Preface — Why a Playbook (Edition 2)

Between March and April 2026, Claude Code subscribers lived through six weeks of regressions stacked one on top of another. Cache TTL shortened from sixty minutes to five. Opus 4.7 shipped a tokenizer that charged 1.35 to 1.46 times more for the same prompts. Pro plan access briefly disappeared and returned forty-eight hours later with no public explanation. Third-party tool policy tightened in the same week an independent engineer published a well-received $45/month DIY alternative to Max. Weekly quota meters drifted. Users hit caps in an afternoon on accounts that had run clean for six months.

Edition 2 adds four new triggers, three new paths, and an expanded decision tree to the original framework. Trigger 11 covers the silent regression and quota burnout cluster that surfaced across the v2.1.13x release window. Trigger 12 covers billing system failures as a migration signal. Trigger 13 covers the claim-reality divergence cluster, a 49-case body of evidence that the platform's success assertions and its runtime behavior have structural gaps. Trigger 14 covers the irreversible operation cluster anchored to four Tier-1 media incidents.

The path framework also expands. Path A' (harness reduction) is a second Stay-and-Fortify posture that reduces always-on prompt volume by 67%. Path B' (model replacement) keeps the harness and swaps the model via GLM Coding Plan or DeepSeek V4 Pro. Path D (hybrid delegation) is a third Path C posture that splits work between Claude Code as orchestrator and an alternative model as worker.

The decision tree expands from six gates and eight leaves to eight gates and thirteen leaves. Edition 1 buyers who re-run the tree under Edition 2 find their recommendation more precise; the original Edition 1 leaf is not invalidated, just narrower-targeted to workflow attributes Edition 1 did not measure.

Edition 2 ships as a free PDF update for every Edition 1 buyer. Gumroad delivers the updated file automatically — no re-purchase, no email request, no notification required from the buyer side.

The sister volume, Claude Code Claim-Verify Handbook ($19, ships 2026-05-22 same day), is Edition 2's evidence companion for the Trigger 13 cluster. Forty-nine cases (fifteen in body, thirty-four in Appendix D's publication-window continuation evidence) with full quotation, a three-stage framework, 14 operator-side defenses, and 5 detection tools (all five implemented and tested, 165+ test cases passing). Buy both if your workflow has irreversible operations downstream of agent assertions.

What this book is not (unchanged from Edition 1):

- Not a Cursor advertisement. Every alternative platform is evaluated on the same six axes with the same skepticism.
- Not a DIY evangelism. Path A — stay on Claude Code and fortify — is a real and common answer for readers below the break-even burn rate.
- Not a speculation about Anthropic's internal decisions. Only the publicly dated record informs the recommendations.
- Not a replacement for Token Book EN. Token Book covers everyday token economics; this covers platform economics.

What this book is:

An aggregator. It takes your daily burn rate, your triggered event count, your subscriber tier, and your dependency posture, and returns one of thirteen recommendations across the expanded decision tree. The decision is measurable and reversible. Chapter 9 ensures it is reversible within forty-eight hours if the tree's output turns out, in retrospect, to have been wrong.

How to read Edition 2

Each chapter stands alone. Edition 2 supplements are clearly labeled and follow the canonical chapter they supplement. Skip to the one that matches the question you are asking right now.

Question	Chapter
"What actually happened in March–April 2026?"	1
"Which events should make me consider migrating at all?"	2 + Edition 2 supplements (triggers 11-14)
"Which signals look alarming but should not trigger a migration?"	3
"I want to stay on Claude Code. How do I fortify?"	4 + Edition 2 supplements (Path A', hook batch, enterprise)
"I am thinking about Cursor, Cline, Codex, or Aider. How do they compare?"	5 + Edition 2 supplement (Path B')
"Someone told me about a $45/month hybrid DIY stack. Is that real?"	6 + Edition 2 supplements (proxy, Path D)
"I want to measure what migration actually costs me."	7
"Give me a decision. One output, not a menu."	8 + Edition 2 supplement (8 gates / 13 leaves)
"I migrated and regret it. How do I roll back?"	9

The book commits explicitly to a revised edition (free update for existing buyers) if any of the following happens: Anthropic ships a documented tokenizer cache fix, Pro plan policy stabilizes for sixty consecutive days, or the third-party tool policy reverses. The cover carries "Edition 2, May 2026" so the evidence horizon is never in doubt.

