SONNET 5 · OPUS 4.8 · FABLE 5

The Trio.

One codebase. Three models. Pick right.

Not a spec sheet. A decision guide for engineers who ship every day with Claude Code.

THE BRIEF, IN THREE BEATS

01 Three tiers, three prices — **Sonnet 5 $2/$10 · Opus 4.8 $5/$25 · Fable 5 $10/$50** per 1M tokens.

02 All three ship inside Claude Code. The default is Sonnet 5 — *and that's usually the right answer.*

03 Routing beats picking. Move up the ladder only when the task's shape demands it.

SLIDE 02 · HOOK

"Just run it on Fable 5"

— the sentence that burned 75% of a monthly quota in minutes.

Fable 5 is 5× more expensive than Sonnet 5 and burns ~1.3× more tokens per task. Reaching for the top of the ladder by default is the fastest way to drain the budget without moving the ship.

Sources: rentierdigital "Fable 5 Ultracode burned 75% of quota" (Jun 2026) · productcompass.pm Fable 5 guide v3 · credyt.ai Fable 5 pricing (Jun 9, 2026).

THE CLAIM

The right model isn't the smartest — it's the one that matches the task's shape. And the shape is decided by how many steps, how much context, and how much money.

THREE COROLLARIES

#1 **Start at Sonnet 5.** It's the Claude Code default for a reason — 82.1% SWE-Bench for 40% of Opus's price.

#2 **Escalate to Opus 4.8** when the task crosses files, systems, or hours. It's the *agentic* pick.

#3 **Reach for Fable 5** only when the ceiling of judgment matters more than the bill. Rare, precious, budgeted.

THE HONEST TRADE

The trap. "Frontier by default" — every task on Fable 5. Feels safe. Burns quota. No lift on 80% of the work.

The move. Match model to shape: quick tweak → Haiku; feature work → Sonnet 5; long-horizon agentic → Opus 4.8; frontier judgment → Fable 5.

The result. ~70% of runs on cheap tiers, ~2% on Fable 5 — same output quality, ~4× lower bill.

Sources: anthropic.com/news/claude-sonnet-5 · mindstudio.ai AI model routing Fable/Opus/Sonnet/Haiku · medium.com naresh 70/20/8/2 routing split.

Three tiers. One workhorse. One agent. One frontier. Same API, different jobs.

THE WORKHORSE

Sonnet 5

codename: Fennec · Jun 30, 2026

What: Claude Code's default. Fast, cheap, near-Opus quality.

SWE-Bench: 82.1% Verified · 72.7% (vendor).

Context: 1M tokens.

Price: $2 / $10 per 1M in/out.

Effort default: `high` on Claude Code.

> The one you should be running by default — matches Opus on most day-to-day tasks at ~40% of the cost.

THE AGENT

Opus 4.8

May 28, 2026 · retires Jun 15, 2026 (Opus 4)

What: Long-horizon agentic coder. Multi-file, multi-hour, high-judgment.

SWE-Bench: 88.6% Verified · 69.2% Pro.

Terminal-Bench 2.1: 74.6%.

Price: $5 / $25 per 1M in/out.

Signature: "Super-Agent" — the only model that completed every case end-to-end.

> The one you switch to when Sonnet 5 stalls or the task has real judgment.

THE FRONTIER

Fable 5

Mythos-class · Jun 9, 2026

What: First broadly available Mythos-class model. Dynamic Workflows + parallel subagents.

SWE-Bench: 95% Verified · 80.3% Pro.

Price: $10 / $50 per 1M in/out.

Unique: Ultracode mode, nested subagents (depth ≤ 5), Opus 4.8 auto-fallback for safety.

> The one you use when a 2% of your workload deserves 20× the token budget.

Sources: anthropic.com/news/claude-sonnet-5 · anthropic.com/news/claude-opus-4-8 · llm-stats.com Fable 5 review · morphllm.com Claude Benchmarks 2026 · platform.claude.com/docs models overview.

MAY-JUN 2026

Three releases in five weeks — the shape of the Anthropic lineup reset.

