



RESOURCE-EFFICIENT ENGINEERING · WHITE PAPER

# The Running Cost of AI

Keeping it low and predictable after the pilot

---

**Rebecca Enonchong, FEng** · Founder & CEO, AppsTech Labs

**Ebot Tabi** · Chief Research Officer, AppsTech Labs

September 2026

**AppsTech Labs**

Breakthrough Technology. Disciplined Engineering.

[appstechlabs.com](https://appstechlabs.com)

## INTRODUCTION

# Running cost after the pilot

---

A company that adopts artificial intelligence (AI) usually begins with a pilot, a trial on a small volume of work. Building a pilot can be a real expense, especially for a smaller company. Its running cost, the monthly bill for using the AI model, generally stays small, because the pilot handles a few requests a day. Once the system is in daily use across the company, it may read every supplier invoice and answer every customer message. The quality of the answers can hold steady while the monthly bill grows with each new use.

This paper explains where the running cost of an AI system comes from and how to keep it low.

AppsTech Labs calls its approach Resource-Efficient Engineering. For every system, our engineers measure what it costs to process one transaction, such as one invoice.

## THE DIAGNOSIS

# Four causes of rising AI costs

---

**Per-token billing.** An AI model is the program that reads a request and writes the answer. The models behind tools such as ChatGPT, Claude and Gemini are large language models, or LLMs. Companies usually rent access to an AI model from a provider such as Anthropic, Google or OpenAI, which charges by the token. A token is a fragment of text, often a short word or part of a longer one. The AI model provider counts every token sent to the model and every token the model writes back, on each request. On the current price lists of all three, a token written by the model costs between five and about eight times as much as a token it reads<sup>[1][2][3]</sup>.

**Growth in use.** A pilot usually starts in one team, such as accounts payable. Once it works, other teams ask for it, and new tasks are added, such as reading contracts or answering customer emails. Each new use adds requests, and the AI model provider bills every request. As opposed to a software license, which has a fixed annual price, pay-per-use pricing moves with volume. A tenfold rise in requests brings a tenfold rise in the bill, and the increase reaches the budget without passing through a purchase approval.

**Oversized requests.** A typical request carries standing instructions, several examples, whole documents and the history of the conversation. The company pays for every part of it on each request. Longer requests also reduce the reliability of the answers. A July 2025 study of 18 leading models found that their performance declined as the input grew longer<sup>[4]</sup>.

**Unmeasured cost.** In a Deloitte survey of 200 North American chief financial officers in mid-2026, 46% named cost uncertainty or lack of transparency as their biggest internal concern about their company's use of AI<sup>[5]</sup>. In the same survey, 43% pointed to insufficient visibility into AI tools or their use<sup>[5]</sup>. Without a figure for the cost of one transaction, the monthly invoice from the AI model provider is the first sign that costs have risen.

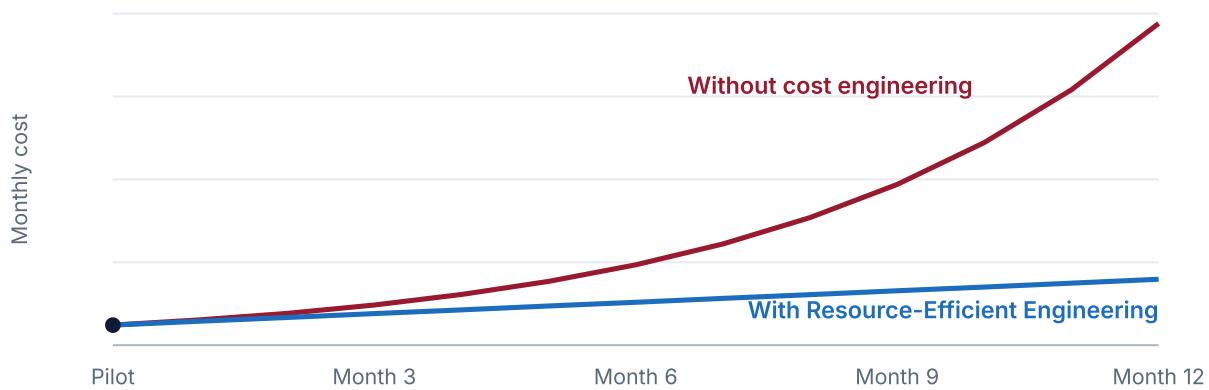


Figure 1. Monthly cost over twelve months for two systems with the same pilot cost. Illustrative.

