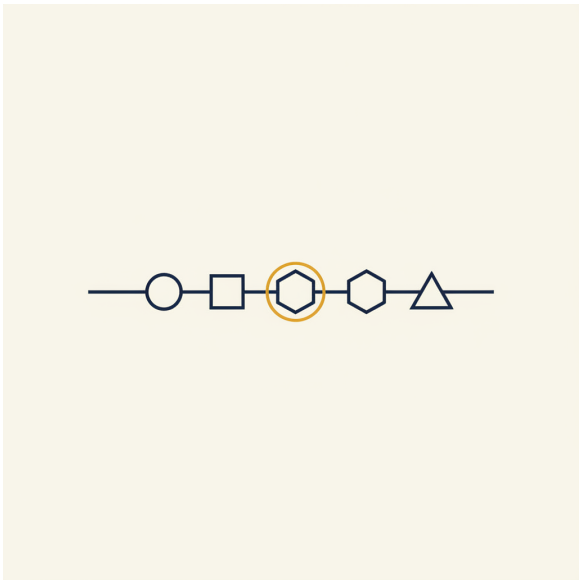
TIER 2 - PROFESSIONAL * V1.0 -- MAY 2026

# PICKING THE RIGHT MODEL

ChatGPT vs Claude vs Gemini vs the open-source pack. Cost-per-task, model strengths, switching costs, and the per-task economics from a real operator running multiple models in production.




**BY** Alex Jahn / Agent Logic



v1.0 -- May 2026 **EDITION**

Operators paying for AI who don't know whether they're overpaying or under-equipped **AUDIENCE**

15-20 minutes **READ TIME**

Free. Forever. **COST**



*Prepared by Agent Logic / alexanderjahn79@icloud.com / theaiguywi.com*

CONTENTS

# What's in here

SECTION 1

# The "use the smartest model" trap

## Most operators are paying 5x to 10x what their tasks actually need

You signed up for ChatGPT or Claude six months ago. You picked the most expensive plan because the marketing said it was "the best." You've been using it for everything -- quick lookups, drafting emails, summarizing meetings, the occasional heavy analysis.

The most expensive plan was the right call for the *lookups* and the model would handle just as well. Across a year, that's hundreds to thousands of dollars in overpayment for the same output.

The math gets more painful when teams scale. Ten employees on the top tier of ChatGPT Enterprise vs ten employees on the right AI spend, with no measurable difference in output quality.

This module is the fix. Not "switch to the cheapest thing" -- that's the equally-wrong opposite mistake. The fix is needs.

By the end of this primer:

- You'll have a map of the model landscape in mid-2026 -- what's available, what's good for what.
- You'll have the no single model is right for everything.
- You'll have a per-task heuristic for picking models on the fly.
- You'll see a and Agent Logic.
- You'll know how to evaluate

> ***The right model isn't the smartest one. It's the one that fits the task and the budget. People who default to the smartest model for everything are leaving 70% of their AI budget on the table.***

SECTION 2

# The model landscape in mid-2026

A quick map. One paragraph each, plain English. Skip the version numbers -- they're stale by the time you read this.

## ChatGPT (OpenAI)

The household name. Available as consumer subscription (ChatGPT Plus / Pro / Team / Enterprise) and as raw API access. Strong general-purpose model with a deep ecosystem (plugins, custom GPTs, voice, image generation built in). The default choice for most knowledge workers. Pricing: $20-200/month consumer; pay-per-token API.

## Claude (Anthropic)

The other big name. Same shape as ChatGPT -- consumer subscription and API. Notable strengths: long context handling, careful following of detailed instructions, code-friendly output. Three model tiers -- Haiku (fast/cheap), Sonnet (balanced default), Opus (heaviest reasoning). My personal default for production work. Pricing similar to OpenAI.

## Gemini (Google)

Google's flagship. Strongest when integrated with the Google Workspace ecosystem (Gmail, Drive, Docs, Calendar). Has a free tier that's genuinely useful, which most other providers don't. Specialized variants for image generation (Gemini Flash Image), heavy reasoning (Gemini Pro), and long-document handling. Worth knowing if your business runs on Google.