Reader disclaimer

This book is reconstructed from public information: GitHub Issues on `anthropics/claude-code`, Hacker News threads, Reddit r/ClaudeAI and r/ClaudeCode top threads, Simon Willison's measurement posts, The Register's coverage, and independent reverse-engineering. It is not affiliated with or endorsed by Anthropic.

When in doubt, measure your own `/usage --json` output. Trust that over anything in this book.

Part 1 — What Actually Broke

Chapter 1 — The Six-Week Timeline

Between early March and late April 2026, Claude Code subscribers lived through six weeks of compounding regressions. Each incident — taken in isolation — could be read as a routine platform hiccup. Taken together, they form the evidence base every migration decision in this book is built on. This chapter is deliberately factual. No recommendations yet; no framing beyond the chronology. The purpose is to put every reader in front of the same dated record before the decision chapters that follow.

If you already lived through this stretch as a daily Claude Code user, parts of this timeline will be familiar — and at least two or three entries will not. Read the ones that are new to you carefully. Every claim is sourced.

Early March 2026 — The cache TTL silently shortens

Sometime in early March, Anthropic shortened Claude Code's prompt-cache time-to-live from sixty minutes (`ephemeral_1h`) to five minutes (`ephemeral_5m`). No changelog entry. No user-visible announcement. The regression was first documented publicly in GitHub Issue #46829, filed 2026-04-12 by @seanGSISG, which correlated the change against 119,866 API calls collected over 92 days on two independent machines.

The issue title — "Cache TTL silently regressed from 1h to 5m around early March 2026, causing quota and cost inflation" — named both the mechanism and the downstream harm. The issue documents zero `ephemeral_5m` tokens across thirty-three consecutive active days (February 1 through March 5), then a gradual reappearance starting March 6, with `ephemeral_5m` dominating the mix by March 8. Applied to Sonnet and Opus 4.6 pricing, the combined dataset showed approximately 17.1% cost waste (roughly $1,582 overpaid over the 92-day window for an Opus-priced workload). A separate community analysis by Jacek Mariański ("Claude Code Cache Crisis: A Complete Reverse-Engineering Analysis", Medium, April 2026) reached the same temporal conclusion and extended the work at the binary layer. Mariański's series later identified four additional cache regressions at the binary layer — the Resume Attachment Relocation Bug (v2.1.69+), the Dynamic Tool Descriptions invalidation (v2.1.36–v2.1.87), the `cch=00000` Native Sentinel Replacement (v2.1.36+, unfixed), and the extractMemories Double Cache Chain (default-on background call).

March 26 — The reasoning history mid-session discard bug

Anthropic later acknowledged this entry in a Fortune interview on 2026-04-24 ("Anthropic says engineering missteps were behind Claude Code's monthlong decline after weeks of user backlash"). On March 26, a bug shipped that caused the model to discard its reasoning history mid-session — silently dropping context the model itself

had just built up. Subscribers experienced this as Claude appearing to "forget" what it had just been working on inside a single session, requiring repeated re-explanation that compounded the cache TTL regression already in flight. Anthropic's public statement classified this as the first of three engineering missteps acknowledged retroactively.

March 31 — The first mainstream press coverage

The Register published "Anthropic admits Claude Code quotas running out too fast" (The Register, 2026-03-31). Anthropic's official statement acknowledged the quota-drain reports without attributing them to the cache change. For the first time, a user's inability to finish a day's work on a $200/month subscription was discussed as a systemic issue rather than individual misuse.

Early April — The Tokenocalypse cluster

The first week of April produced four independent events that compounded the existing cache regression into what the community subsequently called the Tokenocalypse. In chronological order:

- **April 4** — Anthropic announced that third-party harnesses (OpenClaw, Cline, Aider running against a Claude Code Pro/Max subscription) would no longer consume subscription-tier tokens (techcrunch.com, 2026-04-04). Users relying on these tools faced immediate API-credit burn for workflows they had believed were covered.
- **Sonnet backend outage** — A roughly three-hour Sonnet outage on the same day produced an automated retry spike across many users' stacks, multiplying the cache-miss cost described in the March 6 regression.
- **Mythos Preview ship** — Anthropic's Mythos preview entered general availability, landing new tool schemas and cache invalidation points.
- **171 internal emotion vectors paper drop** — An Anthropic-published research paper was absorbed into extended context for many community tools, silently enlarging conversation overhead.

