Architecture Review — Arnold OS

Client: Transformation & Beyond AB (self-review) **Subject:** Arnold OS — the autonomous orchestration layer of the Mondmodel travel-intelligence platform **Engagement:** `4ebc59cf-c7b9-4887-a421-3f71c049179f` · **Review date:** 2026-07-19 **Produced by:** the three production flows of the Architecture Review kit — `agentic-maturity-assessment` v1, `spec-gap-analysis` v1, `decision-memo` v1 (all registry-active, eval-passed 2026-07-12, production label)

> **Why we reviewed ourselves.** A sample review of a fictional company proves nothing. This is a genuine Architecture Review of our own autonomous orchestrator, run with the same flows we sell — and it finds real gaps, because a review that finds no gaps is not a review. All present-tense claims are stated **as of 2026-07-19**.

Method attestations & evidence corpus

Attestations. The six-rung capability ladder was loaded from the flow's reference (`playbooks/flows/agentic-maturity-assessment/references/capability-ladder.md`) and frozen **before** any subject evidence was read: `ladder_frozen_before_evidence: true` . The gap-analysis requirement register was extracted from the named target baseline and frozen **before** current-state evidence was mapped: `register_frozen_before_evidence: true` .

Evidence corpus. Seven source documents from the subject's repository, cited below as SRC-01…SRC-07. Each was ingested into the engagement's memory store with its content hash; the full path→hash→entry-ID table is in the citation appendix.

ID	SOURCE (REPO-RELATIVE)	WHAT IT CARRIES
SRC-01	`CLAUDE.md`	System entrypoint: architecture, mandatory protocols, verification checklist, content inventory
SRC-02	`.claude/docs/architecture.md`	Five-layer architecture; agent roster with per-agent model tiers
SRC-03	`.claude/docs/operator-domain-map.md`	Lived operating snapshot: two skill audiences, overnight session shapes, trigger taxonomy
SRC-04	`.claude/docs/always-on-local-tier.md`	Always-on tier canon: routing rule, auth model, costs, tier policy, contingency, open items
SRC-05	`.claude/docs/loop-engineering.md`	Loop design discipline: 5 patterns, 5 in-code guardrails, pilot calibration
SRC-06	`.claude/skills/arnold-runtime/HEARTBEAT.md`	Watchdog runbook: 7 checks, binary criteria, alert routing
SRC-07	`.claude/skills/arnold-runtime/INBOX.md`	Operator-channel handler: live-query rule, decision buttons

Disclosed method adaptation (self-review). In a client engagement, every cited entry is `status='approved'` by a named human approver before a flow may read it. Here the seven entries remain **draft**: approval is deliberately a human-only act, and the reviewing agent cannot approve its own evidence. Each claim therefore cites the source document (path + SHA-256) *and* its draft entry ID. Two run-time verifications were also performed against the live repository (⊕ in the appendix).

Part A — The Agentic Maturity Map

Exec summary

- Arnold OS operates at **R4 — Supervised Autonomous Agents**, with every rung R0–R4 `met` on cited evidence: agents plan and execute bounded tasks end-to-end overnight (alert → diagnose → fix → pull request) under a named checkpoint owner, with a documented escalation path (SRC-03, SRC-06).
- The **climb frontier is R5 — Governed Multi-Agent Workflows**, placed `partial`: the governance *framework* exists in code and canon — an immutable audit ledger, a multi-model council, orchestrator workflows with retry caps and halt-on-escalate — but the operating mode is govern-by-approval, not govern-by-exception: every irreversible output still routes through one person's explicit review (SRC-01, SRC-02, SRC-05).
- **Most material climb blocker: single-operator checkpoint concentration.** Every human checkpoint in the system — PR review, publish gates, escalation targets — resolves to the same operator-founder. R5's bar ("humans govern by exception") cannot be met by adding more automation; it requires codified exception policies and a second assurance lane (SRC-03, SRC-05).
- The subject's own adoption economics are cited, not invented: measured autonomous-tier spend of **74.73 USD/month, ~37% of a flat 200 USD/month pool**, host-independent (SRC-04).
- The top-rung governance question — *who approves the agent's actions at each checkpoint, and what may proceed without approval?* — is flagged as the input to a **decision-speed-audit** follow-on. This Map does not answer it.