Cloudflare reported that a model could reach all of its services with about 1,000 tokens of description, against 1.17 million by the usual method<sup>[6]</sup>.

#### THE DISCIPLINE · 1

## Prompt caching and request size

The largest savings come from what each request contains, whichever model is chosen.

AI model providers can store the fixed part of a request, such as the instructions and examples, and reuse it for a few minutes or up to an hour. This is called prompt caching. A prompt is the full text sent to the model with a request. Anthropic, Google and OpenAI all bill stored text at about a tenth of the normal input price on their main models<sup>[1][2][3]</sup>. The discount applies when the fixed part comes first and the document comes last. Our engineers therefore treat the order of a prompt as a cost decision.

A second set of savings comes from keeping bulk data away from the model. A system may retrieve ten thousand rows from a database to answer a single question. In the costly design, the model receives all ten thousand rows. In the efficient design, a short program filters the rows first and passes the model the result. In a worked example published in November 2025, Anthropic reports that this change reduced one task from 150,000 tokens to 2,000, a saving of 98.7%<sup>[7]</sup>. Cloudflare reported a similar effect in February 2026 for the descriptions of the services a model can use. Giving a model access to all of Cloudflare's services in this way took about 1,000 tokens of description, against 1.17 million by the usual method<sup>[6]</sup>.

## Model size and price

---

At September 2026 prices, the most capable AI model on Google's price list costs about seven times as much per token as the smallest. The gap is ten times at Anthropic and a hundred times at OpenAI<sup>[1][2][3]</sup>. The right size depends on the task, and our engineers settle it by testing.

The test is written first, as a set of real cases from the customer's own work, including the difficult ones, each with its correct answer. The engineers try the smallest model that could pass, and move to a larger one when the smaller model fails.

For high-volume sorting, such as classifying documents or directing messages to the right team, a small model trained for that single task often gives the best result. An independent comparison published in January 2026 ran 32 experiments on tasks of this kind. A small model trained for the task processed 277 items per second, against 12 for a small general-purpose model that writes out its answers. The trained models were also more accurate on three of the four tasks tested<sup>[8]</sup>.

## Batch processing and self-hosting

---

Some AI work can wait a few hours for its result. An overnight reconciliation of accounts is one example, and a backlog of scanned documents is another. Anthropic, Google and OpenAI all process such work at half the normal price through their batch services, which return results within a day<sup>[1][2][3]</sup>.

For models that AppsTech Labs runs on servers it manages, the same logic applies to the hardware. When a server processes many requests together, energy per token can fall by a factor of three to five, according to measurements across 46 models published by ML.ENERGY in January 2026<sup>[9]</sup>. A model installed on the customer's own servers carries no fee per question, and the customer's data stays inside its network.

## Model compression

---

Compression reduces the size of a model so that it runs faster on less hardware. In distillation, a large model trains a smaller one to reproduce its answers. In quantization, the numbers inside the model are stored with less precision.

These methods remain useful for the models we run ourselves. Their real benefit depends on how many requests the server handles at the same time, so it has to be measured under real working conditions. The January 2026 study shows this with quantization. It compared the same models in two versions, one storing its numbers in 16 bits, the usual format, and the other in 8 bits. The 8-bit version is expected to use less energy. Yet when the server handled 8 to 16 requests at a time, the 8-bit version used up to 56% more energy than the 16-bit version. The 8-bit version began to save energy once the server handled more than about 64 requests at a time, by 11% in the typical case<sup>[9]</sup>.

## Cost per AI transaction

The unit that matters to the business is the cost of one transaction, such as one invoice read or one customer message answered. Our engineers measure it by running a hundred real items through the finished system and dividing the bill by one hundred. They repeat the measurement whenever the instructions change significantly.

### WORKED EXAMPLE · SUPPLIER INVOICES

For each invoice, the system sends the model about 3,000 tokens, made up of the instructions, two examples and the text of the page. The model returns about 400 tokens, with the key details of the invoice laid out in fixed fields. On Claude Sonnet 5 or GPT-6 Sol, which carried the same list price in September 2026, one invoice costs one cent<sup>[1][3]</sup>.

