Pillar #4 — Skills

Progressive-disclosure audit: 79 skills inventoried, L0-L3 rubric, top-20 refactor priority, PoC task-postflight (541→194 LOC). MC #99131 | 2026-05-04

Audit Summary

Pillar #4 Skills Audit — Summary

Source: `~/system/specs/agentic-os-pillar4-skills-audit-2026-05-04.md` (§1–§3 + Reality Anchor)
MC: #99131 | #99176
Date: 2026-05-05
Phase: DESIGN + PoC (Phase 2)

Executive Summary

This audit covers D2 (Top-20 Refactor Priority Table), D3 (Progressive Disclosure Design Pattern), and D4 PoC analysis for the task-postflight skill refactor.

Key findings:

- 79 active skill directories on disk; 94 rows in skill-registry.db (32 phantoms, 17 unregistered)
- Only 15 skills have any log invocations in the 19-day measurement window
- `mehanik` (186 hits) and `update-config` (1 hit) appear in logs but have no disk directory — ghost invocations
- 9 skills with references/ dir; 70 are monolithic (L0/L1)
- 12 TOB skills have nested structure — invisible to Claude Code flat-discovery loader
- Highest-priority refactor target: `task-postflight` (5,367 tokens × 21 measured invocations = priority_score 82.05)
- **Reality anchor:** At current ALAI scale (Claude Max flat-rate subscription), context-bloat incremental cost is approximately $0-2/month. The value of this audit is context window capacity management, not dollar cost reduction.

Environment Constants

```
WORKING_DIR=/Users/makinja
SKILLS_ROOT=/Users/makinja/.claude/skills
TELEMETRY_LOG=/Users/makinja/system/logs/skill-use.log
REGISTRY_DB=/Users/makinja/system/databases/skill-registry.db
```

```
TOKENIZATION_FORMULA=bytes_div_3.7
PRICE_USD_PER_MTOK_INPUT=3.00
SESSIONS_PER_MONTH_BASELINE=600
TELEMETRY_WINDOW_DAYS=19
OUTPUT_DIR=/tmp/pillar4-99131-out/
```

Inventory Summary

Metric	Value	Source
Active skill dirs on disk	79	`ls ~/.claude/skills/ | grep -v _archived | wc -l`
Archived skills	32	`ls ~/.claude/skills/_archived/ | wc -l`
skill-registry.db rows	94	`sqlite3 skill-registry.db 'SELECT COUNT(*) FROM skills;'`
DB-only phantoms	32	comm comparison
Disk-only unregistered	17	comm comparison
Skills with references/ dir	9	find query
Skills with invocations in window	15	log grep
Measurement window	19 days	2026-04-16 to 2026-05-05
Total invocations in window	267	awk filter
Ghost invocations (mehanik, no disk)	186	log grep — mehanik not on disk

---

Appendix A — Inventory CSV (83 rows)

Source: `~/system/specs/agentic-os-pillar4-skills-inventory.csv`

83 total lines: 3 comments + 1 header + 79 data rows, 19 columns, RFC-4180 compliant

Methodology:

- **Tokenization formula:** `skill_md_tokens_est = skill_md_bytes / 3.7` (GPT-4o empirical average for English markdown, ±15% error)
- **Telemetry window:** 19 days (2026-04-16 to 2026-05-05) — `invocations_30d` is lower bound, not exact 30-day count
- **Sessions per month baseline:** 600
- **Price per Mtok (input):** $3.00

Key findings from CSV:

- **32 phantom rows:** Exist in skill-registry.db but no corresponding directory on disk (e.g., `algorithmic-art`, `brand-guidelines`, `sentry-*`, `tob-*` variants)
- **17 unregistered:** Disk skills NOT in skill-registry.db (`ask-board`, `prompt-forge`, `task-postflight`, `lightrag-*`, etc.)
- **Ghost invocations:** `mehanik` (186 hits), `update-config` (1 hit) — no disk directory exists
- **TOB nested structure:** All 12 `tob-*` skills have `README.md + skills/` subdir at root; no `SKILL.md` — invisible to flat-loader

