WHITE PAPER

VENDOR EVALUATION

The Regulated-Automation Vendor Evaluation Kit

How to vet an AI or workflow-automation vendor before it touches regulated work — the questions, the evidence to demand, and a weighted scorecard you can defend to an auditor or a board.

For compliance, vendor-risk, procurement, and program owners · Updated August 2026

The short version

Vetting an automation vendor for regulated work is not a check of whether they "have security." It is a test of whether the platform can prove — with evidence, by design — three things: that it cannot take an action it can't undo, that it cannot invent an approval no human gave, and that it cannot lose the record of what it did.

Most vendor assessments never get there. They collect a security questionnaire, skim a certification logo, and move on — measuring whether the vendor is a competent software company, not whether the *automation* is safe to point at your regulated processes. This paper is the buyer's side of the harder test: how to scope the review to the risk, the questions that separate a real control from a claim, the evidence to demand instead of assurances, the six capabilities that actually make an automation platform safe, and a scorecard that turns all of it into a decision you can put your name on.

1. Scope the risk before you scope the vendor

The depth of your review should match what the automation can actually do on your behalf — not the vendor's size, brand, or how polished the demo is. A tool that only drafts text for a person to send carries a fraction of the risk of one that writes to a system of record unattended. Decide the tier first, because it sets how hard you push on everything that follows.

What the automation can do	Review depth	What to insist on
Read-only — reporting, or drafting for a human to send	Lightweight screen	Data-handling and access terms
Assists a human, who commits the action	Full assessment + evidence review	+ Audit trail and clear human-decision points
Executes unattended against a system of record	Full assessment + live control walkthrough	+ Demonstrated undo, fail-closed behaviour, per-workload isolation

Then do two things before the first vendor call. First, map the data actually in play — personal, health, financial, or records under a retention obligation — to the specific workflows the tool will run; a vendor can be perfectly fine for one workflow and unacceptable for the next. Second, write your own list of which actions must be reversible and which must be human-confirmed, and bring it. If you don't, the vendor's defaults will quietly define your risk posture for you. Finally, name the internal owners — security, privacy, procurement, and the business process owner — and record who signs off on whatever residual risk is left, so the decision is defensible later rather than orphaned.

2. The questions that matter — and how to read the answers

Phrase every question so a vague answer is visibly non-responsive. "We follow industry best practices" and "bank-grade encryption" are not answers — they are the absence of one. A strong answer names the control, its scope, and the evidence behind it. Group the questions by what they protect.

Data handling

Ask	A strong answer	A weak answer
Is our data used to train or improve any model?	A written no-secondary-use commitment, with retention and deletion terms	“Only to improve your experience,” or silence
Where is our data processed and stored?	Named regions, with a data-residency statement you can hold them to	“The cloud,” or a shrug toward a hyperscaler
What happens to our data when we leave?	Defined export format and a deletion timeline, in the contract	“Contact support”

Isolation & access

Ask	A strong answer	A weak answer
How is our data separated from other customers?	Per-workload / per-tenant isolation they can describe concretely	“Logical separation” with no detail
How do your own staff reach our data?	Named, time-boxed, logged access on a least-privilege basis	A shared admin account; “our team is trusted”

Automation authority

Ask	A strong answer	A weak answer
What can the system do with no human in the loop?	A named list of gated action classes, and who can confirm each	A description of the happy path only
Can any action it takes be undone — and has the undo been tested?	A per-action reversal it can demonstrate on request	“We can restore from backup”
Can the system approve its own actions?	No — an approval requires a real human act it cannot generate itself	Unclear, or “the workflow approves it”
What happens when a check or permission does not line up?	It stops and escalates	“It retries,” or “it proceeds with a default”

Incident & change

Ask	A strong answer	A weak answer
Can you or your staff edit or delete the activity log?	No — the record is append-only	“Admins can tidy it up”
What is your notification window when something goes wrong?	A specific, time-boxed commitment in the contract	“We’ll let you know”
What triggers a re-review of an automation?	A model change, a workflow change, a permission change, or a change in the rules that govern it	“It just runs”

For every answer, capture the answer *and* the artifact that backs it. An unbacked “yes” scores the same as a “no.”

3. Evidence, not assurances

A self-completed security questionnaire tells you what the vendor says about itself. Independent evidence tells you what someone else tested. Ask for both and read the evidence critically — the value is in the detail, not the logo. Match every claim to the artifact that would prove it.