When the instructions and examples are stored for reuse, about 2,200 of the 3,000 tokens are billed at a tenth of the price, and the invoice costs about 0.6 cents. When the invoices can wait for an overnight run, the batch price halves that figure, to about 0.3 cents.

| Monthly cost for 20,000 invoices           |       |
|--------------------------------------------|-------|
| Standard processing                        | \$200 |
| With prompt caching                        | \$121 |
| With prompt caching and an overnight batch | \$60  |

The three rows use the same model and produce the same answers.

*Calculated from the published prices of Anthropic and OpenAI, September 2026. The result is the same for both. Illustrative volumes.*

## THE TEST

# Five questions on AI running costs

### QUESTIONS FOR ANY AI PROPOSAL, BEFORE SIGNING AND EVERY QUARTER AFTER

1. What does one transaction cost, and who measures it?
2. How does the monthly cost change when use doubles?
3. Is the proposed model the smallest one that passes a test built on the company's own cases?
4. What does each request send to the model, and why?
5. What will the system cost in its twelfth month, compared with the first-month figure in the proposal?

A useful answer to each question is a figure or a named design decision, given in writing so that the finance team can check it.

## THE POSITION

# Resource-Efficient Engineering at AppsTech Labs

Our engineers apply all five disciplines in this paper. Each request puts the fixed instructions first so that they can be cached, and bulk data is filtered before it reaches the model. Before we choose a model, we write a test from the customer's own cases and pick the smallest model that passes it. Work that can wait runs in overnight batches. When a customer's data has to stay inside its network, the system runs on the customer's own servers. For the models we run ourselves, we use compression where measurements on the real workload show a saving. The cost of one transaction is measured for every system, and measured again whenever its instructions change.

[appstechlabs.com](https://appstechlabs.com)

Breakthrough Technology. Disciplined Engineering.

## ABOUT THE AUTHORS

**Rebecca Enonchong** is the founder and CEO of AppsTech Labs and AppsTech. She is a Fellow of the UK's Royal Academy of Engineering.

**Ebot Tabi** is chief research officer at AppsTech Labs, where he leads research and development of the company's artificial intelligence systems.

## SOURCES

1. Anthropic, Claude pricing, including prompt caching and batch rates. [platform.claude.com/docs/en/about-claude/pricing](https://platform.claude.com/docs/en/about-claude/pricing). Consulted 27 September 2026.
2. Google, Gemini API pricing, including context caching and Batch mode. [ai.google.dev/gemini-api/docs/pricing](https://ai.google.dev/gemini-api/docs/pricing). Consulted 27 September 2026.
3. OpenAI, API pricing, including cached input and the Batch tier. [developers.openai.com/api/docs/pricing](https://developers.openai.com/api/docs/pricing). Consulted 27 September 2026.
4. Chroma, "Context Rot: How Increasing Input Tokens Impacts LLM Performance," 14 July 2025. [trychroma.com/research/context-rot](https://trychroma.com/research/context-rot)
5. Deloitte, "Q2 2026 CFO Signals Survey," 200 North American CFOs at companies with at least \$1 billion in revenue, surveyed 22 May to 7 June 2026. [deloitte.com/us/en/about/press-room/deloitte-q2-2026-cfo-signals-survey.html](https://deloitte.com/us/en/about/press-room/deloitte-q2-2026-cfo-signals-survey.html)
6. M. Carey, Cloudflare, "Code Mode: give agents an entire API in 1,000 tokens," 20 February 2026. [blog.cloudflare.com/code-mode-mcp](https://blog.cloudflare.com/code-mode-mcp)
7. Anthropic, "Code execution with MCP: building more efficient AI agents," 4 November 2025. [anthropic.com/engineering/code-execution-with-mcp](https://anthropic.com/engineering/code-execution-with-mcp)
8. A. Jacobs, "Beating BERT? Small LLMs vs Fine-Tuned Encoders for Classification," 4 January 2026. [alex-jacobs.com/posts/beatingbert](https://alex-jacobs.com/posts/beatingbert)
9. ML.ENERGY, "Diagnosing Inference Energy Consumption with the ML.ENERGY Leaderboard v3.0," 29 January 2026. [ml.energy/blog](https://ml.energy/blog)