DATE	RELEASE	WHAT LANDED	WHY IT MATTERS
May 28, 2026 day 0	Claude Opus 4.8 Anthropic (May 28)	SWE-Bench Verified: **88.6%**. SWE-Bench Pro: **69.2%**. Terminal-Bench 2.1: **74.6%**. Parallel-subagent workflows land in Claude Code.	First model to close every case in Anthropic's Super-Agent benchmark — beating GPT-5.5. Same $5/$25 as Opus 4.7.
Jun 9, 2026 +12 days	Claude Fable 5 Mythos-class (Jun 9)	SWE-Bench Verified: **95%**. SWE-Bench Pro: **80.3%**. Dynamic Workflows: hundreds of parallel subagents from one request. Ultracode mode.	First public Mythos model. New tier — 2× Opus's price. Opus 4.8 auto-fallback for refused / safety-blocked prompts.
Jun 15, 2026 +18 days	Opus 4 · Sonnet 4 retire Deprecation deadline	Legacy Opus 4 & Sonnet 4 endpoints stop serving traffic. Anyone still pinned to those IDs breaks.	Migration to Opus 4.7/4.8 mandatory. Prompts and JSON contracts may need rework.
Jun 30, 2026 +33 days	Claude Sonnet 5 codename Fennec (Jun 30)	SWE-Bench: **82.1%**. Terminal-Bench 2.1 beats Opus 4.8. 1M context. **$2/$10** introductory pricing. Default on Free/Pro *and* Claude Code.	Sonnet-class now within reach of Opus quality — at 40% of the cost. Resets the routing calculus.

THE TAKEAWAY

In 34 days Anthropic shipped a full lineup refresh — **and made Sonnet 5 the model most teams should default to.** Opus and Fable are now escalation tiers, not first-line picks.

Sources: anthropic.com/news/claude-opus-4-8 (May 28) · llm-stats.com Fable 5 review (Jun 9) · explainx.ai Sonnet 5 launch guide (Jun 30) · mindstudio.ai Claude 4 deprecation guide.

SLIDE 06 · SONNET 5 (FENNEC) — THE NEW DEFAULT

$2 / $10 · 1M CTX

Approaches Opus quality at 40% of the cost. The Claude Code default — and usually the right answer.

WHAT SONNET 5 DOES

Positioning. "The most agentic Sonnet yet." Ties Opus 4.8 on knowledge work; trails on the hardest coding tasks.

Benchmarks. SWE-Bench Verified 82.1%. Terminal-Bench 2.1 beats Opus 4.8. Closes the Sonnet ↔ Opus gap.

Context. 1M tokens — whole-repo, day-long meeting, multi-file refactor in one prompt.

Effort parameter. On Claude Code + API it defaults to `high`. Dial down for cost, up for hard tasks.

Batch API. Up to 300k output tokens with `output-300k-2026-03-24` header.

> **Codename:** Fennec. Launched Jun 30, 2026 at introductory **$2 / $10** per 1M in/out.

WHEN TO PICK SONNET 5

Default for:

- Feature work — implement a spec across 3-10 files.
- Bug fixes with reasonable blast radius.
- Terminal / CLI agent tasks (beats Opus 4.8 here).
- Refactors up to ~1M-token context.
- Anything you would have sent to Opus 4.6.

Skip when:

- Long-horizon multi-hour agentic runs → Opus 4.8.
- Frontier-judgment tasks (novel arch, security review) → Fable 5.

> **Habit to build:** "Start at Sonnet 5. Always." Escalate only when it stalls or the token budget forces it.

Sources: anthropic.com/news/claude-sonnet-5 · vellum.ai Claude Sonnet 5 benchmarks explained · cosmicjs.com Sonnet 5 benchmarks · explainx.ai Sonnet 5 launch guide 2026.

SLIDE 07 · OPUS 4.8 — THE AGENTIC WORKHORSE

$5 / $25 · SUPER-AGENT

The only model to complete every Super-Agent case end-to-end — including beating GPT-5.5.

WHAT OPUS 4.8 DOES

Positioning. Anthropic's flagship for long-horizon *agentic* coding — real-world SWE tasks that unfold over hours.

Benchmarks. SWE-Bench Verified **88.6%** · SWE-Bench Pro **69.2%** · Terminal-Bench 2.1 **74.6%** · GDPval-AA **1890 Elo**.