Overall placement

`CURRENT_RUNG`	`CLIMB_FRONTIER_RUNG`	`MOST_MATERIAL_BLOCKER`
R4	**R5**	Single-operator checkpoint concentration: all approval and escalation paths terminate in one person; no codified govern-by-exception policy or second assurance lane exists (SRC-03, SRC-05)

The Maturity Map

RUNG	CAPABILITY (FROZEN REFERENCE)	MAPPED SUBJECT TOOLS	PLACEMENT	EVIDENCE
R0 — Data & BI Consumption	Team consumes governed dashboards/reports to inform decisions; no AI authoring	Normalized content store with measured inventory; scheduled watchdog metrics; operator status surfaces	**met**	SRC-01, SRC-03, SRC-06
R1 — Assisted Authoring	Individuals use AI assistants to draft, summarize, translate, query inside existing tools; individual task-time gains	Interactive agent sessions; ~20% of the skill catalog is operator-interactive process commands	**met**	SRC-01, SRC-03

RUNG	CAPABILITY (FROZEN REFERENCE)	MAPPED SUBJECT TOOLS	PLACEMENT	EVIDENCE
R2 — Grounded Self-Service Q&A	Team interrogates its OWN governed data/knowledge in natural language, sourced answers	Graph-RAG over the platform's own relational store; cross-tool memory search; inbox status answers from live queries only	**met**	SRC-01, SRC-07
R3 — Supervised Task/Workflow Automation	AI executes multi-step workflows; human reviews each output before effect	Generator × critic skill pairs; overnight fix-mode sessions ending in "PR ready to review"; commit verification checklist	**met**	SRC-01, SRC-03, SRC-05
R4 — Supervised Autonomous Agents	Agent plans and executes a bounded task end-to-end under checkpoint supervision (approve/deny/escalate)	Autonomous overnight sessions (alert → diagnose → fix → PR); watchdog with escalation routing; named checkpoint owner	**met**	SRC-03, SRC-06
R5 — Governed Multi-Agent Workflows	Multiple agents coordinate under a governance framework; humans govern by exception	Multi-model council; immutable ledger; orchestrator fan-out with retry caps + halt-on-escalate	**partial**	SRC-01, SRC-02, SRC-04, SRC-05

Rung detail

R0 — met. The platform maintains a governed, normalized content store whose inventory is measured and dated (296 published hotels across 5 production cities in 4 locales, re-measured 2026-07-11; 428 hotels including bootstrap-stage cities — SRC-01). The watchdog consumes queue, failure, and freshness metrics on a fixed schedule (SRC-06); the operator consumes alert and status surfaces routinely (SRC-03). Regular use — the climbed-when bar — is evidenced.

R1 — met. Roughly 20% of the skill catalog is the operator's interactive layer: process commands, review commands, council-agent handles in regular use (SRC-03). The domain map documents assistant use displacing by-hand workflows (ad-hoc API queries and SQL replaced by invocable skills the same day the gap was named) — a credibly-reported individual task-time effect, cited to the subject's own record (SRC-03). No adoption percentage is invented for this rung; none exists in the corpus beyond the catalog-share observation, and we say so.

R2 — met, with a disclosed reading. The platform interrogates its own governed corpus in natural language as a production behavior: graph-RAG over the relational store and a cross-tool memory search reachable from four tool surfaces (SRC-01); the operator-channel handler is *required* to answer status questions from live database queries, never from memory files (SRC-07). Disclosure: the frozen ladder's bar reads "more than a single pilot user"; this is a solo-operator organisation in which the additional regular consumers are agent runtimes, not additional humans. We place `met` on the regular-production-use evidence and disclose the headcount reading rather than hide it.

R3 — met. The evaluator-optimizer pattern is the house idiom: generator skills paired with critic skills against binary criteria, and a commit-gating verification checklist (SRC-01, SRC-05). Overnight sessions terminate in "PR ready to review" — a human approves before anything takes effect (SRC-03). The loop-engineering guardrail "human checkpoint for irreversible actions" is mandated *in code*, not prose (SRC-05).

R4 — met. One lived overnight cycle documents four root autonomous sessions: internal error detection, external alerts, and a deliberate queue, each running alert → diagnose → fix → PR end-to-end without mid-task human input (SRC-03). The checkpoint owner is named (the operator reviews every PR); the escalation path is defined (watchdog alert routing with a critical lane that mandates human investigation and forbids auto-fix on audit-chain breach — SRC-06). This satisfies the climbed-when bar: bounded tasks, end-to-end execution, checkpoint supervision, named owner, named escalation.

