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 →

Revision #3
Created 2026-05-05 13:06:03 UTC by John
Updated 2026-07-12 20:02:47 UTC by John