SG1 CONSULTING

WHITE PAPER

PUBLIC SECTOR

Government Regulatory Workflow Automation: An Evaluation Guide

How public-sector and regulated-industry buyers separate an authorization-ready records platform from one that will fail an audit — the axes to test, a weighted scorecard, the disqualifiers, and a proof that survives an inspector looking over your shoulder.

For records officers, security/authorization reviewers, procurement, and program owners · Updated August 2026

Start here: what a defensible evaluation actually tests

The failure you are guarding against is not a missing feature. It is standing in front of an auditor, a records officer, or a citizen who is questioning an automated action, and being unable to answer four plain questions: who did this, when, under what authority, and can you undo it. A feature comparison never surfaces that failure. A defensible evaluation is built to.

Fitness turns on four axes, and they have to be tested *together*: whether the platform sits at the right cloud-authorization level for the sensitivity of your data; whether it can honor records retention and disposition on a schedule you could defend; whether every automated action leaves a history you could neither lose nor alter; and whether consequential actions stop for a named human before they commit. A platform can be strong on three and disqualifying on the fourth — an elegant workflow builder that lets automation delete records on a schedule with no legal-hold override is unfit no matter how good its interface is.

> **Hard gate vs. weighted criterion**
>
> Throughout this guide, a **hard gate** is a question where a failing answer ends the evaluation — you do not score it against strengths elsewhere. A **weighted criterion** is scored and compared. Marking which is which up front is what stops a procurement team from spending three months piloting a platform that was never eligible.

Bring the right people before the first demo, not after: a records officer who owns retention and disposition, a security or authorization reviewer who can read a cloud-authorization claim critically, and a program owner who owns the decision an inspector would trace back — not just the team that liked the interface.

1. The authorization floor

Cloud-authorization programs exist to answer one question before a system touches sensitive government data: has an independent assessor verified this specific service, at this specific sensitivity level, inside a defined boundary. The two most common instruments a buyer will meet are the federal cloud-authorization program and its state-and-local counterpart run by a separate non-profit; where criminal-justice information is involved, a distinct security policy applies on top. Treat all of them as principle-level context to confirm with your own authorizing official — none of what follows is legal advice.

Match the level to the data, not to the marketing

Authorization is issued at an impact level tied to how bad a breach would be. In plain terms: a **Low** baseline is for public or non-sensitive information; **Moderate** is the common floor for controlled-but-unclassified operational records; **High** is for missions where a breach would be severe or catastrophic. Asking for High where Moderate fits wastes budget. Accepting Low where the data demands Moderate is the more dangerous error, because it fails quietly.

Authorization attaches to an offering, never to a company

An authorization covers a named service offering at a named impact level within a defined boundary — specific components, data flows, and subprocessors. “The company is authorized” tells you nothing about the module you would actually run. And authorization is a continuing obligation, not a one-time badge: ongoing vulnerability scanning, a live plan for open items and their remediation, and current

inventory reporting are all expected. Ask to see evidence the continuous-monitoring is current, not just that an authorization was granted once.

> **Federal and state paths are not symmetric**
>
> A federal authorization can generally be leveraged toward the state-and-local program through a reuse process; a state authorization carries no formal credit in the other direction. If your procurement rules recognize one program, a logo from the other is not a substitute — confirm the vendor's status in the exact program your rules name.

The table below maps the data you are automating to the minimum posture to demand and the evidence that proves it. It is a starting frame for a conversation with your authorizing official, not a compliance determination.

Data you are automating	Minimum posture to expect	Evidence to demand	Type
Public / non-sensitive information	Low baseline authorization	Named offering + level + boundary in writing	**Hard gate**
Controlled-but-unclassified operational records	Moderate baseline authorization	Named offering at Moderate; current continuous-monitoring evidence	**Hard gate**
High-consequence mission data (severe/catastrophic if breached)	High baseline authorization	Named offering at High; boundary + subprocessor list; monitoring evidence	**Hard gate**
State or local government records	Status in the state-and-local program your procurement recognizes	Membership/authorization in the named program (not the federal logo alone)	**Hard gate**
Criminal-justice information	Advanced/multi-factor authentication + a signed vendor security addendum	The signed addendum for any vendor touching the data; MFA in force	**Hard gate**
Export-controlled or residency-restricted data	Data-center location *and* who can access it from where	Location statement + access-control staffing you can hold them to	**Verify**

Residency is about access, not just storage: the defensible question is not only where the data sits but who can reach it and from where. Export-controlled staffing requirements are your responsibility to enforce, so pin them down explicitly rather than assuming the authorization covers them.