## Llama and the open-source pack (Meta + others)

Cheapest by far. The Llama models from Meta are genuinely capable; many smaller competitors (Mistral, DeepSeek, Qwen) sit in the same ballpark. Run them via OpenRouter or any provider that hosts open-source models. Pricing per token is 5-50× cheaper than the top-tier Claude/ChatGPT models. Worth knowing for anything cost-sensitive at scale.

### Specialized models

For some tasks, a general-purpose chat model is the wrong tool. Image generation has its own ecosystem (DALL-E, Stable Diffusion, Gemini Flash Image, Midjourney). Voice transcription has Whisper and faster-whisper. Code-specific models (Cursor's models, GitHub Copilot's) outperform general models on coding tasks. If you have a specialized task, check whether a specialized model is faster, cheaper, or better -- often it's all three.

> **The mid-2026 default mental model:**

- *Don't know? Use Claude Sonnet or ChatGPT (paid tier).* Both are solid general defaults.
- *Cheap, simple, high-volume tasks?* Llama 8B or Claude Haiku.
- *Heaviest reasoning, novel synthesis?* Claude Opus or ChatGPT's top tier.
- *Image generation?* Gemini Flash Image or DALL-E.
- *Living in Google Workspace?* Gemini integrates better than the alternatives.

That's enough to make 90% of your daily decisions. The rest of this module sharpens the 10% where it really matters.

SECTION 3

# The three axes -- cost, capability, latency

### Pick any two

Models live in a three-way tradeoff between cost, capability, and latency (response speed). You don't get to optimize all three. Bigger models are smarter, slower, and more expensive. Smaller models are cheaper, faster, and dumber. Specialized models are great at one thing and bad at others.

> **The three-axis tradeoff:**

| Want | Tradeoff |
|---|---|
| **Cheap + f** | tasks. |
| **Cheap + c** | chain-of-thought. |
| **Capable +** | **All three** | Doesn't exist. Anyone selling you this is either lying or using a small specialized model on |

a narrow task. |

The implication: there's no "best model." There's a best model *for this tas*
*requirement*. The whole module is built on this point.

### Where each axis bites

- **Cost** matters when you're running a task at high volume. A $0.05-per-call cost is fine if you call it once a day. It's a $1,800/month problem if you call it once a minute (cron jobs, bulk processing, customer-facing chat).
- **Capability** matters for the hard tasks. Heavy reasoning, novel synthesis, anything where being wrong is expensive. Don't cheap out on these. The top-tier model costs 10× more -- it's worth 10× more on tasks where the output quality is load-bearing.
- **Latency** matters for human-facing interactions. If your customer is staring at a chat window waiting for a response, 8 seconds is acceptable, 30 seconds is brand damage. If a cron job is processing in the background overnight, latency is irrelevant.

Pick which axis you're optimizing for *per task*, a

SECTION 4

## Per-task model picking -- the heuristic

### Match task complexity to model size

The simplest heuristic that holds up in practice:

> **The four-tier task heuristic:**

1. **Trivial tasks** (categorize, extract, simple lookup) -> smallest available model. Llama 8B, Haiku, Gemini Flash. *Cost: ~$0*
2. **Standard tasks** (drafting, summarizing, code review, light analysis) -> mid-tier model. Claude Sonnet, GPT-4 (default), Llama 70B. *Cost: ~$0*
3. **Heavy reasoning** (multi-step planning, novel synthesis, deep analysis, anything where being wrong is expensive) -> top-tier model. Claude Opus, GPT-4 (top tier). *Cost: ~$0*

4. **Specialized tasks** (image generation, voice transcription, OCR, code-specific work) -> use the specialist. Gemini Flash Image, Whisper, Cursor's coding model, etc. *Cost varie*

90% of your daily tasks fit tier 1 or 2. The expensive tier 3 model should be the exception, not the default.

