not.

FIVE

Shadow workflow prevalence measurement. Detecting users running parallel manual processes alongside the agent requires correlating agent usage with system-of-record activity. No observability vendor integrates the two, which means the clearest signal that your supervisory design failed is the one nobody can see.

SIX

Real-time kill switch upstream of action execution. Most platforms ship monitoring and alerting downstream of the action. Two large clouds come closest with budget policies that can deny further calls at an account threshold. Nothing ships a per-agent, per-window gate evaluated before the call executes.

SEVEN

Retirement workflow. Something that simultaneously preserves the audit trail, blocks new invocations, and routes orphaned approval events. The components exist via identity deprovisioning and audit retention. The composite does not.

EIGHT

Affected-person audit view. A person-centric query surface over agent decisions, for right-of-access requests and for supervisory operations. Enterprise audit products approximate it. None is agent-native, and the people who need it most are the ones who are not your users.

Two things follow from that list. It is where the discipline is ahead of the field, which means it is where your own education should concentrate, because these are the capabilities that will be table stakes in eighteen months and are differentiating today. And it is a dated snapshot rather than a permanent gap. This is exactly the kind of capability that closes fast once a vendor decides to ship it, so re-run the list before you build anything on it.

PART SIX

Taking this into an evaluation

Six questions, in the order that separates a real platform from a good deck.

- For each capability on the page, which of the five categories is it? Ask the vendor to answer in those terms and watch whether they can.
- Which events do you emit that a base OpenTelemetry implementation does not?
- Which of the six observation instruments ship as named metrics, and which would we compose?
- Can you show per-task cost at the trajectory level, including coordination overhead? If not, what would we build to get it?

- Is there a gate that evaluates a call against a per-agent budget before it executes, or only alerting after?
- When a model version changes underneath us, what do we get: notice, a pin, a diff, or an email after the fact?

Then price the answers. Every capability that lands on the derivation line is a small engineering task nobody has scoped. Every one on the gap line is a decision to build, to demand it contractually, or to accept the exposure deliberately. Any of those three is defensible. Discovering it after launch is not.

THE ONE LINE

The platform sells you mechanisms. It cannot sell you the obligation.

Read the page for category rather than for capability, and most vendor arguments resolve in a sentence. What ships is real and you should consume it. What you compose is a scoping task nobody wrote down. What your team writes and your organization enforces was never on the page. And the handful of things nobody ships yet are the ones worth knowing by name, because they are the difference between a roadmap that works and a roadmap that assumed.

Sources and status. Drawn from the *Agentic AI for Product Leaders* series by Yoram Friedman, principally *Agentic AI for Busy Product Managers* (second edition), Appendix A, the platform taxonomy, together with the six observation instruments and the four-owner pre-launch readiness memo; *Why Agentic AI Products Fail*, chapter 9 on evals and chapter 10 on operational guardrails and the absence of a hard cost kill-switch; and *The Agentic AI Team* on who owns the composition work. The five categories, the four lines and the gap line are the author's taxonomy. The capability survey behind Parts two through five is stamped mid-2026 and is a snapshot rather than a standing claim; platform capability moves quarterly and the gap line closes fastest. Vendors are described by role or category rather than ranked, deliberately, because a ranking would date faster than the taxonomy and would be wrong in a different way for every reader. Security framework references are the OWASP Top 10 for Agentic Applications and the Cloud Security Alliance's MAESTRO. Figures 1 and 2 are structural maps rather than measured data.