2. Records lifecycle fitness

Automation that accelerates a workflow while quietly breaking recordkeeping is a net loss, because the recordkeeping failure surfaces years later at the worst possible moment. Records-management principles — the kind reflected in public electronic-records guidance — translate into concrete platform behaviors you can demand and test. Confirm the specifics against the retention obligations that actually bind your organization.

Retention and disposition are properties of the record, not afterthoughts

Every record the platform creates, moves, or files should be bindable to an approved retention schedule, with a disposition action — transfer, delete, or preserve permanently — attached to a record class rather than set by hand after the fact. Disposition must then be a *supervised* event, not a silent background sweep: records that become eligible for destruction should surface for review and produce their own audit entry, and permanent records should be identifiable for transfer rather than aging into deletion. “The system auto-deleted it on schedule” is not a defense if you cannot show the record was eligible and the action was logged.

> **Test the legal hold directly — do not take it on faith**
>
> A legal hold or freeze must override every automated disposition. Place a hold on a record, then trigger a disposition the schedule would otherwise allow, and confirm the platform *refuses and escalates* rather than destroying held material. A platform whose automation can outrun a hold is disqualifying for regulated use — this is a hard gate, not a scored line.

Records must survive the platform

A record and its context — who created it, when, its provenance and version history — must survive format and storage change, and must be exportable intact. Records trapped in a proprietary shape you cannot migrate out of are a disposition-and-migration failure waiting to happen, and a lock-in your successor inherits. Access to

records must be controlled and searchable up to the moment of approved disposition, with unauthorized modification or destruction prevented by design. Recordkeeping integrity is a permission model, not a policy PDF.

Ask for the report the platform can produce unprompted. If it cannot show a records officer “here is everything currently held, everything disposed, and the basis for each,” the recordkeeping story is incomplete — regardless of how the workflow builder looks.

3. AI governance and human oversight

Where a platform uses AI, evaluate it as neither magic nor menace but as a component whose behavior you can govern. A widely referenced public-sector AI risk-management structure organizes oversight into four moves, which make a useful review lens even though the structure itself is voluntary and should be applied in consultation with your own policy owners:

- **Govern** — establish who is accountable and what the policy is, across every automated use.
- **Map** — understand exactly where and on what the AI acts within your process.
- **Measure** — test it against real-world performance and its failure modes, not a demo.
- **Manage** — turn those findings into monitoring, escalation, and a way to shut it off.

The higher bar for rights- and safety-affecting work

Where automated handling affects people's rights or safety — benefits, eligibility, enforcement, or records that determine an outcome — public-sector oversight expectations converge on a heightened standard: an impact assessment before deployment, testing in a real-world context, ongoing monitoring, a documented human role, plain-language public notice that AI is in use, and a route for an affected party to contest a decision. Name these to your own counsel or authorizing official as the bar to hold the platform to; do not treat this paragraph as the prescriptive requirement itself.

> **The two oversight rights that are the practical test**

> “Human in the loop” is easy to claim. Two capabilities make it real: a person can review an AI-influenced decision *before* it takes effect, and an affected party has a route to contest or appeal it afterward. A platform that cannot pause for review or surface a decision for appeal is not ready for rights-affecting work, whatever the deck says.

Keep a human genuinely accountable: the people supervising the automation must understand its limits, and the platform should make its inputs, its confidence, and its reasoning visible enough that supervision is real rather than a rubber stamp on whatever the system proposed.