→ See Top-20 Refactor Priority Table

Appendix B — Inventory README

Source: `~/system/specs/agentic-os-pillar4-skills-inventory.README.md`

Frequency Sources

Two independent sources per spec §CONSTRAINTS #4:

1. **skill-use.log (PRIMARY)** — `/Users/makinja/system/logs/skill-use.log`
   - 300 total entries
   - 267 entries within 30d window (2026-04-04 to 2026-05-04)
   - Hook fires on `SKILL=<name>` log line
2. **skill-registry.db (SECONDARY)** — `/Users/makinja/system/databases/skill-registry.db`
   - Query: `SELECT name, use_count FROM skills ORDER BY use_count DESC;`
   - 94 rows in registry

Hook Coverage Gap

- Hook fires only when `tool_name="Skill"` (i.e., the `/command` invocation)
- **Skills invoked by sub-agents as inline tool calls (not via Skill tool) are NOT counted**
- **Skills triggered via `references/` file Read operations are NOT counted**
- `invocations_30d` column = **minimum lower bound**, not exact count

Monthly Cost Assumptions

```
SESSIONS_PER_MONTH_BASELINE = 600
PRICE_USD_PER_MTOK_INPUT = 3.00
```

```
est_monthly_context_tokens = (frontmatter_description_bytes / 3.7) * 600
est_monthly_invocation_tokens = skill_md_tokens_est * invocations_30d
est_monthly_cost_usd = (est_monthly_context_tokens + est_monthly_invocation_tokens) * 3.00 /
1,000,000
```

Reality anchor: At current ALAI scale (Claude Max flat-rate subscription), context-bloat incremental cost is approximately $0-2/month. The value of this audit is context window capacity management, not dollar cost reduction.

Cross-links:

- Top-20 Refactor Priority Table
- Progressive Disclosure Design Pattern
- PoC: task-postflight Refactor

Top-20 Priority

Top-20 Refactor Priority Table

Source: `~/system/specs/agentic-os-pillar4-skills-audit-2026-05-04.md` (§4)
MC: #99131 | #99176
Date: 2026-05-05

Methodology

Priority score formula:

```
priority_score = log10(skill_md_tokens_est) * (1 + invocations_30d)
```

Bonus weight ×1.5 if `frontmatter_description_bytes > 500`.

Tie-break rule: Higher `skill_md_tokens_est` wins.

Exclusion list:

- `owner=anthropic` vendor skills (docx, pdf, pptx, xlsx, figma-design) — VENDOR_REFACTOR_IMMUNE
- `_archived/` skills
- TOB skills with `skill_md_loc=0` (no SKILL.md at root, tokens_est=0, score undefined)
- Skills where `invocations_30d=NO_DATA` (none in this dataset — all zero values are grounded in grep)

Note on invocations_30d=0 skills: Ranked separately at bottom of table with priority_score computed as `log10(skill_md_tokens_est) * 1` (no invocation multiplier). This represents their per-session load cost without usage frequency.

Note on est_monthly_cost: The columns below show estimated cost per month. These projections assume sessions_per_month=600 and invocations_30d as a proxy for monthly rate. Per-turn savings are the honest metric; monthly projections are estimates only.

Top-20 Table (sorted descending by priority_score)

rank	skill_name	LOC	tokens	inv_30d	est_$/mo (current)	est_$/mo (post-L3)	savings_$/mo	priority_score	owner
1	task-postflight	541	5,367	21	$0.547	$0.078	$0.469	82.054	john
2	prompt-forge	224	2,372	20	$0.350	$0.070	$0.280	70.877	john
3	plan-with-team	140	1,177	13	$0.105	$0.042	$0.063	42.991	john
4	build-plan	90	923	7	$0.126	$0.063	$0.063	23.722	john
5	ask-board	307	2,623	3	$0.125	$0.038	$0.087	13.675	john
6	build	79	838	3	$0.113	$0.057	$0.056	11.693	john
7	sentinel	105	990	2	$0.116	$0.058	$0.058	8.987	john
8	sync	46	346	2	$0.087	$0.087	$0.000	7.617	john
9	learning-opportunity	165	1,433	1	$0.067	$0.034	$0.033	6.313	john
10	vault-unlock	117	1,312	1	$0.142	$0.071	$0.071	6.236	john
11	incident-response	122	1,051	1	$0.067	$0.034	$0.033	6.043	john
12	youtube-learning	93	877	1	$0.136	$0.068	$0.068	5.886	john
13	code-review	87	674	1	$0.002	$0.001	$0.001	5.657	john
14	lightrag-upload	87	659	1	$0.117	$0.059	$0.058	5.638	john
15	lightrag-status	101	625	1	$0.121	$0.061	$0.060	5.592	john
16	product-lifecycle	491	5,103	0	$0.081	$0.041	$0.040	3.708	john
17	skill-creator	362	4,911	0	$0.088	$0.044	$0.044	3.691	john