Super-Agent benchmark. The only model to complete every case end-to-end — GPT-5.5 falls short here.

Parallel subagents in Claude Code. Introduced with 4.8. Dynamic workflows spin up hundreds of subagents from one request.

Effort parameter. Since 4.5. 4.7 added `xhigh`. 4.8 keeps the same knob.

Fast mode. 2.5× faster serving profile on 4.8 vs 4.7.

> **Knowledge cutoff:** Jan 2026. **Price:** $5 / $25 per 1M in/out — unchanged from 4.6/4.7.

WHEN TO PICK OPUS 4.8

Escalate to Opus 4.8 when:

- Sonnet 5 stalled or produced brittle output after 2-3 iterations.
- Task spans **many files**, **multiple systems**, or a **full-repo** change.
- You want **parallel subagents** in Claude Code (Opus 4.8's default mode).
- Novel architecture / API design where judgment matters.
- Security-sensitive reasoning (constitutional defaults matter).

Skip when:

- Sonnet 5 handled the task in 1-2 turns — no lift, real cost.
- Pure terminal / CLI agent — Sonnet 5 leads Terminal-Bench 2.1.

> **Migration:** Opus 4 & Sonnet 4 retire Jun 15, 2026. If you're pinned to those, cut over now.

Sources: anthropic.com/news/claude-opus-4-8 · llm-stats.com Opus 4.8 launch · vellum.ai Opus 4.8 benchmarks · truefoundry.com Opus 4.8 SWE-Bench Pro · vals.ai SWE-Bench leaderboard.

SLIDE 08 · FABLE 5 — THE FRONTIER TIER

$10 / $50 · MYTHOS-CLASS

First public Mythos-class model. 95% SWE-Bench Verified. And a new price tier.

WHAT FABLE 5 DOES

Positioning. Long-running autonomous work. Frontier judgment for the 2% of tasks that need it.

Benchmarks. SWE-Bench Verified **95%** · SWE-Bench Pro **80.3%** — 11 points over Opus 4.8, 21.7 over GPT-5.5.

Dynamic Workflows. A single request can spin up **hundreds of parallel subagents**, each with its own context.

Nested subagents. Depth capped at 5 to start — trees of workers, not just fan-out.

Ultracode mode. High-intensity coding profile. *Watch quota*: reports of burning 75% of a monthly plan in minutes.

Safety fallback. Refusals & safety-blocked prompts auto-swap to Opus 4.8 — you see the switch in the trace.

> **Launched:** Jun 9, 2026. **Price: $10 / $50** per 1M in/out — 2× Opus 4.8.

WHEN TO PICK FABLE 5

Reach for Fable 5 when:

- **Parallel fan-out** — 50-500 subagents on one problem (research spike, migration audit, coverage sweep).
- **Frontier judgment** — one hard architectural or security decision where being right pays for the whole month.
- **Long-running autonomous** — 4-8 hour runs where Opus 4.8 loses coherence.

Do NOT default-run when:

- Sonnet 5 or Opus 4.8 can finish the task at all.
- You haven't set a per-task credit ceiling.

> **Rule:** Fable 5 is the *orchestrator*. It coordinates cheaper models on subagents; the orchestrator alone runs on Fable 5, subagents on Sonnet/Haiku.

Sources: llm-stats.com Fable 5 review · mindstudio.ai Fable 5 dynamic workflows · developersdigest.tech Fable 5 orchestrator playbook · productcompass.pm Fable 5 guide v3 · anthropic Fable 5 & Mythos 5 system card.

NUMBERS & READS

Fable 5 leads. But the Sonnet 5 ↔ Opus 4.8 gap is small on most tasks.

BENCHMARK	SONNET 5	OPUS 4.8	FABLE 5	READ
SWE-Bench Verified	82.1%	88.6%	95.0%	Fable 5 leads
SWE-Bench Pro (harder, less contaminated)	~60%*	69.2%	80.3%	Gap widens
Terminal-Bench 2.1	Leader*	74.6%	–	Sonnet 5 beats Opus
Super-Agent (Anthropic internal)	partial	every case	every case	Opus 4.8 signature
Context window	1M	1M	1M	Same across the trio
Batch output (with header)	300k	300k	300k	All 4.5+ models

THE THREE READS THAT MATTER

1. The Sonnet 5 ↔ Opus 4.8 gap on Verified is only **6.5 points**. 2. The gap widens on Pro (harder) and shrinks (or inverts) on Terminal-Bench. 3. Fable 5 is a real step up on Pro (11 pts over Opus) — but only pays for itself on truly hard problems.