4. How safety is actually enforced under the hood

Behind every one of the questions above is a mechanism, and the difference between a safe platform and a hopeful one is whether the mechanism exists in the engineering or only in the adjective. Ask the vendor to *demonstrate* each of the six below on a real action, not describe it in a slide. These are stated in plain language on purpose — do not accept a coined brand term in their place.

Undo by design

For every reversible action, a defined and tested reversal is built alongside it — the way back exists before the action is ever run, not improvised after something breaks.

Make them show it: name an action, then ask them to reverse it live.

Confirm before commit

Consequential or irreversible actions do not run unattended; they stop and wait for a named human to confirm. The default for “this cannot be taken back” is a halt, not a proceed.

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

No self-approval

The system cannot fabricate its own sign-off. A real human act is structurally required, so the automation can never invent an approval to unblock itself — the single most important guard against an agent approving its own step.

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

A record that can't be rewritten

Every action and every reversal is written to a history no one — including the vendor — can edit or delete. Permanence is what makes "who did what, when, under what authority" answerable months later.

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

Stops when unsure

When a permission, a check, or a version does not line up, the step halts and escalates instead of committing something it cannot take back. Ambiguity is a reason to stop, not to guess.

Make them show it: ask what happens on a mismatch. The safe default is "stop."

Least access per task

Each function runs with only the permissions it needs, on separate credentials, so a fault or a leaked secret in one workload cannot reach another. The blast radius of any single failure is bounded by design.

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

> **How the six fit together in one action**
>
> Trace a single automated action end to end and every mechanism should appear in sequence: a request arrives, the platform **checks permission and version and stops if anything is off**; if the action is reversible it **registers the tested undo** before running; if it is irreversible it **halts for a named human to confirm** — a confirmation the system cannot forge; only then does it execute, running with **only the access that task needs**; and the whole sequence, including any reversal, lands in a **record that cannot be rewritten**. If any link is missing, that is the link to probe.

5. The vendor evaluation scorecard

The instrument below converts everything above into a defensible, comparable score. Twelve criteria span the four axes; each is scored **0** (only asserted), **1** (documented), **2** (shown in config), or **3** (demonstrated live or in the pilot), and every row requires a note on what proof was actually seen — the score itself has to be auditable. Weight reflects consequence: audit permanence, hold-override, and human-confirm-on-irreversible carry the most weight because they are what an

inspector traces first. Hard-gate rows are flagged: a **0 on any gate fails the vendor outright**, regardless of total. Run it with two evaluators independently and reconcile — the reconciliation column is where a claim one reviewer took on trust and the other tested comes to light.

#	Criterion & the evidence to demand	Axis	Weight	Gate	Score
1	Cloud authorization at the impact level your data requires — named offering, level, and boundary in writing	Authorization	**High**	**Gate**	0–3
2	Continuous-monitoring is current — recent scan and open-item evidence, not a one-time certificate	Authorization	**High**	**—**	0–3
3	Every record bindable to an approved retention + disposition schedule — show the config on a record class	Records	**High**	**—**	0–3
4	Legal hold overrides all automated disposition — run the hold-then-dispose test live	Records	**Highest**	**Gate**	0–3
5	Disposition is supervised and logged, not a silent sweep — produce the disposition report	Records	**High**	**—**	0–3
6	Records + metadata exportable intact; migration without loss — export a record and inspect its context	Records	**Med**	**—**	0–3
7	Append-only audit history no one can edit or delete — ask who *can* alter it; answer must be no one	Audit & control	**Highest**	**Gate**	0–3
8	Consequential/irreversible actions halt for a named human — trigger one and confirm it stops	Audit & control	**Highest**	**Gate**	0–3
9	The system cannot generate its own approval — ask how it proves an approval came from a person	Audit & control	**High**	**Gate**	0–3
10	Least privilege per workload; no single all-powerful credential — ask if one compromise exposes everything	Audit & control	**Med**	**—**	0–3

#	Criterion & the evidence to demand	Axis	Weight	Gate	Score
11	Human review before an AI-influenced decision takes effect + an appeal route — show the pause and the appeal path	AI governance	**High**	**Gate***	0–3
12	Plain-language disclosure of where AI is used; impact assessment on record — show the notice and the assessment	AI governance	**Med**	**—**	0–3

*Row 11 is a hard gate only when the automation performs rights- or safety-affecting work. Reconcile the two independent scores row by row; where they diverge, the lower score usually means one reviewer tested what the other assumed. Treat any **0 on a gate row as a stop, not a deduction** — a vendor can post a strong total and still be disqualified by a single gate zero.