None of these four events were regressions in themselves. Their simultaneity, overlaid on the cache TTL change, produced the burst of quota-cap reports that dominated r/ClaudeAI and r/ClaudeCode through the second week of April.

April 12 — Tyler Folkman publishes the first DIY alternative

Independent engineer Tyler Folkman published "I Replaced Claude Code with a $45/month DIY Stack" (tylerfolkman.substack.com, 2026-04-12). The post documented a working substitution: an Anthropic API key with a self-imposed budget cap, Claude Desktop plus MCP servers, Continue.dev as the IDE surface, and a cloud dev-

environment orchestrator. Folkman claimed equivalent capability at roughly one-quarter of the $200 Max subscription cost. The post was heavily circulated on HN and set a baseline the later "stay-versus-switch" debate would anchor against.

April 13 — Two Register articles on the same day

The Register ran two connected pieces in twenty-four hours. "Claude is getting worse, according to Claude" (The Register, 2026-04-13) documented a 99% uptime SLA met while subjective output quality fell. "Anthropic: Claude quota drain not caused by cache tweaks" (theregister.com, 2026-04-13) carried Anthropic's direct denial that the TTL change had any meaningful quota impact. The same week, an r/ClaudeAI 1,140-session analysis published on XDA Developers measured average cache busts per user rising from 39/day to 199/day and average spend rising from $6.28/day to $15.54/day month-over-month (xda-developers.com).

April 16 — Opus 4.7 ships

Opus 4.7 reached general availability. Within 48 hours, two community-scale reactions formed. A Reddit post titled "Opus 4.7 is not an upgrade but a serious regression" collected 2,300 upvotes. An X post making the same argument collected 14,000 likes. Anthropic's own release notes acknowledged that 4.7 would consume "roughly 1x to 1.35x as many tokens" than 4.6 for comparable work. Simon Willison's independent measurement (simonwillison.net, 2026-04-20) put the system-prompt ratio at 1.46x and total cost inflation at approximately 40%.

Nine breaking changes shipped with 4.7 — extended thinking budgets removed, sampling parameters removed, thinking content omitted by default, adaptive thinking off by default, literal instruction following, calibrated response length, fewer tool calls, fewer subagents, and a more direct opinionated tone. Five of those nine were silent changes, producing errors with no deprecation signal.

Objective long-context retrieval regressed as well. Multi-Round Coreference Resolution (MRCR) scores moved from 78.3% on 4.6 to 32.2% on 4.7 — a 46-point drop. An 8-needle retrieval test at a 256k window moved from 91.9% to 59.2% (blog.wentuo.ai). Anthropic later indicated it would phase out MRCR in favor of GraphWalks, a benchmark on which 4.7 performs better.

April 16–20 — The 25-word response cap

Anthropic's 2026-04-24 Fortune acknowledgment named a second engineering misstep: a change shipped on April 16 that capped the model's response length to roughly 25 words between tool calls. The change was framed internally as a latency optimization; in practice it broke coding quality on multi-step refactors where the model needs more than 25 words to explain why a particular tool call is happening or to coordinate two

related edits. The cap was reverted on April 20, four days after shipping. Subscribers who used Claude Code intensively during that four-day window saw measurably lower coding-task success rates that resolved when the cap was rolled back.

April 17 — Claude Design launch

Anthropic shipped Claude Design (Anthropic docs; coverage: venturebeat.com, 2026-04-17), a Figma-style prototyping surface. Unrelated to the quota and regression debates in the same week, but relevant to migration decisions because it signaled where Anthropic's near-term product investment was aimed — away from the power-user CLI path.

April 21–22 — The Pro plan scare

On April 21, Anthropic removed Claude Code access from the $20/month Pro plan without prior notice. HN thread #47855832 collected 400+ comments in twelve hours. Search volume for mass-cancellation queries rose roughly 10x over 48 hours. On April 22, Anthropic reversed the change for approximately 98% of affected users (with 2% retained as an ongoing test) and Head of Growth Amol Avasare published a correction. Simon Willison's summary (simonwillison.net, 2026-04-22) captured the pricing uncertainty that persisted even after the reversal.