R5 — partial. What exists: a five-layer governance architecture with an immutable SHA-256-chained ledger and a multi-model council for critical decisions (SRC-01, SRC-02); orchestrator fan-out workflows with retry caps and halt-on-escalate as the codified idiom (SRC-05). What is missing, named per the placement rules:

BLOCKING FACTOR	EVIDENCE
Checkpoint concentration — every approval resolves to one operator; govern-by-exception requires a policy stating which classes proceed without approval, and a second assurance lane for those that do not	SRC-03, SRC-05
Autonomous initiation deferred — goal-trigger firing was deliberately deferred and never built (three seeded triggers, all disabled, nothing polls them); coordination is scheduled batch, not autonomous multi-agent initiative	SRC-04
The exception surface does not exist yet — external alerts still reach the system through the operator's e-mail relay (2 of 5 documented sessions); the consolidated alert inbox the subject's own map names as the highest-value surface is unbuilt	SRC-03

Change path (R5), staged and qualitative: (1) codify the exception policy — which alert/action classes auto-proceed, which escalate, in code alongside the existing guardrails; (2) build the consolidated exception surface the domain map already identifies; (3) add a second assurance lane for irreversible actions (cross-model checking is already house practice in the council and critic patterns — extend it to publish gates); (4) pilot govern-by-exception on the lowest-risk loop class first, with the existing binary-criteria harness as the gate. No dates, durations, or effort figures are stated — none exist in evidence. **Change model used:** Kotter's 8-step (John Kotter, *Leading Change*, 1996; reframed in *Accelerate*, 2014) — **attributed generic model, labelled illustrative, not mandated**; its org-scale steps must be scaled to a solo-operator context, where "guiding coalition" realistically means cross-model assurance plus external review, not headcount.

Governance hand-off

- `top_rung_governance_question`**:** Who approves Arnold's actions at each checkpoint — and which checkpoints may lawfully and safely become exception-only? Today the answer to every instance is "the same one person," which is precisely where the question bites.
- `is_decision_speed_audit_input`**: true.** Approval-chain latency and decision-rights design are the diagnostic object of the decision-speed-audit flow, not this Map.
- **Note:** this Map flags the question and hands it off; recommending a sign-off change without that audit's regulatory check is the malpractice the sibling boundary exists to prevent.

Evidence gaps (Part A)

- **None at rung level** — all six rungs are touched by approved-corpus-equivalent evidence (no `unknown` placements this run).
- Standing disclosure: the corpus carries one lived overnight snapshot and one measured cost series; no longitudinal adoption KPI series exists. Thin evidence is reported as thin — it caps confidence downstream (see Part C).

Part B — The Spec-Gap Register

Target statement

FIELD	VALUE
`name`	**Arnold OS Governance Baseline** — the subject's own published operating canon
`source_type`	`architectural-standard` (client-internal baseline)
`source_ref`	SRC-01 (mandatory protocols, verification checklist) · SRC-04 (tier policy, named pending mitigations, continuity action) · SRC-05 (five loop guardrails) · SRC-06 (watchdog contract)

Disclosed narrowing. In a client engagement the target is typically an external standard or an interface contract. A self-review against one's own baseline answers a narrower question — *does the system keep its own promises?* — and inherits a known bias (a self-authored target can be self-flattering). Two properties limit that bias here: the register was frozen before evidence mapping, and the baseline itself contains dated, falsifiable commitments (pending mitigations with named tests; a continuity deadline), which is where the gaps below come from.

Exec summary

One-liner: the system keeps its data-integrity and supervision promises (ledger, checkpoints, watchdog) and is currently breaking its own resilience and continuity promises (auth failover unimplemented, hosting continuity unconfirmed 12 days before a hard date).