Sources: morphllm.com Claude Benchmarks 2026 · contracollective.com GPT-5.5 vs Opus 4.8 · vals.ai SWE-Bench · llm-stats.com Opus 4.8 · explainx.ai Sonnet 5 launch guide. *Sonnet 5 Pro / Terminal-Bench figures vary by run.

SLIDE 10 · THE PRICING LADDER — THREE TIERS, THREE ORDERS OF MAGNITUDE

$ PER 1M TOKENS

Sonnet 5 : Opus 4.8 : Fable 5 = 1 : 2.5 : 5 on input. On output, 1 : 2.5 : 5.

SONNET 5 · WORKHORSE

$2

input · $10 output

Illustrative task: 200k in / 20k out

Cost: $0.40 + $0.20 = **$0.60**

Effort default: `high` on Claude Code.

> Priced to be the default. 40% of Opus with 90% of the quality on typical tasks.

OPUS 4.8 · AGENT

$5

input · $25 output

Same task: $1.00 + $0.50 = **$1.50**

Multiplier vs Sonnet 5: ~2.5×

Real-world tokens: Opus tends to burn ~40% fewer tokens than Sonnet at equivalent depth.

> Sometimes cheaper end-to-end than Sonnet — because it needs fewer turns.

FABLE 5 · FRONTIER

$10

input · $50 output

Same task: $2.00 + $1.00 = **$3.00**

But Dynamic Workflows can spawn 100+ subagents — real bills reach $50-200 per hard task.

Multiplier vs Sonnet 5: 5× base — often 20-30× effective on Ultracode.

> Budget per task, not per month. Set a hard credit ceiling in Claude Code.

COST PER FINISHED TASK ≠ COST PER TOKEN

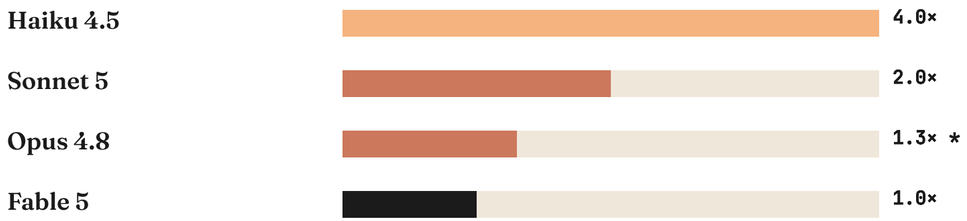
A "cheaper" model that needs 3× the turns is more expensive end-to-end. Opus 4.8 sometimes *beats* Sonnet 5 on total cost by finishing in one shot. Fable 5 rarely does — it multiplies tokens through subagents. Measure **$ per merged PR**, not $ per 1M.

Sources: platform.claude.com/docs pricing · anthropic.com/news/claude-sonnet-5 · credyt.ai Fable 5 pricing (Jun 9, 2026) · finout.io Opus 4.8 pricing 2026 · reddit r/ClaudeAI Sonnet 4.6 vs Opus 4.6 token counts.

WALL-CLOCK & VERBOSITY

Two invisible taxes: wall-clock and tokens produced. Both cost money.

SPEED · RELATIVE WALL-CLOCK



Relative speed on a fixed prompt. **Haiku ≈ 2× Sonnet 5 ≈ 4× Fable 5** in tokens/second.

* Opus 4.8 added a **2.5× fast mode** vs 4.7 — the ordering above assumes standard mode. Fast mode brings Opus close to Sonnet's throughput.

> **Why it matters for Claude Code:** a 20-step agent loop on Sonnet is ~half the wall-clock of the same loop on Fable 5 — and the human is watching.

VERBOSITY · TOKENS PER TASK

The pattern: lower tiers burn MORE tokens per task — because they need more back-and-forth.

