DOCUMENTATION PAGE

Governed Orchestration Buyer Decision

Governed Orchestration Buyer Decision

Buyer Decision

The concrete buyer question is whether governed orchestration provides a trustworthy, appropriately bounded way to run unattended work. The approved source boundary supports a narrower decision: continue a reversible evaluation of whether the documentation's provenance and evidence route can be traced without an unresolved disclosure concern. It does not establish that unattended work can run, that execution will be controlled or enforced, or that the product is ready for adoption. A buyer may therefore treat traceability as an evaluation criterion, but must keep the runtime decision open until separate authorized evidence addresses it. “Trustworthy” here means that the boundary and the evidence status are stated plainly; it is not a security, reliability, compliance, performance, availability, or customer-results guarantee. The appropriate outcome is a scoped go-forward for evidence review, not a purchase, deployment, approval, publication, promotion, scheduling, rollback, or release decision.

Decision in plain terms: yes, the approved evidence supports continuing a bounded unattended-work evaluation when “evaluation” means reviewing documentation provenance and evidence traceability. No, it does not support running unattended work, claiming that execution is governed, or adopting the product. That distinction is the stopping rule for this page.

What Governed Orchestration Changes

For this review, governed orchestration changes the decision frame rather than proving a new runtime capability. It makes the buyer ask what the sealed source boundary can support, what an evaluator can observe, and where the evidence must stop. The authority names exactly three logical source IDs and the source registry supplies the permitted public-safe reference fields for those IDs. That makes provenance and disclosure alignment reviewable. It does not turn source metadata into an execution attestation. The

same distinction applies to the reader check: the evaluator can inspect the documentation surface and record the command's actual result, while the result remains documentation evidence. It cannot be used here to infer unattended execution, policy enforcement, isolation, security, operational availability, pricing, customer benefit, or deployment status. The change for a buyer is a better-bounded question and a clear stopping rule: proceed with traceability evaluation only when the approved records align, and request new authorized evidence before making a broader runtime or adoption decision.

Evidence and Limits

The sealed authority is `sources/authority.json` at revision `2026-08-08`; it lists `source.naming-decision`, `source.documentation-boundary`, and `source.platform-delivery-contract`, and records `employee_may_modify` as `false`. The corresponding registry entries are the approved public-safe reference boundary. Their disclosed fields are source ID, locator, revision, digest, authority class, and disclosure value:

Source ID	Authority class	Revision	Disclosure
`source.naming-decision`	`canonical-platform-source`	`2026-08-02`	`public-safe`
`source.documentation-boundary`	`product-owned-public-safe-source`	`2026-08-04`	`public-safe`
`source.platform-delivery-contract`	`canonical-platform-source`	`agent-orch-main-e65a64fcf429`	`public-safe`

These records authorize disclosure of reference metadata, not disclosure of a private source corpus or runtime evidence. This page has no claim references, because the sealed inputs do not establish a capability claim. Nothing here reports deployment, publication, pricing, a customer outcome, a security guarantee, performance, availability, compliance, or successful unattended execution. The independent evaluator should run `python3 tools/validate_content.py` and record its actual exit status and observable output. This deterministic repository check is documentation evidence: it verifies the public page and registry/source contracts and the reproducible projection, but it is not a runtime attestation. A

missing, stale, conflicting, or non-public-safe record ends the supported traceability conclusion rather than being repaired with an inferred source or a stronger product statement.

Decision Path

1. Inspect `sources/authority.json`. Confirm revision `2026-08-08`, the exact three authority-listed IDs, and `employee_may_modify: false`.
2. Resolve every ID in `registries/sources.json`. Confirm that each record is present, its disclosed reference fields are readable, and its disclosure is `public-safe`. Do not copy or reconstruct a source corpus.
3. Compare this page with its entry in `registries/pages.json`: the page ID, title, public visibility, prospective-buyer audience, reading modes, empty claim references, and three source references must agree.
4. Run `python3 tools/validate_content.py` and record the actual exit status and observable output. Treat the deterministic result as documentation evidence, not a runtime attestation.
5. If the authority, source disclosures, page metadata, and evaluator record are aligned, answer yes to continuing the bounded unattended-work evaluation described above, limited to documentation provenance and evidence traceability. If any part is unresolved, record the concern and stop the traceability conclusion.

In either case, leave the decision to run unattended work open. A buyer still needs separately authorized evidence for execution behavior, control or enforcement, operational performance, security, availability, compliance, deployment, pricing, customer outcomes, and any adoption commitment. This path does not authorize a transition or imply that one has occurred.

Acceptance Checks

- **AC-1 — Authority and synchronization:** The reader verifies authority revision `2026-08-08`, exactly the three listed logical source IDs, `employee_may_modify: false`, and a matching `public-safe` record for each. The reader also confirms that this page and its registry entry agree on ID, title, visibility, audiences, reading modes, claim refs, and source refs.
- **AC-2 — Bounded evaluator evidence:** The reader runs `python3 tools/validate_content.py`, records its actual exit status and observable output, and identifies

the deterministic result as documentation evidence only. The reader does not infer unattended execution, enforcement, security, customer outcome, deployment, approval, or publication from it.

- **AC-3 — Buyer decision boundary:** When the approved source records are present, public-safe, and synchronized with the page, the reader answers yes to continuing a bounded unattended-work evaluation only when that evaluation is limited to documentation provenance and evidence traceability. The reader answers no to treating the records as support for running unattended work or broader adoption, keeps the unattended-runtime decision unresolved, and takes no deployment, publication, pricing, or other transition action.

Provenance: registry-listed (no candidate packet)
Registry Source: `content/public/governed-orchestration-buyer-decision.md`