rank	skill_name	LOC	tokens	inv_30d	est_$/mo (current)	est_$/mo (post-L3)	savings_$/mo	priority_score	owner
18	doc-coauthoring	375	4,274	0	$0.208	$0.104	$0.104	3.631	john
19	mcp-builder	236	2,457	0	$0.135	$0.068	$0.067	3.390	john
20	plan-build-test	293	2,437	0	$0.099	$0.050	$0.049	3.387	john

est_$/mo (post-L3) = estimate assuming 50% body-token reduction via progressive disclosure

---

Per-Skill Triage (Top 10)

#1 task-postflight

- **Current footprint:** 541 LOC / 5,367 tokens
- **Why bloated:** BLOAT_LOC_GT_300 — Contains anomaly decision tree (Section 3), learning-opportunity dispatch template (Section 4), memory writer procedure (Section 5), and failure mode reference table (Section 8) all inline in one file. Most of this content is only needed after an anomaly is detected.
- **Recommended action:** Split — progressive-disclose. Trigger skeleton ≤200 LOC stays in SKILL.md; Sections 3-5+8 move to references/.
- **Predicted savings:** ~3,500 tokens/session on typical PASS flows (63% context reduction); full 5,367 tokens only loaded on ANOMALY path.

#2 prompt-forge

- **Current footprint:** 224 LOC / 2,372 tokens
- **Why bloated:** Single references/agent-briefs.md exists but body still contains full 5-panelist dispatch protocol, model tier assignments, and synthesis rules inline. Most body content is needed only during the forge step.
- **Recommended action:** Split — move per-panelist briefs and synthesis rules to references/; keep trigger condition and dispatch skeleton in core.
- **Predicted savings:** ~1,200 tokens/session when invoked without full panelist detail read (50% reduction).

#3 plan-with-team

- **Current footprint:** 140 LOC / 1,177 tokens
- **Why bloated:** No references/ dir. Builder/validator role descriptions, round-robin protocol, and output templates are all inline. Frequently invoked (13x in window) — every invocation carries full load.
- **Recommended action:** Progressive-disclose — move builder brief and validator brief to references/. Keep selection logic in SKILL.md.
- **Predicted savings:** ~700 tokens/session (59% reduction) across 13 monthly invocations.

#4 build-plan

- **Current footprint:** 90 LOC / 923 tokens
- **Why bloated:** No references/ dir. Moderate size but high invocation frequency (7x). Output templates and TaskList format examples inline.
- **Recommended action:** Progressive-disclose — move TaskList format examples and edge-case handling to references/quick-ref.md.
- **Predicted savings:** ~400 tokens/session (43% reduction).

#5 ask-board

- **Current footprint:** 307 LOC / 2,623 tokens
- **Why bloated:** BLOAT_LOC_GT_300 — 5-agent dispatch briefs are fully inline. Each panelist persona description (50-80 lines each) loads for every board invocation.
- **Recommended action:** Split — move per-panelist briefs to references/panelist-<name>.md. Keep dispatch skeleton (trigger, model tier, synthesis format) in SKILL.md.
- **Predicted savings:** ~1,800 tokens/session (69% reduction).

#6 build

- **Current footprint:** 79 LOC / 838 tokens
- **Why bloated:** No references/ dir. Build mode toggle and autocoder integration details inline. Reasonably compact but no progressive disclosure path for edge cases.
- **Recommended action:** Progressive-disclose — move edge-case handling (yolo mode, concurrency) to references/.
- **Predicted savings:** ~300 tokens/session (36% reduction). Low priority given small absolute size.