Claim	Ask for	What weakens it
We are independently audited	A current control-attestation report (SOC 2 Type II or equivalent) — then check its date, the period it covers, and that *your* product is in scope	A years-old report, or one that covers a different service line
Your data stays in your region	An architecture and data-residency statement in writing	A verbal assurance only
We don’t train on your data	The commitment written into the contract, not the marketing page	A sentence on a web page that can change without notice
We build on trusted infrastructure	The list of sub-processors and their own attestations	Un-named dependencies “carved out” of the audit scope
Our AI is governed	Evidence that covers model behaviour, human-oversight design, and logging of automated decisions — not just cloud security	A cloud-security certificate offered as if it answers the AI question

> **Read the exceptions, not the logo**
>
> The useful part of an attestation report is the section listing what the auditor found and where controls fell short. Confirm the report actually tested controls relevant to *your* use, follow the sub-processor thread (what the vendor builds on is risk you inherit), and treat an unusually clean report as a prompt to ask more questions, not fewer. Put every artifact under NDA and log a date to re-request it — evidence expires.

4. The six capabilities to demand

Underneath the questions are six capabilities that separate a platform that is genuinely safe from one that merely intends to be careful. Ask the vendor to *demonstrate* each on a real action, not describe it in principle.

Undo by design

Every reversible action has a built-in, tested reversal, so there is always a way back.

Make them show it: pick one action and ask them to reverse it live.

Confirm before commit

Consequential or irreversible actions stop and wait for a named person before running.

Make them show it: ask which action classes are gated, and who can be the confirmer.

No self-approval

The system cannot manufacture a human sign-off; a real person has to act, so it can't authorise itself.

Make them show it: ask how the platform proves an approval came from a person, not from the automation.

A record that can't be rewritten

Every action and every undo is written to a log no one — including the vendor — can edit or delete.

Make them show it: ask who *can* alter the log. The right answer is "no one."

Stops when unsure

If a permission, check, or version doesn't line up, it halts and escalates instead of guessing.

Make them show it: ask what happens on ambiguity. The safe default is “stop,” not “proceed.”

Least access per task

Each function runs with only the permissions it needs, so a fault or leak in one path can’t reach another.

Make them show it: ask whether one compromised component exposes everything, or only its own slice.

5. Red flags that should slow or stop a deal

Warning signs

- Slogans in place of specifics — “best practices,” “military-grade,” “trust us.”
- Rollback described as a support process (“contact us and we’ll sort it out”) rather than a per-action feature.
- No clear answer on whether your data trains or improves a model.
- Staff access that can’t be named, scoped, or logged.
- An activity log the vendor is able to edit.
- Unattended execution with no stated stop condition.
- Sub-processors or model providers left out of the audit scope.

6. From answers to enforceable terms

Verbal answers in a sales call are not controls. The point where a “yes” becomes something you can rely on is the contract and the data-processing terms. The commitments most worth getting in writing are the ones this paper keeps returning to: no secondary use of your data, defined data residency and deletion, named sub-processors, a real breach-notification window, and the right to obtain current evidence on a cadence. Treat these as the terms that make the verbal answers real — and confirm the exact wording with your own counsel or authorising official before you sign. Nothing here is legal advice; it is the list of things to ask counsel to look for.

7. The weighted scorecard

Score each vendor on *evidence*, not on the answer alone. Weight the rows by what your workflows actually need — a read-only tool can carry a low score on "demonstrated undo"; an unattended one cannot. Score each row **0** if it is only asserted, **1** if it is documented, **2** if it is demonstrated.

Criterion	Weight	Score
No secondary use of your data, in writing	**High**	0 / 1 / 2
Data residency & deletion terms	**Med**	0 / 1 / 2
Tenant isolation + named, logged staff access	**High**	0 / 1 / 2
Per-action undo it can demonstrate	**High***	0 / 1 / 2
Confirm-before-commit on consequential actions	**High**	0 / 1 / 2
Cannot approve its own actions	**High**	0 / 1 / 2
Fails closed when something doesn't line up	**Med**	0 / 1 / 2
Append-only activity log	**High**	0 / 1 / 2
Least privilege per workload	**Med**	0 / 1 / 2
Independent attestation — in scope and current	**High**	0 / 1 / 2
Sub-processor transparency	**Med**	0 / 1 / 2
Contract: data terms + a real notification window	**Med**	0 / 1 / 2

*High only if the automation executes unattended. Total and compare — but treat any **0 on a High-weight row as a stop, not a deduction.** A vendor can have a strong total and still be disqualified by a single high-weight zero.

> **Worked example**
>
> Two vendors both score well overall. Vendor A scores 2 on every High row and 1 on the Mediums. Vendor B has a higher raw total but scores 0 on "append-only activity log" (their admins can edit it). Vendor B is out — not because of the total, but because a regulated process that can't produce an untouchable record of what happened is one you can't defend in an investigation, whatever else it does well.

How to run the evaluation

Set the tier first. Send the question bank ahead of a working session so answers arrive with evidence attached, not improvised on the call. In the session, ask the vendor to *demonstrate* the six capabilities on a real action rather than describe them. Score on what they showed, not what they claimed. Route the contract terms through counsel. Then keep the evidence under NDA with a date to refresh it — a vendor that passed last year may not pass today.

SG1 CONSULTING

sg1consulting.us · contact@sg1consulting.us