April 23 — The all-subscriber usage limits reset and public acknowledgment

On April 23, Anthropic reset the usage limits for all subscribers — a one-time across-the-board credit applied without prior announcement. The next day, in the Fortune interview cited above, the company acknowledged a monthlong Claude Code decline driven by three engineering missteps (the March 26 reasoning-discard bug, the April 16 25-word response cap, and a third issue not named in the public coverage). Anthropic's stated position: "This isn't the experience users should expect from Claude Code," with a promise of greater transparency around future changes. The reset is the first time Anthropic has issued a blanket credit specifically attributable to acknowledged engineering errors rather than to platform-wide outage compensation.

This entry matters for the migration decision in two directions. First, it argues that Anthropic can acknowledge and compensate when the engineering chain breaks down — a meaningful counter-argument against the "Anthropic ignores power users" narrative that drove the early-April community frustration peak. Second, the Fortune statement explicitly excludes the cache TTL regression: the three acknowledged missteps are now resolved; the cache TTL change is not on the acknowledged list and remains closed `not planned` as of the same week. The two postures coexist deliberately.

April 23–24 — The third-party tool ban

Anthropic clarified and tightened the third-party-tool policy first announced April 4. Workflows using Cline, Cursor-via-Claude-API-key, or Aider against a subscription account were explicitly out of policy. Tyler Folkman's April 12 $45/month stack — built partly on tooling now outside policy — became a contested reference point rather than a working recipe.

April 24–25 — The trailing incidents

Three final entries close the timeline covered by this book's first edition:

- **Issue #52890** (opened 2026-04-24) — The `claude update` command re-downloads approximately 226 MB each invocation, consuming roughly 7% of a Max 5-hour quota window per update.
- **Issue #52893** — Opus 4.7 reinvention behavior: the model rewrites existing project code under apparent good-faith interpretation of its own instructions.
- **Issue #52929** — Permission-visibility bug: settings required to view the effective permission set no longer match what Claude Code is enforcing in practice.
- **2026-04-25 — Liz Fong-Jones publishes the per-version diagnostic.** @lizthegrey's reply on Issue #46829 shipped a single `grep + jq` one-liner over `~/.claude/` that returns per-day per-version counts of sessions where `cache_creation.ephemeral_5m_input_tokens > 0`. The diagnostic moves Trigger 1 from "trust the issue thread" to "verify against your own logs," and is the first user-runnable measurement of the cache TTL regression that does not require re-running the binary disassembly Mariański's series performs. Appendix C reproduces the diagnostic with a 5m/total ratio variant for direct comparison against the 0.40 threshold in Chapter 2.

Reading the timeline

This is the evidence base. The rest of the book converts this record into decisions. Chapter 2 identifies which of these events constitute a migration trigger and how to measure each one against your own usage. Chapter 3 identifies which ones look like triggers but are not — a counter-anchor against over-reaction in an emotional week. Everything that follows rests on the dated, sourced entries above.

Chapter 2 — The Five Migration Triggers

The April 2026 record in Chapter 1 is noisy. Some of it is real cause-and-effect; some of it is subjective commentary; some of it is press. This chapter isolates the five events that each, on their own, constitute a measurable migration trigger. A trigger is defined narrowly: a regression or policy change that (a) is sourceable to a dated event, (b) can be measured against your own usage data, and (c) is the kind of thing reasonable Claude Code users will make different decisions about depending on their burn rate and posture.

Five triggers. Each with a threshold you can evaluate against your own `/usage --json` output today. If three or more of your thresholds are tripped, the book's central recommendation is that you at minimum begin the cost-forecast exercise in Chapter 7 — not that you migrate, but that you gather the data. If one or two trip, the Stay-and-Fortify path in Chapter 4 likely suffices. If none trip, you are probably not the buyer of this book.

Trigger 1 — Cache TTL Regression (1h → 5m)

Source event. In early March 2026, Claude Code's prompt-cache time-to-live shortened from `ephemeral_1h` to `ephemeral_5m`. No changelog entry; no user announcement. Surfaced publicly in Issue #46829 with 119,866 calls over 92 days as evidence. A separate r/ClaudeAI analysis of 1,140 sessions measured the before/after shift as 39 cache-busts-per-day rising to 199 per-day, with average daily spend moving from $6.28 to $15.54.

Why it is a trigger, not just noise. The 5-minute TTL is short enough that any conversational pause — reading a file, waiting for a build, replying to a Slack message, refilling coffee — invalidates the cache. Cache-miss cost is real dollars on the API side and real minutes against the 5-hour Max quota window. Users who never bumped the quota cap before began hitting it in April for the first time.

