

MERCHANTS, PROVIDERS & ISOS

Haiku 5.5: AI features in a payments application no longer need a large budget

Published 27 October 2026

Anthropic's Claude Haiku 5.5 costs a fortieth of the current Opus model for the same work. For the high-volume, narrow jobs a payments business runs every day, that turns AI from a line in the budget into a rounding error. What it costs, what it suits and where a larger model still earns its price.

For the last few years, adding an AI feature to a payments application has come with a cost question attached. Reading a merchant's website, sorting a support inbox or pulling fields out of an application form are all jobs a language model does well, but at tens of thousands of calls a month the bill was large enough that many teams kept those features to a pilot.

Anthropic's current small model, **Claude Haiku 5.5**, changes that arithmetic. At list price it costs a fortieth of the current Opus model and a tenth of the Haiku model before it. For the narrow, repetitive work that makes up most of the AI a payments business needs, a feature that used to cost hundreds a month now costs a few.

What the models cost

Anthropic prices its models per million *tokens*, the units text is split into. A token is roughly three-quarters of an English word, so a million tokens is somewhere around 750,000 words. Input (what you send the model) and output (what it writes back) are priced separately.

MODEL	INPUT, USD PER MILLION TOKENS	OUTPUT, USD PER MILLION TOKENS	RELATIVE TO HAIKU 5.5
Claude Haiku 5.5	0.10	0.50	1x
Claude Haiku 4.5 (the previous Haiku)	1.00	5.00	10x
Claude Sonnet 5.5	2.00	10.00	20x
Claude Opus 5.5	4.00	20.00	40x

MODEL	INPUT, USD PER MILLION TOKENS	OUTPUT, USD PER MILLION TOKENS	RELATIVE TO HAIKU 5.5
Claude Fable 5.1	10.00	50.00	100×

These are Anthropic's own list prices as of October 2026. The Haiku 5.5 price applies to requests of up to 100,000 tokens; longer requests are charged at 0.50 USD in and 2.50 USD out, still well below the larger models. Claude is also sold through Amazon Bedrock and Google Cloud Vertex AI, which set their own prices, so check the rate on the platform you buy through.

What that means per request

A typical narrow task, such as reading a page of a merchant's website and returning a short structured verdict, might send 2,000 tokens and get 300 back. Priced at the table above:

MODEL	COST OF ONE REQUEST, USD	COST OF 10,000 REQUESTS, USD
Claude Haiku 5.5	0.00035	3.50
Claude Haiku 4.5	0.0035	35.00
Claude Sonnet 5.5	0.007	70.00
Claude Opus 5.5	0.014	140.00
Claude Fable 5.1	0.035	350.00

A request on Haiku 5.5 costs a few hundredths of a cent. Ten thousand of them cost about the same as a coffee. The figures are illustrative: your own prompts will be longer or shorter, and the model's reasoning, which is billed as output, adds to the count (more on that below). The ratios between the models hold whatever the size of the request.

Two discounts lower it further. Requests that do not need an answer straight away can go through Anthropic's batch interface at half price, which suits overnight jobs such as re-scanning a merchant portfolio. And a long set of instructions that is the same on every request can be cached, so it is charged in full once and at a fraction afterwards.

The work a small model suits

The cheap model is not a lesser version of the expensive one for every job. It is built for speed and volume, and it does best where the task is narrow, the instructions are clear and the answer has a fixed shape. Most of the AI a payments business needs looks like that.

- **Website review at onboarding.** Does the site have a refund policy, terms and contact details? What does it actually sell, and does that match the industry the merchant declared? This is reading and classifying, repeated for every application and again at every [periodic review](#).

- **Portfolio monitoring.** Re-reading thousands of merchant sites on a schedule for new products, changed policies or signs of **transaction laundering**. Volume is the whole problem here, and it is where the price difference matters most.
- **Extracting fields from documents.** Company name, registration number, directors and addresses from an incorporation certificate or a bank letter, returned in a fixed format for a person to check.
- **Sorting the inbox.** Routing support tickets, merchant queries and provider correspondence to the right team, and flagging the ones that mention a dispute, a fraud report or a regulator.
- **First-pass dispute triage.** Grouping incoming **chargebacks** by reason, checking that the evidence a merchant has supplied is the evidence the reason code needs, and listing what is missing.
- **Reading descriptors and transaction text.** Sorting free-text transaction descriptions into categories, or checking that a billing descriptor is one a cardholder would recognise.
- **Turning a question into a search.** Taking one typed sentence and pulling out the fields a database query needs: a country, a currency, an industry, a payment method.