#7 sentinel

- **Current footprint:** 105 LOC / 990 tokens
- **Why bloated:** No references/ dir. 5-agent audit team definitions inline. Hardcoded audit procedure steps.
- **Recommended action:** Progressive-disclose — move per-agent audit checklists to references/; keep dispatch skeleton.

- **Predicted savings:** ~500 tokens/session (50% reduction).

#8 sync

- **Current footprint:** 46 LOC / 346 tokens
- **Why bloated:** Small and clean — no significant refactor needed. Score driven by 2 invocations.
- **Recommended action:** Keep as-is. Already close to L3 trigger skeleton.
- **Predicted savings:** Negligible. Lowest absolute token size in top-20.

#9 learning-opportunity

- **Current footprint:** 165 LOC / 1,433 tokens
- **Why bloated:** No references/ dir. Root-cause classification procedure, GOTCHA layer mapping, and fix-type catalog inline. Invoked only on anomaly path (1x).
- **Recommended action:** Progressive-disclose — move GOTCHA layer catalog and fix-type table to references/.
- **Predicted savings:** ~700 tokens/session (49% reduction).

#10 vault-unlock

- **Current footprint:** 117 LOC / 1,312 tokens
- **Why bloated:** HARDCODED_PATH (/Users/makinja) — breaks Pillar #9 VM portability. Caddy proxy restart sequence and bw CLI flags inline.
- **Recommended action:** Progressive-disclose + fix HARDCODED_PATH — move Caddy sequence to references/; replace /Users/makinja with $HOME.
- **Predicted savings:** ~600 tokens/session (46% reduction) + VM portability fix.

Aggregate Savings (per-turn, not monthly $)

skills loaded per turn	tokens saved vs. baseline	% context window recovered (128K window)
Only task-postflight (PASS path)	3,500 tokens	2.7%
task-postflight + prompt-forge	4,700 tokens	3.7%
Top-5 hot-path skills (ranks 1-5)	7,300 tokens	5.7%
All top-20 (max benefit, full session)	19,500 tokens	15.2%
All 79 skills at L3 (theoretical max)	~35,000 tokens	27.3%

Assumes 40-50% body-token reduction per skill post-refactor. Calculations: savings_tokens = current_tokens × 0.45. Context window basis: 128K tokens (Claude standard context). These are per-turn estimates derived from body-token reduction; monthly projections without measured session counts would be phantom claims.

← Back to Audit Summary | Design Pattern →

Design Pattern

Progressive Disclosure Design Pattern

Source: `~/system/specs/agentic-os-pillar4-skills-audit-2026-05-04.md` (§5)

MC: #99131 | #99176

Date: 2026-05-05

Derived from `~/.claude/skills/skill-creator/SKILL.md` ("context window is a public good"). This section codifies what is implicit in the canonical reference — it does not invent a new framework.

Definition

Progressive disclosure for skills means that skill content is loaded in tiers based on actual need:

- **Tier 1 (frontmatter — always-loaded):** Every time a Claude session starts, all SKILL.md frontmatter `description` fields are loaded to determine which skills to activate. Frontmatter is the highest-cost content per byte because it loads regardless of usage.
- **Tier 2 (SKILL.md body — loaded on trigger):** After a skill matches its trigger condition, the full SKILL.md body loads. This is the decision-making and branching layer.
- **Tier 3 (references/ — loaded on demand):** Content in `references/` is loaded explicitly via `Read <path>` only when the agent reaches a branch that needs it. Scripts in `scripts/` are invoked without being loaded into context.

The principle: never load content that is not needed for the current branch of execution.

L0–L3 Rubric

This rubric is used for the `progressive_disclosure_score` column in the inventory CSV.