> **Field data (production office refactor):**
> - Sonnet 4.6 → ~260M tokens
> - Opus 4.6 → ~160M tokens (~40% fewer)
> - Net result: **Sonnet cost more $** despite lower per-token price.

Fable 5: uses **1.0-1.35× more tokens** than Opus 4.8 for the same text output, because Dynamic Workflows fan-out into subagents that each pay context tax.

> **Rule of thumb:** the cheaper model wins on cost only when the task fits in 1-2 turns. Past that, escalate.

Sources: morphllm.com Claude Code models (Sonnet 1.5-2× faster than Opus, Haiku ~2× Sonnet) · reddit r/ClaudeAI Sonnet 4.6 vs Opus 4.6 token counts · llm-stats.com Opus 4.8 (2.5× fast mode) · reddit r/ClaudeAI Sonnet 5 breakdown (1.0-1.35× token ratio).

CLI · IDE · SDK

Claude Code makes routing decisions for you — knowing them is half the fight.

FEATURE · DEFAULT · WHY

Main session → **Sonnet 5** at `effort: high`.
Fastest path to near-Opus quality on daily work.

Explore subagent → **Haiku 4.5**.
Fast, read-only codebase search — grep with judgment. Cheap by design.

Plan mode → **Sonnet 5** (opt in to Opus/Fable manually).
Planning is Sonnet-shaped — one long structured pass, not a fan-out.

Parallel subagents → **Opus 4.8** orchestrator + Sonnet/Haiku workers.
Introduced with Opus 4.8 in Claude Code. Nested subagents cap at depth 5.

Dynamic Workflows → **Fable 5** orchestrator, hundreds of subagents.
Opt-in. Set a credit ceiling before you press enter.

THE FIVE COMMANDS YOU ACTUALLY USE

`/model sonnet-5`
Reset to default. Do this at the top of every new session.

`/model opus-4-8`
Escalate for multi-file / long-horizon work.

`/model fable-5`
Frontier tier. Set a credit ceiling first.

`/effort low | high | xhigh`
Dial capability↔cost within the current model. xhigh added with Opus 4.7+.

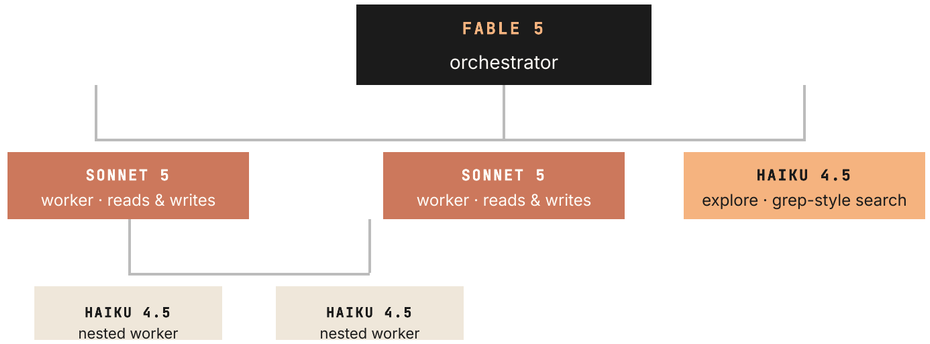
`/subagents on | off`
Toggle parallel-subagent workflows. Off = plain single-agent loop.

Sources: platform.claude.com/docs models overview (Sonnet 5 defaults to high on Claude Code) · developersdigest.tech Fable 5 orchestrator playbook (Explore subagent pinned to Haiku) · totalum.app Claude Code subagents 2026.

FABLE 5 SIGNATURE MOVE

Fan-out is the new frontier — but only if the orchestrator is on Fable 5 and the workers are cheap.

THE TOPOLOGY · ORCHESTRATOR + WORKERS



Depth cap: 5 levels (Fable 5 launch). Any deeper and coherence collapses.

Isolation: each subagent has its own context window — no cross-contamination.

> **The pattern:** Fable 5 *coordinates*; the workers do the reading and writing on cheaper tiers.

WHERE FAN-OUT PAYS FOR ITSELF

Migration audit. Sweep every file in the repo for a deprecated API. 500 files → 500 subagents → one summary in 8 min instead of 4 hours.