## Concrete examples per tier

**Tier 1 -- Trivial:**

- *"Is this email about a project status update or about a bug report?"* (categorization)
- *"Extract the names of any people mentioned in this paragraph."* (extraction)
- *"Is this a 5-star or a 1-star customer review?"* (sentiment classification)
- *"Convert this date string to ISO format."* (transformation)

**Tier 2 -- Standard:**

- *"Draft a 200-word follow-up email to this customer."*
- *"Summarize this 2-page meeting transcript into 5 bullet points."*
- *"Review this customer-facing doc for tone."*
- *"Help me think through whether this is a good vendor offer."*

**Tier 3 -- Heavy:**

- *"Synthesize three industry reports into a recommendation about whether we should expand to a second location."*
- *"Read this 50-page contract and flag every clause that's unusual or unfavorable."*
- *"Reverse-engineer the pricing strategy this competitor is probably running based on these data points."*

**Tier 4 -- Specialized:**

- Image generation (cover art, illustrations).
- Voice transcription (call recordings, voice memos).
- OCR (scanned documents, photos of forms).
- Code-specific work (refactoring, code review at depth).

If you can't tell which tier a task belongs in, default to tier 2. You'll be wrong sometimes -- too smart for trivial tasks, not smart enough for heavy ones -- but it's the safe middle.

SECTION 5

# Worked example -- a real operator's stack

## The mix I actually run in production

I'll be specific because abstract talk is useless. Here's the actual model mix I run across my carpentry business + Agent Logic agent stack, as of May 2026:

> **My production model stack (May 2026):**

- **Llama 3.1 8B** -- All cron-job triage tasks. Daily news scoring, lead categorization, scan organization. ~$0.0001/call. Runs hundreds of times a day.
- **Claude Haiku 4.5** -- Fast user-facing responses on Telegram. Routine assistant chat where speed matters more than the deepest possible reasoning. ~$0.001/call.
- **Claude Sonnet 4** -- Daily ops default. Most proposal drafting, customer email handling, anything that needs decent reasoning but runs at moderate volume. ~$0.01/call. The workhorse.
- **Claude Opus 4** -- Heavy reasoning only. Complex multi-step pipeline work. Code architecture decisions. The hardest debug sessions. ~$0.10/call. Used sparingly -- maybe 5% of total calls but 30% of total cost.
- **Llama 3.1 70B** -- Bulk batch processing. When I need decent quality but at high volume. ~$0.005/call.
- **Gemini 2.5 Flash Image** -- Cover hero illustrations for distribution PDFs (including this one). ~$0.04/image. Specialized.

That's six models in active use. They're all reachable through one provider (OpenRouter), so switching between them is just changing a config string.

## The economics, honestly

Last month's bill across the whole stack: roughly $35-50, depending on volume. That's running:

- 20+ daily cron jobs
- Multi-agent system handling proposals, invoices, scheduling
- Voice phone agent
- Telegram bridge for ops
- All distribution-PDF cover image generation
- Plus my personal direct chat use

If I ran everything on Claude Opus, that bill would be roughly $300-500/month for the same workload. The 6-10× savings comes entirely from per-task model picking. No tradeoff in output quality -- Opus would be wasted on cron triage where Llama 8B is fine.

**6**

**Models in my production stack.**

That sounds complicated. It isn't -- they're all accessed through one provider with one API. The config is a dictionary mapping task type to model. Setup time: an afternoon. Lifetime savings: hundreds of dollars per month, forever.

## How to design your own stack

You don't need six models. Most small businesses can run on three:

- **Cheap workhorse** for high-volume, low-stakes tasks (Llama 8B, Haiku, Gemini Flash).
- **Default workhorse** for daily ops (Sonnet, GPT-4 default tier).
- **Heavy hitter** for the rare hard tasks (Opus, GPT-4 top tier).

Plus specialized tools for image / voice / OCR / code as needed.

Three tiers, used right, gets most of the savings of a six-model stack. You can always add more as you find tasks that don't fit the three.