Material gaps, ranked (`material_gaps`):

#	ROW	WHY MATERIAL
1	AR-08 (continuity funding)	Dated exposure: provider credits expire 2026-07-31 — 12 days after this review. The baseline's own continuity action — confirm a valid payment method, or "complete-and-verify" a migration to an alternative host — is unconfirmed; an unfunded VM can be powered off, halting the entire autonomous tier (SRC-04).
2	AR-07 (auth resilience)	The baseline names its own single point of failure — "a stalled refresh halts Arnold silently" — and names the mitigation (standby key, failover wrapper, refresh heartbeat, forced-failure test) as pending. The exposure is live daily; the fix is specified but unimplemented (SRC-04).
3	AR-04 (guardrail coverage)	The five-guardrail rule is dated 2026-07-03; seven scheduled in-daemon jobs predate it, and no evidence records a retroactive five-guardrail audit across that population. An unaudited scheduled loop is the exact class the rule exists to bound (SRC-04, SRC-05).
4	AR-09 (pool-identity confirmation)	The baseline marks one billing question "load-bearing" for any scheduled-cloud migration and it remains unconfirmed — which blocks the zero-VM contingency from being exercisable on short notice and caps confidence in Part C (SRC-04).
5	AR-03 (verification enforcement)	The subagent-verification protocol is mandatory in prose but names no mechanical gate; adherence rests on agent and caller discipline. The protocol exists because confident-prose-without-source-join happened twice (SRC-01).
6	AR-06 (routing codification)	The tier-routing rule is a documented convention pending codification; convention-layer routing depends on every enqueuer having read the doc (SRC-04).

#	ROW	WHY MATERIAL
7	AR-10 (alert-degradation test)	`unknown` — no evidence touches whether the alert-dispatch degradation path has ever been exercised; an untested failure mode on the channel that reports failures (SRC-06).

The register

ID	REQUIREMENT (TARGET'S VOCABULARY)	CLASS	CURRENT STATE	VERDICT	EVIDENCE
AR-01	Every significant data mutation MUST be logged to the ledger via the single sanctioned appender; the linear projection is generated, never hand-edited	architectural	Per-entry Merkle-DAG store with one sanctioned append script; projection generated and gitignored; 389 entry files present ⊕	**met**	SRC-01, ⊕V1
AR-02	The ledger chain MUST verify under the canonical hash formula; historical entries frozen	architectural	Chain verification run **this review**: PASS — anchored tail verifies; 272 clean entries from the documented anchor forward; 47 pre-anchor issues documented-and-frozen, not enforced ⊕	**met**	SRC-01, ⊕V2
AR-03	Any subagent producing a legal, regulatory, financial, factual, or strategic memo MUST pass a verification step before being treated as authoritative; unverifiable load-bearing claims MUST be labelled uncited	architectural	Protocol documented with two forms (self-invoked / caller-side), tool-whitelist changes shipped, escape hatch defined. The target names **no mechanical gate**: nothing blocks an unverified memo from being treated as authoritative except discipline	**partial** — missing element: mechanical enforcement	SRC-01
AR-04	Every agentic loop MUST carry all five guardrails **in code, never prose**: binary exit condition, iteration cap, budget cap, sandbox isolation, human checkpoint	architectural	Discipline adopted 2026-07-03 with a NO-LOOP refusal verdict; pilot calibration shows guardrails in code for audited loop classes (60/60 price-QA verdicts, 16/16 first-cycle editorial pass, no caps hit). Seven scheduled in-daemon jobs predate adoption; no recorded retroactive audit	**partial** — missing element: coverage of the pre-existing scheduled population	SRC-04, SRC-05
AR-05	The watchdog MUST run quick hourly checks and deep twice-daily checks executing all 7 categories; an audit-chain breach alerts critical and is never auto-fixed	architectural	Runbook v2.0 defines both modes, all 7 checks, the no-auto-fix rule, structured output, and machine-checkable binary criteria; the watchdog is listed among the live scheduled in-daemon jobs	**met**	SRC-04, SRC-06