Level	Definition	Body size	References	Frontmatter	Hardcoded paths

L0	Monolithic — entire skill in one file, no references/ dir	any (often > 200 LOC)	absent	any size	allowed
L1	SKILL.md exists + references/ dir may exist, but body > 200 lines OR references are read proactively (not conditionally)	> 200 LOC	optional	any size	allowed
L2	SKILL.md body ≤ 200 lines; references/ loaded conditionally on branch; no hardcoded paths	≤ 200 LOC	conditional	any size	not allowed
L3	SKILL.md ≤ 60-line trigger skeleton; references/ strictly on-demand per branch; frontmatter ≤ 500 bytes; no hardcoded /Users/makinja paths	≤ 60 LOC	on-demand only	≤ 500 bytes	not allowed

Distribution in current corpus:

- L0: 32 skills (40.5%) — monolithic, no references
- L1: 38 skills (48.1%) — body > 200 LOC or proactive refs
- L2: 2 skills (2.5%) — sentry-skill-scanner, task-splitter
- L3: 0 skills (0%) — no skill fully meets all L3 criteria

Note: skill-creator comes closest to L3 intent but is 362 LOC (exceeds the 60-line body target).

Reference Exemplar

The canonical reference for the L3 pattern is `~/.claude/skills/skill-creator/`.

This skill demonstrates:

- `references/output-patterns.md` — loaded only when generating skill output

- `references/workflows.md` — loaded only for the workflow design step
- Clear "when to Read" callouts in the body
- Frontmatter description that covers all trigger cases without bloat

The canonical pattern from skill-creator states:

> "Keep SKILL.md body to the essentials and under 500 lines to minimize context bloat. Split content into separate files when approaching this limit. When splitting out content into other files, it is very important to reference them from SKILL.md and describe clearly when to read them, to ensure the reader of the skill knows they exist and when to use them."

A true L3 implementation would reduce this further to ≤60-line skeleton with all procedural content in references/.

Anti-Pattern Catalog

All 9 anti-patterns documented (minimum 8 required per spec):

#	Pattern	Detector heuristic	Example skill	Fix
1	`BLOAT_LOC_GT_300`	`wc -l SKILL.md > 300`	task-postflight (541L), product-lifecycle (491L), doc-coauthoring (375L)	Move decision trees and reference tables to references/
2	`FRONTMATTER_GT_500B`	description field bytes > 500	docx (785B), xlsx (945B), pptx (690B), task-splitter (469B)	Condense to single-line trigger sentence; move examples to body
3	`INLINED_SCRIPT`	bash/python block embedded in markdown body	plan-build-test (Playwright CLI commands inline)	Move to scripts/run-tests.sh; invoke without loading into context
4	`DUPLICATE_PROCEDURE`	Same workflow steps appear in 2+ skills	product-lifecycle delegates to plan-with-team (6,266 tokens combined on product-lifecycle invocation)	Extract shared procedure to references/ in one skill; the other references it

#	Pattern	Detector heuristic	Example skill	Fix
5	`NO_TRIGGER`	No `description:` field in frontmatter, or field is empty	code-review (0B), qa-doc-review (0B), financial-overview (0B), invoice (0B), onboard-client (0B), onboard-partner (0B), send-for-signing (0B), form-filler (0B)	Add description: field with "Use when..." trigger condition
6	`NO_REFS_DIR`	No references/ subdirectory; entire skill in one file	70 of 79 skills	Create references/ dir; move branch-specific content
7	`DEAD_30D`	use_count=0 AND no log hits in 19-day measurement window	doc-coauthoring, product-lifecycle, design-system, debugging (all 0 invocations)	Audit whether skill is still needed; consider retire or merge
8	`HARDCODED_PATH`	`/Users/makinja` embedded in skill body	learning-opportunity, vault-unlock, form-filler, plan-build-test	Replace with `$HOME` or relative path; required for Pillar #9 VM portability
9	`UNREGISTERED`	Disk directory exists but missing from skill-registry.db	17 skills (ask-board, deploy-verify, fiken-agent, hop-build, incident-response, library, lightrag-*, prompt-forge, sync, task-postflight, task-splitter, template-meta-prompt, vault-unlock, web-search)	Run `INSERT INTO skills (name) VALUES ('<name>');` or skill-creator registration step

---

Three-Tier Load Model

The canonical Anthropic pattern (derived from skill-creator/SKILL.md):

Tier 1 — Always-Loaded (frontmatter only)