Measure it against yourself. From a recent Claude Code session on v2.1.118 or later, run `/usage --json` and examine the `cache_creation_tokens` versus `cache_read_tokens` fields over your last full working day.

- **Trigger threshold:** `cache_creation / (cache_creation + cache_read) > 0.40` across a full day of work.
- **Interpretation:** if more than 40% of your cache-relevant traffic is creation rather than read, the 5-minute TTL is actively costing you. The healthy baseline under 1h TTL for a typical heavy user sat in the 0.10–0.20 range.

- **Anthropic response record.** Through an April 13 Register piece, Anthropic stated the TTL change did not cause quota drain at the scale users reported. The Issue was subsequently closed "not planned" on the API-user side, with Max-quota implications handed off to Issue #45756. The configurational closure shipped two days after that close: v2.1.108 (2026-04-14) added `ENABLE_PROMPT_CACHING_1H` to opt into 1-hour TTL on API key, Bedrock, Vertex, and Foundry, plus `FORCE_PROMPT_CACHING_5M` for cost-sensitive subagent workflows (changelog). Users running v2.1.121 should hold the env var until Issue #54485 (cache_control payload rejection bug) is resolved; users on prior versions can export `ENABLE_PROMPT_CACHING_1H=1` immediately and pair it with the operational levers in Chapter 4.

Trigger 2 — Tokenizer Inflation on Opus 4.7

Source event. Opus 4.7 shipped 2026-04-16 with a new tokenizer. Anthropic's release notes acknowledged "roughly 1x to 1.35x as many tokens" relative to 4.6. Independent measurement by Simon Willison (simonwillison.net, 2026-04-20) recorded a system-prompt ratio of 1.46x and total cost inflation around 40%. The long-context retrieval benchmark MRCR moved from 78.3% (4.6) to 32.2% (4.7), a 46-point regression; 256k 8-needle retrieval moved from 91.9% to 59.2%.

Why it is a trigger, not just noise. The change is fully silent from the user's perspective: same prompts, same codebase, same IDE, but the session cap arrives 30–40% earlier. Users running 4.7 on repetitive work measured the effect within days; users on 4.6 via `ANTHROPIC_MODEL=claude-opus-4-6` experienced no inflation — the divergence is clean and attributable.

Measure it against yourself. Compare the two most recent equivalent working days — same project, similar task type, both on the same Claude Code version. Export `/usage --json` for each. Compute `total_input_tokens` on the 4.7 day divided by the 4.6 day.

- **Trigger threshold:** measured ratio > 1.30× on equivalent work.
- **Interpretation:** you are paying 30%+ more for the same output. On the $200/month Max 20× plan, that is effectively the difference between meeting and missing your daily quota before end-of-business.
- **Workaround caveat.** Pinning to 4.6 via `ANTHROPIC_MODEL=claude-opus-4-6` works today. Its availability is tied to Anthropic's deprecation schedule and is not guaranteed indefinitely.

Trigger 3 — Pro Plan Policy Instability

Source event. On 2026-04-21, Anthropic silently removed Claude Code from the $20/month Pro plan. HN thread #47855832 collected 400+ comments within twelve hours and a measurable 10x spike in mass-cancellation search queries over 48 hours. On 2026-04-22, Anthropic reversed the change for an announced 98% of users, with 2%

retained as a continuing test. Head of Growth Amol Avasare published a public correction. Simon Willison's subsequent coverage (simonwillison.net, 2026-04-22) captured the residual uncertainty.

Why it is a trigger, not just noise. The event is a trigger only for Pro subscribers, but for that cohort it is uniquely serious: the contractual stability of a $20/month subscription is now a question rather than a given. The 2% continuing test means some readers of this book are still in the removed state as of 2026-04-25. Even for the 98% restored, the precedent is established.

Measure it against yourself. This trigger is not measured from `/usage`; it is measured from your subscription status.

- **Trigger threshold (Pro users):** any of the following is true: (a) you were in the affected 2% cohort, (b) your workflow has a critical daily dependency on Claude Code that you cannot tolerate losing on 24 hours' notice, or (c) your total monthly AI-tooling budget is elastic but your Pro subscription is the only Claude-capable entry.
- **Interpretation:** for Max subscribers, this trigger does not fire. For Pro subscribers meeting any of the three conditions above, Chapter 5 (Path B — Switch Platforms) and Chapter 6 (Path C — Hybrid DIY Stack) both deserve evaluation.