Coverage sweep. "Which endpoints lack input validation?" Fan out over the router table, converge into a ranked list.

Bug bisect. Run failing test against 20 candidate commits in parallel. Return the offender.

Doc-drift check. Compare code to docs across 100+ pages simultaneously.

> **Where it does NOT pay:** single-file refactors, feature work that has a natural sequential order, anything a normal 1-agent loop finishes in 5 minutes.

Sources: mindstudio.ai Fable 5 dynamic workflows parallel sub-agents · developersdigest.tech Fable 5 orchestrator playbook · totalum.app Claude Code subagents 2026 · x.com PawelHuryn "nested subagents landed, depth=5" (Jun 2026).

SLIDE 14 · THE EFFORT PARAMETER — THE SECOND DIAL NOBODY TALKS ABOUT

SINCE OPUS 4.5

Model choice is one dial. Effort is the other — and it's often the right one to move first.

LEVEL	WHAT IT DOES	TOKENS · LATENCY	WHEN TO USE
low `effort: "low"`	Minimal reasoning. One-shot answer. No hidden CoT tokens.	Cheapest. Sub-second responses on Sonnet 5.	Quick edits, autocomplete-like tasks, chatty UI where latency > quality.
medium `effort: "medium"`	Balanced thinking budget. Some hidden reasoning, planning per step.	~2× tokens vs low. Multi-second responses.	Standard feature work. The "just answer well" tier for most day-to-day.
high `effort: "high" · DEFAULT`	Extended reasoning. Multi-step planning before each output.	~4-6× tokens vs low. This is what Claude Code defaults Sonnet 5 to.	Real coding, cross-file work, anything you'd send to Opus 4.6 last quarter.
xhigh `effort: "xhigh"`	Maximum thinking budget. Deep multi-hop reasoning. Added in Opus 4.7.	~2× tokens vs high. Slower responses.	Hard bugs, novel architecture — where dialling up beats switching model tiers.

Cheap escalation ladder: Sonnet 5 medium → Sonnet 5 high → Sonnet 5 xhigh → Opus 4.8 high. Move up the *effort* ladder before the *model* ladder — often solves the task at ~40% of the cost of jumping straight to Opus.

Sources: platform.claude.com/docs models overview · anthropic.com/news/claude-opus-4-7 (xhigh effort tier) · hidekazu-konishi Anthropic release timeline (effort parameter since Opus 4.5) · mindstudio.ai Fable 5 agentic workflows.

SLIDE 15 · TASK → MODEL — 10 REAL CLAUDE CODE JOBS, ONE RECOMMENDATION EACH

THE DECISION TABLE

If your task doesn't match a row, default to Sonnet 5 at effort high.

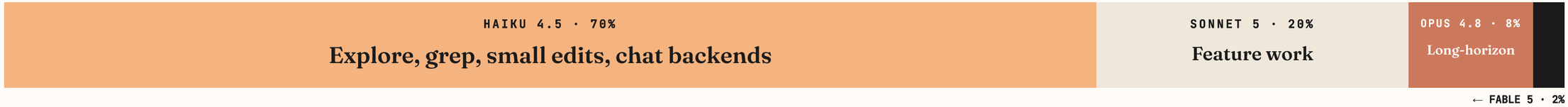
TASK	MODEL	EFFORT	WHY
01 · Grep-style search "who calls this function?"	Haiku 4.5	`low`	Read-only search. Cheap by design.
02 · Add a new endpoint + tests (3-5 files)	Sonnet 5	`high`	Bread-and-butter feature work.
03 · CLI / shell agent (curl → parse → jq → HTTP call)	Sonnet 5	`high`	Leads Terminal-Bench 2.1.
04 · Bug fix — Sonnet 5 stalled after 2 turns	Sonnet 5	`xhigh`	Try dialling effort *before* switching model.
05 · Full-repo refactor (20-50 files, 2 systems)	Opus 4.8	`high`	Fewer turns = cheaper end-to-end.
06 · 4-hour autonomous agent (self-driving)	Opus 4.8	`xhigh`	Super-Agent — only model to close every case.
07 · Security review of one PR	Opus 4.8	`xhigh`	Judgment + constitutional defaults win.
08 · Repo-wide migration audit (500+ files, parallel)	Fable 5	`high (orch)`	Dynamic Workflows earn their bill here.
09 · Novel architecture design (greenfield service)	Fable 5	`high`	Frontier judgment. One decision pays for month.
10 · Chat UI backend "explain this error message"	Haiku 4.5	`low`	Latency matters, quality bar is low.