- **Content:** trigger condition + one-paragraph overview + when-to-use
- **Location:** YAML `description:` field
- **Target:** ≤ 60 lines total frontmatter / ≤ 1.5K tokens
- **Cost:** paid on every session, regardless of whether skill fires
- **Rule:** Never put procedural steps, code examples, or reference tables here

Tier 2 — Loaded on Trigger (SKILL.md body)

- **Content:** process steps, branching logic, tool whitelist, output contract
- **Location:** SKILL.md body (everything after frontmatter `---`)
- **Target:** 60-200 lines / token budget ≤ 5K
- **Cost:** paid when skill trigger matches
- **Rule:** Include branch decision table; link to Tier 3 files explicitly

Tier 3 — On-Demand (references/ and scripts/)

- **Content:** detailed procedures, examples, anti-pattern tables, worked code samples, branch-specific rules
- **Location:** `references/<branch>.md`, `scripts/<action>.sh`
- **Target:** unbounded; each file should be independently useful
- **Cost:** paid only when the agent reads the file on a specific branch
- **Rule:** Agent must see the file reference in Tier 2 SKILL.md body with explicit "when to read" instruction

---

Canonical Skill Skeleton Template

```
---
name: <kebab-case-name>
description: Use when <concrete trigger>. Does <one-line outcome>.
argument-hint: <stdin-arg>
---

# <name>

## 1. Preconditions (<= 30 lines)
- Hard checks. Abort fast. Cite the hook that enforces if any.

## 2. Branch decision (<= 30 lines)
Pick the procedure, then load it:

| Condition | Procedure |
|---|---|
| <condition-A> | Read `./references/<branch-a>.md` |
| <condition-B> | Read `./references/<branch-b>.md` |
```

```
## 3. Sub-agent dispatch contract (<= 40 lines)
- Model tier (Haiku/Sonnet/Opus + rationale)
- Tool whitelist
- Brief path: `./references/<role>-brief.md`
- Output contract (path + format)

## 4. Closure (<= 30 lines)
- mc.js submission shape
- Memory write rule (cite owning skill; do NOT reimplement)

# Body MUST stay under 200 lines.
# Anything longer goes into references/<branch>.md.
```

This template will be promoted to `~/system/specs/skill-skeleton-canonical.md` as a separate Skillforge step.

← Top-20 Priority | PoC Analysis →

PoC: task-postflight

PoC: task-postflight Refactor

Source: `/tmp/pillar4-99131-out/poc-task-postflight-tier1.md` + `~/system/specs/agentic-os-pillar4-skills-audit-2026-05-04.md` (§6)

MC: #99131 | #99176
Branch: `feat/pillar4-skills-poc` (merged to master 2026-05-05 ef8536ad)
Date: 2026-05-05

Overview

The PoC refactor of `task-postflight/SKILL.md` validates the three-tier progressive disclosure pattern on the highest-priority target.

- **Target:** 541 LOC → ≤ 200 LOC core trigger skeleton
- **New references/ files:** `anomaly-decision-tree.md`, `proveo-rubric.md`, `memory-writer.md`
- **Existing references/ preserved:** `proveo-brief.md`, `learning-loop.md` (unchanged)

Token Reduction Analysis

Metric	Before	After (PASS path)	After (ANOMALY path)
SKILL.md LOC	541	≤190	≤190
SKILL.md bytes	19,859	~8,200	~8,200
Tokens loaded	5,367	~2,216	~2,216
Additional refs loaded	0	0	~3,000 (anomaly-decision-tree)
Total tokens (PASS path)	5,367	2,216	N/A
Reduction on PASS path	—	**59%**	—
Bytes reduction check	—	11,659 bytes saved / 19,859 = 58.7% ≥ 40%	**PASS**

The ≥40% byte reduction target is met on the typical PASS path. On anomaly paths, the `anomaly-decision-tree.md` is loaded (~3,000 additional tokens) but this is appropriate because the anomaly path requires that content.