ID	REQUIREMENT (TARGET'S VOCABULARY)	CLASS	CURRENT STATE	VERDICT	EVIDENCE
AR-06	Tasks requiring filesystem, local CLIs, or host-bound secrets route to the always-on host at enqueue time; the rule codified in a routing-rules table	architectural	Rule documented and applied as a design convention; codification explicitly pending a future phase (the target says so itself)	**partial** — missing element: codification	SRC-04
AR-07	Auth resilience for the always-on daemon: standby platform API key, one-shot failover wrapper with failure reporting, heartbeat on the token-refresh path — implemented and forced-failure-tested	architectural	The target marks the entire mitigation **pending**: the OAuth path is kept alive by a 6-hourly refresh cron plus a terminal-multiplexer keepalive, and "a stalled refresh halts Arnold silently" (target's own words). Documented absence	**gap**	SRC-04
AR-08	Continuity of the always-on tier: hosting plan, price, and payment method confirmed before the 2026-07-31 credit expiry, OR a migration completed-and-verified	architectural	The baseline's own spec table holds TBD in every cost/payment field; the open-items list confirms still-TBD as of 2026-07-17. Documented absence, 12 days before the dated exposure	**gap**	SRC-04
AR-09	The pool-identity billing question MUST be confirmed with the vendor before migrating scheduled jobs to the scheduled-cloud tier	architectural	The target lists the question as open and labels it load-bearing; no confirmation exists in evidence	**gap**	SRC-04
AR-10	Alert-dispatch degradation (secrets tooling unavailable → scripts fail gracefully) is verified by test	architectural	no evidence in approved memory	**unknown**	—

Dual-lane note: this register contains no techno-legal rows, so every `regulatory_citation` is `null` by rule. In a client engagement under FI supervision, rows of that class routinely appear and carry two citations each — the technical clause *and* the legal instrument it discharges, both verified against primary sources.

Severity rationale & remediation candidates (non-met rows)

- **AR-03 · partial.** *Severity:* the failure class this protocol exists for — confident prose without a source-of-truth join — is documented as having occurred twice before the protocol's adoption; a prose-only mandate leaves recurrence to discipline under load. *Remediation candidate:* a delivery-side mechanical gate (a hook- or checklist-class check) that blocks "authoritative" handling of a memo lacking a verification footer; evidence of done = the gate plus a committed red test.
- **AR-04 · partial.** *Severity:* the target's key principle ("no binary, verifiable exit condition → no loop") exists because unbounded agentic loops are a documented runaway-cost class; an unaudited scheduled job is exactly where such a loop hides. *Remediation candidate:* run the existing loop-classification audit across the seven-job scheduled inventory and record a per-job verdict; evidence of done = a dated audit table in canon.
- **AR-06 · partial.** *Severity:* convention-layer routing fails silently — a task enqueued to the wrong tier lacks its filesystem/secrets and fails there, not at enqueue. *Remediation candidate:* the codification the target

already plans (a routing-rules table applied at enqueue), plus an enqueue-time check.

- **AR-07 · gap.** *Severity:* single point of failure on the tier that runs everything else; failure mode is silent by the target's own description; the watchdog runs *inside* the same daemon whose halt it would need to report (SRC-04, SRC-06). *Remediation candidate:* implement the already-specified standby credential + failover wrapper + refresh heartbeat, then force the failure in a test and keep the record.
- **AR-08 · gap.** *Severity:* hard-dated: after 2026-07-31 the hosting account moves to standard billing against whatever payment method is on file; the baseline states an unfunded VM can be powered off — taking the autonomous tier with it. *Remediation candidate:* the funding decision (Part C) executed and verified before the date; evidence of done = confirmed billing state or a completed-and-verified migration.
- **AR-09 · gap.** *Severity:* the zero-VM contingency is only as real as this answer; if scheduled-cloud runs draw from the interactive pool, the contingency's economics change materially. *Remediation candidate:* obtain the vendor's written answer; record it in canon next to the tier policy.
- **AR-10 · unknown.** *Severity:* an untested graceful-degradation path on the alerting channel is a silent-failure mode at the worst location — the component that reports failures. *Remediation candidate:* forced-failure test of both dispatch scripts with secrets tooling absent; record the observed behavior.

Evidence gaps (Part B)

`[AR-10]` — the sole `unknown` row. No abstained input classes: all three required classes (target spec, current state, profile) were present.

Part C — The Decision Memo

Decision statement

FIELD	VALUE
`decision`	Decide, before the credit expiry, the hosting and funding model for the always-on tier: **(A)** keep the current small always-on VM on standard billing, **(B)** migrate the daemon to an equivalent low-cost always-on host, or **(C)** exercise the zero-VM contingency (migrate the scheduled workload to vendor-hosted scheduled runs).
`decision_owner`	The operator-founder, Transformation & Beyond AB (SRC-01 — the platform is operated by T&B AB; SRC-03 — single-operator model)
`deadline`	**2026-07-31** — provider credits expire; the account then bills against the payment method on file (SRC-04)
`why_board_level`	Continuity of the entire autonomous tier rests on one VM — the baseline warns the host "can be powered off" if billing fails; the choice moves spend between meter classes (a flat, already-paid pool vs. ~+75 USD/month of metered spend); and a migration must be "complete-and-verify", not a reversible toggle (SRC-04)
`citations`	SRC-01, SRC-03, SRC-04

Options

Option A — Keep one small always-on VM (confirm standard billing)

The baseline's own continuity action, branch one.