Pattern: escalate one dial at a time — *effort* first, then *model*. Log the reason next to the escalation so you can review the routing later.

Sources: platform.claude.com/docs models overview · morphllm.com Claude Code models 2026 · augmentcode.com AI model routing guide · dev.to suraj "I ran Claude Code on every model" · mindstudio.ai model routing.

THE STARTING SPLIT

Aim for this mix. Then measure and adjust — it's a starting point, not a law.



THE FOUR TIERS · WHAT THEY COVER

Haiku 4.5 · 70% — explore subagents, grep-like search, chatbot backends, high-volume classification, "explain this error" endpoints. The invisible workhorse.

Sonnet 5 · 20% — feature work, mid-size refactors, terminal agents, code review. Your daily driver in Claude Code.

Opus 4.8 · 8% — long-horizon multi-file work, autonomous agents, security-sensitive judgment, parallel-subagent orchestration.

Fable 5 · 2% — hard-to-catch bugs, novel arch decisions, repo-wide fan-out. Rare, precious, budgeted.

DO THE MATH · BLENDED COST PER 1M OUTPUT

```
70% × $5   =  $3.50 (Haiku)
20% × $10  =  $2.00 (Sonnet 5)
 8% × $25  =  $2.00 (Opus 4.8)
 2% × $50  =  $1.00 (Fable 5)

Blended: ~$8.50 / 1M output
```

"All Fable 5": 100% × $50 = $50 / 1M
→ 5.9× cheaper with routing.

Same output quality on 98% of tasks. And most of the 2% top tier *is* Fable 5 — where quality still wins.

Sources: medium.com naresh "Which Claude Model Should You Actually Use in 2026" (70/20/8/2 routing split) · pub.towardsai Haiku vs Sonnet vs Opus tiered setup · mindstudio.ai AI model routing.

FABLE 5 · QUIET FEATURE

Fable 5 silently swaps to Opus 4.8 on refusals. Great UX. Terrible if you're not watching.

HOW THE SAFETY SWAP WORKS

Trigger. Fable 5 detects a prompt/context near a safety refusal, or a policy-blocked instruction inside a subagent trace.

Behaviour. The request is silently re-run on **Opus 4.8**. Response comes back with a `fallback_model` field in the trace.

Why. Anthropic wants Fable 5 to hit hard problems *without* becoming the safety-refusal frontier — Opus 4.8's constitutional defaults are more mature.

Cost impact. You're billed at **Opus 4.8 rates** for the swap step — cheaper than Fable, still more than expected if you didn't budget for it.

> **What you must do:** log the `fallback_model` field. When it fires unexpectedly, your prompt probably drifted into a policy-sensitive area.

THE FIVE FALLBACKS YOU SHOULD WIRE UP

01 **Rate-limit fallback.** Opus 4.8 → Sonnet 5. Don't stall the pipeline when Anthropic tightens headers.

02 **Timeout fallback.** Fable 5 → Opus 4.8 → Sonnet 5. If a Dynamic Workflow exceeds N seconds, degrade gracefully.

03 **Cost-ceiling fallback.** Per-task credit ceiling → downgrade one tier once you hit 70%.

04 **Safety refusal fallback.** Fable's built-in. Log it so you can audit.

05 **Deprecation fallback.** Opus 4 & Sonnet 4 retire Jun 15, 2026. Any code still pinned there needs a hard cutover to 4.7/4.8 / Sonnet 5.

> **Rule:** the fallback ladder points *down* the price ladder, not up. Never let a rate-limit surprise-upgrade you to Fable 5.

Sources: atlascloud.ai Fable 5 review (Opus 4.8 safety fallback) · productcompass.pm Fable 5 guide v3 (the safeguard that swaps your model) · mindstudio.ai Claude 4 deprecation migration guide · platform.claude.com/docs models overview.