Trigger 4 — Weekly Quota Reset Bugs

Source event. A cluster of five Issues landed between 2026-04-23 and 2026-04-24 documenting Anthropic-side quota-meter bugs: #52472 (reset occurring before scheduled time), #52484 (reset schedule misalignment), #52497 (counter dropping mid-cycle while `resets_at` remained in the future), #52498 (reset occurring ~2 days early with the displayed time shifting backward), and #52921 (Max 20× weekly counter resetting on a ~24-hour cycle instead of the documented 7-day cycle; the reporter confirmed via Anthropic's Fin support channel that the observed behavior contradicts documented specifications but Fin could not escalate, making GitHub Issues the only remaining engineering-reach surface). The cluster represents a quota-metering bug distinct from consumption-side issues like the cache TTL regression — here the meter itself is unreliable.

Why it is a trigger, not just noise. Quota that cannot be predicted cannot be budgeted. Teams that schedule heavy-tokens work around known reset windows lose the ability to do so. Individual users lose the "Friday afternoon I have quota left to try the big refactor" plannability.

Measure it against yourself. Over any recent 7-day cycle, compare the quota reset time displayed in Claude Code at the start of the cycle against the time the counter actually resets.

- **Trigger threshold:** displayed reset time differs from observed reset time by more than 6 hours in either direction, or the counter changes by more than 20% in a single observed hour outside a known bulk job.
- **Interpretation:** if your cycle shows this pattern, your quota budgeting is fictional, and the cost-forecast worksheet in Chapter 7 cannot produce reliable projections on this plan until the Anthropic-side fix lands.

Trigger 5 — Third-Party Tool Ecosystem Ban

Source event. 2026-04-04 announcement that third-party harnesses (OpenClaw, Cline, Aider running against a Claude Code subscription) will require separate API billing rather than consuming subscription tokens (techcrunch.com, 2026-04-04). Tightening and clarification through 2026-04-23–24 extended the constraint to Cline-based workflows and to subscription-account use of Claude API keys from non-Anthropic IDEs. Tyler Folkman's widely circulated "$45/month DIY stack" (tylerfolkman.substack.com, 2026-04-12) — published between the two policy moves — became partially out of compliance within eleven days.

Why it is a trigger, not just noise. The third-party CLI/IDE ecosystem was the primary cost-optimization path for high-burn users. Policy moves that close that path change the migration calculus materially for exactly the subscribers most likely to consider migration.

Measure it against yourself.

- **Trigger threshold:** you were using, planning to use, or have budgeted for any of: Cline / Cursor-via-Claude-API / Aider / OpenClaw against a Claude Code subscription in the next 60 days.
- **Interpretation:** if so, either Path B (Switch Platforms entirely, with a non-Anthropic API key underneath) or Path C (Hybrid DIY Stack on a personal API tier) replaces the previous expectation. Chapter 6 walks through the compliance-safe version of Folkman's recipe.

Stacking the triggers

The book does not treat one tripped trigger as a migration recommendation. One trigger is a signal to watch; two to evaluate; three or more to begin the forecast exercise in Chapter 7 and to read Chapters 4–6 with intent. Zero tripped triggers means you are likely in the Path A cohort whether you realize it or not — which is fine, and Chapter 4 is written for you specifically.

Chapter 3 is the counter-anchor: signals that look like triggers but are not. Read it before you let a noisy week produce a decision you will regret at 60-day switching cost.

This is the free sample

You have reached the end of the free sample of Claude Code Migration Playbook (Edition 2). The full edition is 251 pages.

The remaining pages cover:

- The later trigger clusters, Trigger 11 through Trigger 16 — silent regression and quota burnout, claim-reality divergence, irreversible operations, and project-scoped settings as an MCP injection vector.
- Chapter 3, the counter-anchor: the signals that look like migration triggers but are not.
- The three paths in full — stay and fortify, switch platforms, hybrid DIY stack — with six supplements, including harness reduction, model substitution without a platform switch, the backend-swap proxy, and hybrid delegation.
- Chapter 7's cost forecasting worksheet, and Chapter 8's decision tree, expanded from three paths to six.
- Chapter 9's 48-hour rollback checklist, and Chapter 10 on team performance metrics after the June 15 billing split.
- Four appendices: source citations, a competitor evaluation grid, further reading, and the updates log.

The hooks referenced throughout are MIT-licensed and free at github.com/yurukusa/cc-safe-setup.