Evidence for:

- The daemon's compute is host-independent and rides an already-paid flat pool: measured 74.73 USD/month against a 200 USD/month allotment (~37%) — the VM is "the cheap meter, not the architecture" (memory lane · SRC-04).
- A daemon/cron/workflow box is typically a small VM in the 12–24 USD/month band per the baseline's own estimate (memory lane · SRC-04).
- The tier decision leaves the architecture unchanged — the baseline calls it "a funding decision on one always-on VM, not an Arnold re-design" (memory lane · SRC-04).

Evidence against:

- The single-host fragility class is documented and unmitigated: token-refresh cron + terminal-multiplexer keepalive as a single point of failure with silent-halt behavior (AR-07 gap) (memory lane · SRC-04).
- The subject's own facts about the host are unverified: plan, price, and payment method all TBD; the live cron inventory has never been confirmed against the box (memory lane · SRC-04).
- The auth posture (subscription OAuth for unattended automation) is recorded as *tolerated, not blessed* — a standing policy exposure (memory lane · SRC-04).

Assumptions: the vendor's tolerated position on subscription-OAuth automation continues (assumption, not evidence); a suitable VM remains available in the documented price band (assumption pending AR-08 verification).

Option B — Migrate to an equivalent always-on host

Evidence for:

- Compute cost is host-independent — the flat-pool economics survive any always-on host (memory lane · SRC-04).
- Removes the expiring-credit dependency while preserving the architecture (memory lane · SRC-04).

Evidence against:

- The baseline requires a migration be "complete-and-verify", a real undertaking for a solo operator (memory lane · SRC-03, SRC-04).
- The fragility class migrates with it: a new box still needs the token-refresh cron, keepalive, deploy path, and service management the zero-VM path would delete (memory lane · SRC-04).

Assumptions: an equivalent host lands in the same price band (assumption; no quote in evidence).

Option C — Zero-VM: migrate scheduled workload to vendor-hosted scheduled runs

The baseline's fully-specified contingency.

Evidence for:

- Deletes the fragility class wholesale: token-refresh cron, keepalive, deploy scripts, and service management are "eliminated entirely" in the baseline's own migration table (memory lane · SRC-04).
- The live workload is "almost entirely Tier-2-shaped scheduled batch": the daemon's seven scheduled jobs and the daily pipeline; the only sub-minute consumers are operator-present conveniences (memory lane · SRC-04).
- The migration path is fully specified, job by job, in canon (memory lane · SRC-04).

Evidence against:

- Cost delta ~ +75 USD/month: pool-covered usage becomes metered spend — to eliminate a VM in the 12–24 USD/month band (memory lane · SRC-04).
- The load-bearing billing question (whether scheduled-cloud runs share the flat pool) is unconfirmed (AR-09 gap) (memory lane · SRC-04).
- Loses sub-minute queue pickup — an operator-present convenience, but a real one (memory lane · SRC-04).

Assumptions: the specified migration performs as documented (assumption — the path is specified but unexercised).

Recommendation

FIELD	VALUE
`option_id`	**A** — keep one small always-on VM; confirm billing now; execute the already-specified auth-resilience mitigation (AR-07) immediately after
`confidence`	**medium**
`confidence_reason`	The economics lane is strong and effectively one-sided: a box in the documented 12–24 USD/month band preserves ~75 USD/month of pool-covered compute, and the tier decision is explicitly not an architecture decision. But confidence is capped: material unknowns sit on Option A's own facts (AR-08: price/payment unverified; live host state unconfirmed), the fragility exposure is real until AR-07 ships, and the corpus carries no longitudinal reliability record. Per the rubric, real-but-incomplete support with a material unknown is `medium`, not `high`.