WHAT NOT TO DO

If you recognise your workflow in any of these — fix it this week.

AP1

Frontier by default

Every task on Fable 5 "to be safe". Burns 75% of quota in minutes. **Fix:** start at Sonnet 5. Escalate only when it stalls.

AP2

Sonnet-only for everything

Refusing to escalate. Sonnet 5 grinds 8 turns on a task Opus 4.8 solves in 2. **Fix:** after 3 failed turns, jump one tier — *with* reason logged.

AP3

Ignoring the effort dial

Jumping model tier before touching `effort`. Skipping the cheaper escalation path. **Fix:** ladder is effort → model. xhigh Sonnet 5 often beats high Opus 4.8.

AP4

Fable 5 as sole worker

Running Fable 5 for both orchestrator *and* workers. Multiplies token bill 5× with no lift. **Fix:** Fable 5 orchestrates, cheaper tiers do the work.

AP5

Pinning to a retired model

Still hard-coding Opus 4 / Sonnet 4 in prod. They retire Jun 15, 2026. **Fix:** put model IDs in SSM/env, not in source. Migrate now.

AP6

No credit ceiling on Fable 5

Ultracode + Dynamic Workflows with no per-task budget. One hard task can drain a plan. **Fix:** set a per-task credit ceiling. Alert at 70%.

Sources: rentierdigital "Fable 5 burned 75% quota" · productcompass.pm Fable 5 guide v3 · mindstudio.ai Claude 4 deprecation guide · dev.to toffy "Use Fable 5 where it pays for itself" · medium.com addyosmani LLM coding workflow 2026.

WHERE ARE YOU?

L2 is where the routing math starts paying. Aim for L3 by end of Q3.

L0

Everyone on Opus

Signals: No routing. Every session on Opus 4.7/4.8 "to be safe".

Bill: 3-4× what it should be.

Risk: Fable 5 arrives, team switches, bill 8×.

Next move: Set Sonnet 5 as the default in Claude Code.

L1

Sonnet-default

Signals: Sonnet 5 is the daily driver. Manual /model switch when it stalls.

Bill: 40-60% of L0.

Risk: Under-escalation — Sonnet grinding tasks Opus would finish faster.

Next move: Add Haiku for read-only / explore work.

L2

Tiered routing

Signals: Haiku + Sonnet 5 + Opus 4.8 mix. Effort dial in use. Fable 5 as escalation only.

Bill: ~20% of L0.

Risk: Routing rules drift — no measurement.

Next move: Instrument $ per PR by tier. Publish weekly.

L3

Measured routing

Signals: $ / merged-PR tracked per tier. Weekly review. Escalation reasons logged.

Bill: Predictable per feature.

Risk: Over-instrumentation slows the team.

Next move: Build a Fable 5 playbook for the 2%.

L4

Routing as platform

Signals: Routing rules in code. Fallback ladders. Per-task credit ceilings. Fable 5 orchestrator + cheap workers is the default agent shape.

Bill: Grows sub-linearly with team size.

Next move: Publish routing manifest for other teams to fork.

Synthesised ladder — not an official framework. Anchors: medium.com naresh 70/20/8/2 · augmentcode.com AI model routing guide · dev.to suraj Claude Code on every model.

MONDAY MORNING

If a stranger opened your team's Claude Code bill —

which model would they think you use for everything?

SONNET 5

You're at L1-L2. Good defaults. **Next play:** add Haiku 4.5 for explore/search and instrument $ per PR by tier.

OPUS 4.8

You're at L0. Bill is 3-4× what it should be. **Next play:** switch default to Sonnet 5 this sprint. Aim for L2 in 30 days.

FABLE 5

Something's off. Frontier-by-default is expensive with no lift on 80% of work. **Next play:** reserve Fable 5 for <2% of tasks with a credit ceiling.

The right answer is "a mix" — 70% Haiku, 20% Sonnet 5, 8% Opus 4.8, 2% Fable 5. Anything else is a routing bug.

Sonnet 5 launched Jun 30, 2026. Opus 4.8 launched May 28, 2026. Fable 5 launched Jun 9, 2026. Opus 4 & Sonnet 4 retire Jun 15, 2026.