SECTION 6

# Switching costs -- when they're real and when t

## Three real switching costs

If you're considering switching from one model or vendor to another, these are the costs that matter:

**1. Workflow-embedded prompts.** Your prompt library is tuned for one model's quirks. Some prompts that work great on Claude produce mediocre output on ChatGPT, or vice versa. Switching means re-testing every template and adjusting where needed. For a small library (5-15 templates), this is half a day of work. For a large one, it's a real project.

**2. Data-flow contracts.** If you have an enterprise contract with a specific vendor (DPA signed, SOC 2 reviewed, billing integrated), switching means re-doing all of that. Procurement teams hate this. Legal teams hate this more. Real cost: weeks to months for enterprise deals.

**3. Tool integrations.** If your vendor's model is wired into N other systems -- your CRM, your email, your code editor, your custom apps -- switching means re-wiring all of them. The deeper the integration, the higher the switching cost.

## Three fake switching costs

Costs that *feel* like re

**1. "I've gotten used to it."** Habit, not lock-in. Three days of using a new model and the muscle memory transfers. If the new model is genuinely better for your tasks, this is the weakest possible reason to stay.

**2. "My ChatGPT history is there."** History export is universally available. You can move your history if it actually matters (it usually doesn't -- old prompts are rarely the load-bearing artifact).

**3. "What if the new model is worse?"** You can run both side-by-side for a month. Most providers offer trial access. The cost of evaluating is much lower than the cost of being wrong about staying.

## How to evaluate before switching

The minimum-viable switching evaluation:

1. **Pick the 5 highest-volume prompts** in your library. These are where switching costs will hurt most if it goes wrong.
2. **Run them on both models** for a week. Compare outputs side-by-side.
3. **Calculate the cost difference** at your actual volume.
4. **Make the call** based on (a) output quality difference, (b) cost difference, (c) any of the three real switching costs above.

If the new model is meaningfully better and the savings are real, switch. If output is roughly equal and there's no cost advantage, don't bother.

SECTION 7

# The privacy / model-tier overlap

A bridge to the next module.

Different models have different data-handling postures, which means **the privac**
**your task constrains which model you can use.** Public data -> any model is fine. Internal -> most consumer tiers OK with care. Confidential -> enterprise tiers only. Regulated -> enterprise + DPA, or don't use AI at all.

This means model picking isn't just a cost+capability question. It's a *cost + cap*
*data-classification* question. The cheapest model that handles the task may not be approved for the data the task uses.

This is the entire premise of Module 12 (*Privacy and What Not to Paste*), the next and last Tier 2 module. If you're an operator who's about to roll AI out across a team, that module is mandatory reading. The model-picking framework in this module gives you the cost+capability axis. Module 12 gives you the data-classification axis. Together they make the picks defensible.

SECTION 8

# Where to go from here

You're now five modules into Tier 2. One module left in the tier:

- **Privacy and what not to paste** -- the workplace data classifications, consumer vs enterprise tiers, the 1-page AI usage policy, the SOC 2 conversation.

After Tier 2 closes, Tier 3 (Employable) is next -- the operator-level skills that make you the AI-fluent person on a team.

***Get the next module the day it drops: theaiguywi.com/training***

One email per release. No drip. No spam. Opt out anytime.

If you want me to design and install the right model stack for your business -- pick the tiers, configure the routing, build the cost-tracking dashboard, train your team on which task uses which model -- that's the consulting offer. Same approach I run on my own carpentry business stack: per-task picking, real economics, no overpaying for capability you don't need.

***Reach out: alexanderjahn79@icloud.com***

A short call. Honest scope. We figure out together if it's a fit.

## Closing -- the lock-in line

If you remember nothing else from this module:

# 10

**Times most operators overpay for AI by always using the top-tier model.**

Per-task model picking -- match task complexity to model size -- recovers most of that. Three tiers (cheap workhorse, default, heavy hitter) plus specialized tools is enough for most small businesses. The rest is execution.

You have the framework. The next module makes sure your model picks survive contact with your company's data.

**Agent Logic** -- *theaiguyw*
Fond du Lac, WI. This is module 5 of 6 in Tier 2 (Professional).