FIELD	VALUE
`flip_conditions`	(1) The vendor ends its tolerated position on subscription-OAuth unattended automation — the baseline already names zero-VM as that contingency; flip to C. (2) The pool-identity question (AR-09) resolves such that scheduled-cloud runs draw from the same flat pool — C's cost objection collapses; re-decide. (3) Confirmed standard-billing cost for a suitable VM materially exceeds the documented 12–24 USD/month band — re-score B and C. (4) The AR-07 mitigation cannot be shipped and forced-failure-tested, or a silent-halt incident recurs after it ships — fragility then outweighs meter economics; flip toward C.

Consequences & risks

STATEMENT	AGAINST REC?	CITATION
Steelman for C: choosing A *preserves* the documented fragility class rather than deleting it. The strongest argument against our recommendation is structural: the daemon's failure mode is silent, and the watchdog that would report a halt runs inside the same daemon — self-monitoring cannot see its own death. C removes the cron, the keepalive, and the silent-halt mode wholesale; A merely promises to mitigate them.	**true**	SRC-04, SRC-06
Choosing A keeps the ~75 USD/month delta unspent in the pool — capacity that funds the AR-07 mitigation work instead of vendor metering.	false	SRC-04
A does not defer the deadline: the payment-method confirmation (AR-08) is required *now* regardless — choosing the status quo without funding it is choosing a power-off.	false	SRC-04
Choosing A does not burn C: the zero-VM path is specified in canon and remains exercisable if a flip condition fires.	false	SRC-04

Regulatory implications

None recorded — and that is a read, not an omission. The corpus was read for regulatory scope; this infrastructure decision carries none in evidence. (Contrast: in a client engagement under FI supervision this section routinely carries verified citations to primary legal text, per the flow's verification protocol.)

Advisory judgment

The following sentences are inference, not cited fact — placed here by the two-lane rule. A solo-operator system's scarcest resource is operator attention, and the zero-VM path's true price is a migration project competing with revenue work for that attention; the VM's true price is carrying a known, currently-unmitigated fragility for another cycle. On the evidence, the funded-VM-plus-mitigation course dominates until a flip condition fires — but it only dominates if AR-07 actually ships; executed lazily, Option A is the steelman's scenario.

Evidence gaps (Part C)

- Hosting plan/price/payment facts (TBD in the corpus — AR-08).
- Live host cron inventory (never verified against the box).
- The pool-identity billing answer (AR-09).
- Any longitudinal availability/incident record for the always-on tier (none in corpus; the review relies on the baseline's own characterizations).

Verification footer

Sources checked: the seven-document corpus (paths + SHA-256 digests in the appendix; ingested as engagement memory entries) · two run-time repository verifications (⊕V1 entry-file count; ⊕V2 chain verification PASS, 2026-07-19) · flow registry state for the three producing flows (read from the live registry 2026-07-19: v1, active, eval-passed 2026-07-12, production label).

Disclaimer (mandatory): This memo is neither personalized investment advice nor legal advice. It supports the subject organisation's decision-makers and their advisors and does not replace them; external counsel confirms any regulatory interpretation before this memo drives a filing or a binding commitment.

Advise agenda crosswalk

The four questions the Architecture Review is sold to answer, answered here on ourselves:

QUESTION	WHERE THIS REVIEW ANSWERS IT
Which tasks to automate	Part A: everything through R4 is automated and evidenced; the frontier finding says the *next* automation is not another generator but the exception surface (consolidated alert inbox) and a second assurance lane. Part B AR-04 bounds *how*: no loop without its five guardrails in code.
When open-source models suffice	**Not assessed for this subject** — the corpus records per-task model *tiering* (small models for hourly checks, larger for deep reasoning — SRC-02, SRC-06) but contains no open-weight evaluation, so this review honestly abstains. In a client run this enters through the capability-inventory intake.
How the economics work	Measured, cited, public: 74.73 USD/month of autonomous compute (~37% of a flat 200 USD/month pool), host-independent; the hosting box is a 12–24 USD/month decision separated cleanly from the compute meter (SRC-04, Part C).
How to hedge token cost	Part C is the hedge, demonstrated: keep compute on a flat already-paid pool; hold a fully-specified migration path to metered scheduled-cloud as the contingency; know the exact flip conditions under which the hedge inverts. Model tiering per task (SRC-02, SRC-06) is the second lever.