6. What disqualifies a platform

Some findings should end an evaluation early rather than cost a scored point, so scarce pilot time is not spent on a platform that was never eligible. They are ordered here from most to least fatal, each paired with the one question that confirms it.

Red flag	Why it disqualifies
Automation can dispose of records with no enforced hold override	Recordkeeping integrity cannot be bolted on later; a hold that automation can outrun is no hold

Red flag	Why it disqualifies
Audit trail is editable, purgeable, or reconstructed after the fact	If the history can be changed, it is not evidence — it cannot survive an inspection or a records request
Irreversible actions run unattended, or the system can approve itself	A platform that can manufacture its own sign-off has no real human control, whatever the workflow diagram shows
A single all-powerful credential behind every workload	One key that can read mail, move files, change users, and delete records turns any leak into a tenant-wide breach
Authorization claim dissolves under a question	A brand-level “we're authorized,” the wrong program's logo, or lapsed monitoring is an unverifiable claim — treat it as a no
Records cannot be exported intact	Proprietary formats trap your records inside the vendor — a migration-and-disposition failure your

Red flag	Why it disqualifies
	successor inherits

7. Running a defensible proof and sign-off

Turn the evaluation into a decision an inspector could trace. That means piloting the guarantees rather than the demo, deciding your own rollout by reversibility, and recording who owns the go-live before anyone asks.

Pilot the guarantees, not the demo

Script tests that force the escapes, then read the record. Place a hold and try to dispose. Trigger an irreversible action and confirm it halts for a named human. Attempt an action with a mismatched permission and confirm it fails closed. Then open the audit trail and confirm every one of those attempts was recorded and that the record cannot be altered. A guarantee you did not try to break is a guarantee you have not tested.

Baseline before you automate

Capture before go-live	Why it matters later
Current cycle time for the target workflow	Proves a real improvement and gives you the “before” if the automation is ever challenged
Current error / rework rate	Distinguishes genuine quality gains from a faster way to make the same mistakes
Current backlog / volume	Sets the scale the pilot has to hold up at, not just a happy-path sample

Decide your own rollout by reversibility

Apply the same principle the platform should enforce internally to your own deployment plan: expand automation freely where a tested undo exists, and keep a

human confirm on anything you could not take back.

The action is…	Reversible (tested undo exists)	Irreversible / consequential
Low sensitivity	Automate; monitor	Human confirm before commit
High sensitivity / rights-affecting	Automate with review + logged undo	Human confirm + review + appeal path; expand last

Sign-off packet and reassessment cadence

Name an accountable owner for the go-live decision and assemble the packet before you launch, so “who authorized this and on what basis” already has an answer. Set the reassessment cadence at sign-off rather than waiting for the next contract cycle.

> **Sign-off packet checklist**
>
> - Named accountable owner for the go-live decision.
> - Impact assessment for any rights- or safety-affecting use.
> - The plain-language notice language describing where AI is used.
> - The documented human-review and appeal path.
> - The reassessment trigger — a near-miss, a failed check, a change in the rules that govern the work — and a scheduled date to re-check continuous-monitoring evidence and retention schedules.

A vendor that passed last year may not pass today. Keep the evidence you were shown, under NDA, with a date to re-request it — authorization status, monitoring currency, and retention schedules all expire, and a defensible decision is one you can re-defend when they do.

This paper describes principles for evaluating automation platforms in regulated and public-sector environments. It is general information, not legal, compliance, security, or records-management advice; confirm specifics against the obligations that apply to your organization and with your own counsel, authorizing official, and records officer.