PoC Target Rationale

Selection: task-postflight

Priority score comparison:

Skill	tokens_est	inv_30d	priority_score
task-postflight	5,367	21	**82.054**
prompt-forge	2,372	20	70.877
doc-coauthoring	4,274	0	3.631
product-lifecycle	5,103	0	3.708

Petter's preference for `doc-coauthoring` (376L, clean three-stage structure) was overridden by frequency data. `task-postflight` fires on every H/BLOCKER closure — 21 times in 19 days. `doc-coauthoring` has 0 measured invocations. Frequency × size dominates structural elegance.

Trigger-Map Table

Content migration plan (required before coding per D3 pre-refactor requirement):

Current SKILL.md section	Lines	Always needed?	Move to
Frontmatter (description)	8	YES — trigger	Keep in SKILL.md
§1 Preconditions	32	YES — fail fast	Keep in SKILL.md
§2 Proveo dispatch	40	YES — every invocation	Keep in SKILL.md
§3 Anomaly decision tree	38	NO — only after Proveo returns	→ references/anomaly-decision-tree.md
§4 Learning-opportunity dispatch template	62	NO — only on ANOMALY path	→ references/anomaly-decision-tree.md
§5 Memory writer procedure	38	NO — only if learning returns memory	→ references/memory-writer.md

Current SKILL.md section	Lines	Always needed?	Move to
§6 Postflight marker writer	72	PARTIAL — Section 6a-6b always needed, 6c-6f only on success path	6a-6b keep; 6c-6f → references/marker-writer.md
§7 mc.js ready writer	52	PARTIAL — format in SKILL.md; details → references/	Keep dispatch shape; move table → references/
§8 Failure modes table	30	NO — reference only	→ references/anomaly-decision-tree.md
§9 Audit trail	20	YES — always runs	Keep in SKILL.md
v0.1 TODO + References footer	30	NO	Drop from PoC (TODO deferred)

Content Split

Stage 1 content (stays in SKILL.md core)

- Preconditions (1a-1c)
- Proveo dispatch inputs and expected output format
- Anomaly routing decision (IF/THEN — 4 cases, each with "Read ./references/anomaly-decision-tree.md")
- Postflight marker check (6a-6b only)
- mc.js ready submission shape
- Audit trail append

Stages 2-N content (moved to references/)

- Full anomaly class decision tree → `references/anomaly-decision-tree.md`
- Learning-opportunity invocation template → `references/anomaly-decision-tree.md` (same file, appended)
- Memory writer procedure → `references/memory-writer.md`
- Marker writer full procedure (6c-6f) → `references/marker-writer.md`
- Failure modes table → appended to `references/anomaly-decision-tree.md`

Existing references/ files

- `references/proveo-brief.md` — keep as-is (already correct progressive disclosure)
- `references/learning-loop.md` — keep as-is

Verification Commands

```
# 1. Branch exists
git -C /Users/makinja/.claude branch --list "feat/pillar4-skills-poc"

# 2. LOC check
wc -l /Users/makinja/.claude/skills/task-postflight/SKILL.md  # must be <= 200

# 3. Reference files exist
ls /Users/makinja/.claude/skills/task-postflight/references/anomaly-decision-tree.md
ls /Users/makinja/.claude/skills/task-postflight/references/proveo-rubric.md
ls /Users/makinja/.claude/skills/task-postflight/references/memory-writer.md

# 4. Before/after snapshots exist
ls /tmp/pillar4-99131-out/poc-task-postflight-before.md
ls /tmp/pillar4-99131-out/poc-task-postflight-after.md

# 5. Byte reduction >= 40%
python3 -c "
before=$(wc -c < /tmp/pillar4-99131-out/poc-task-postflight-before.md)
after=$(wc -c < /tmp/pillar4-99131-out/poc-task-postflight-after.md)
print(f'Reduction: {(before-after)/before*100:.1f}%')
print('PASS' if (before-after) >= 0.4*before else 'FAIL')
"

# 6. No section header permanently lost
# Proveo verifies headers in before.md appear in {after.md + 3 ref files}
```

PR Merged

Branch: `feat/pillar4-skills-poc`
Merged: 2026-05-05T09:17:35Z
Commit: ef8536adba17
Status: Merged to master
Result: 541 LOC → 194 LOC (64.7% reduction)

← Design Pattern | Audit Summary