Citation appendix

Technical sources (memory lane — self-review adaptation disclosed):

ID	SOURCE & VERIFICATION
SRC-01	`CLAUDE.md` · SHA-256 `cbde8098e50e12b67c41a623eed752ee891fe02ad27e3acd1612ed1df3a6df16` · engagement memory entry `2b81b560-5da6-4a50-88ba-6b4ba4a28ff5` (draft — human approval is an open operator gate, disclosed above)
SRC-02	`.claude/docs/architecture.md` · SHA-256 `1791de057f71673271a6ed43938a6c67f1aedb5419b87d97ea2f5194cc83724c` · entry `cc2fa0fc-6555-4ecb-8110-43b6cc2a240f` (draft)
SRC-03	`.claude/docs/operator-domain-map.md` · SHA-256 `dfe3266d2b8580a8a8f1b2b6df2d88f8e4ef03e5e19c8bf1450186159db2d00d` · entry `8ac858b4-cfce-4340-b9c8-66d03015c705` (draft)
SRC-04	`.claude/docs/always-on-local-tier.md` · SHA-256 `39f381b52112fa6482f5cae87acb05a35a70f02da4fb8cbc806140c04a6534b7` · entry `51a8ad86-8fca-4f70-b1fc-81ec3f1f053b` (draft)

ID	SOURCE & VERIFICATION
SRC-05	`.claude/docs/loop-engineering.md` · SHA-256 `d72cffe65672fd092bccef5b886fd3322dc40667d2dd7089c62f41912e8fe365` · entry `d1752a1c-6936-46a9-8159-adc6fa9a8d30` (draft)
SRC-06	`.claude/skills/arnold-runtime/HEARTBEAT.md` · SHA-256 `440e670ca3d4c3bc00493f10299bc5a98aee310b2437dc9689dba6ea8528c1f5` · entry `62b368f1-d5c9-4fc1-9227-302729ec374f` (draft)
SRC-07	`.claude/skills/arnold-runtime/INBOX.md` · SHA-256 `b6b5a8abf1c9a81567f23f310ddbcfa35c3cc34078ce58bec72b0f4d7f74b53e` · entry `a77243f6-ef4b-41bb-b069-efe19472f31f` (draft)
⊕V1	Run-time verification, 2026-07-19: 389 ledger entry files counted in the per-entry store
⊕V2	Run-time verification, 2026-07-19: canonical chain verification script → **PASS** (anchored tail verifies; 272 clean entries from the documented anchor; 47 pre-anchor issues documented-and-frozen, not enforced)
Ladder	`playbooks/flows/agentic-maturity-assessment/references/capability-ladder.md` — Dignami's vendor-neutral method, authoring-verified 2026-07-12
Change model	Kotter 8-step — John Kotter, *Leading Change* (1996); *Accelerate* (2014). Attributed generic model, labelled illustrative (verified against the flow reference, which verified against Kotter Inc.'s published methodology 2026-07-12)

Regulatory sources (external lane): *empty — no techno-legal register rows and no regulatory scope in evidence for the Part C decision; the lane is empty by honest absence, not by omission.*

Colophon

- **Deliverable:** `offers/architecture-review/sample-deliverable.md` — the SHA-256 of the final committed file is printed in the PDF rendering of this colophon and recorded on the engagement's append-only audit chain (`deliverable_generated` event, engagement `4ebc59cf-c7b9-4887-a421-3f71c049179f`).
- **Date:** 2026-07-19.
- **Produced with the same flows we sell:** `agentic-maturity-assessment` v1 · `spec-gap-analysis` v1 · `decision-memo` v1 — registry-active, eval-passed 2026-07-12, production-labelled (registry state read-verified this run).
- **Honesty register for this sample:** (1) evidence entries are draft, pending the human-approval gate — disclosed in Method; (2) flow *runs* for this self-review could not be recorded through the sanctioned agent interface (run creation is a client-application lane), so the production record of this sample is the audit-chain deliverable event plus the engagement's memory entries — the gap is reported, not papered over; (3) the gap register grades the subject against its own baseline — a disclosed narrowing (Part B).

Transformation & Beyond AB · auditable AI operations.

DELIVERABLE HASH — SHA-256 OF `SAMPLE-DELIVERABLE.MD` AS COMMITTED

`be39c1445d48c89f7220525629218794c9d6f36c045677a6c630fcee7070fdb7`

Rendered 2026-07-19 · Produced with the same flows we sell.