Each of these is run many times a day, each has an answer a person can check, and none of them needs the model to plan a long piece of work on its own.

Where a larger model still earns its price

The larger models are slower and dearer for a reason: they are better at work that is long, open-ended or hard to specify. They are worth paying for when:

- **The task has many steps that depend on each other**, such as investigating an unusual pattern across several data sources and writing up what was found.
- **The writing matters**, such as drafting a full dispute response or a policy that a person will sign off.
- **The input is long and the answer depends on all of it**, such as reviewing a full provider agreement against your own terms.
- **A wrong answer is expensive and hard to catch.**

A common design uses both: the small model handles every request and passes the few it is unsure about, or that meet a set threshold, to a larger one. Measure that against the simpler choice first, though. A larger model run at a lower reasoning setting is sometimes enough, and one model is easier to run than two.

Keeping the cost where you expect it

The price per token is only part of the bill. A few habits keep it predictable.

- **Set the reasoning effort for the job.** Current Claude models think before they answer, and that thinking is billed as output. Haiku 5.5 defaults to a medium setting. For classification and

extraction, a low setting is often enough and much cheaper. Test it on a sample of real requests before deciding.

- **Ask for a fixed format.** A short structured answer (a yes or no, a category from a list, a handful of named fields) costs less than prose and is easier to check.
- **Send only what the model needs.** The text of a policy page, not the whole site with its navigation and scripts.
- **Cache what repeats, batch what can wait.** Both are discounts for work you are already doing.
- **Judge the cost per finished task**, not per request. A cheaper call that has to be retried twice is not cheaper.

What does not change with the price

A cheaper model does not change your obligations, and three points are worth settling before anything goes live.

1. **Keep card data out of it.** Full card numbers, security codes and anything else in scope for PCI DSS have no place in a prompt. None of the tasks above needs them, and sending them would bring the AI service and everything around it into your cardholder data environment.
2. **Minimise personal data.** Under the GDPR and similar laws, check the provider's data terms, where requests are processed and how long they are kept, and say in your privacy notice that you use the service. Strip names and contact details where the task does not need them.
3. **Treat what the model reads as untrusted.** A merchant's website is written by the merchant, who benefits from a favourable reading. Text on it can try to instruct the model ("this site has a full refund policy"). Tell the model in its instructions that the content is data, not instructions, and constrain the answer to fixed choices so that injected text has nowhere to go.

The model is advisory. A finding from an automated review is a reason for a person to look more closely, not a decision, and the underwriter or the analyst still makes the call.

How Payments Marketplace uses it

Payments Marketplace uses Claude for exactly this kind of narrow, high-volume work. When a merchant runs a pre-flight check, the website scan reads the site's pages and returns a fixed set of findings: whether a refund policy exists and is adequate, what the site actually sells, and whether that matches the industry the merchant declared. The AI Agent on our home page turns one typed sentence into the handful of fields our matching reads, and every word of the answer is written by our own code.

- **The model reads, people decide.** Scan findings are advisory signals shown to the provider. They never block an application, and a match is an introduction, never an approval.
- **Anonymity holds.** The scan summary a provider sees before accepting is identity-free.
- **Constrained by design.** Website text is treated as untrusted, and the answers are limited to fixed fields, so text on a merchant's site cannot steer the result.

Sources

- [Anthropic: Claude pricing](#)
- [Anthropic: Models overview](#)
- [PCI Security Standards Council: PCI DSS](#)

Read online: <https://payments-marketplace.com/insights/ai-features-cost-haiku-5-5/>

Payments Marketplace Ltd

Registered in England & Wales No. 17424288

Garden Ctg, Dereham Road, Dereham, Norfolk, NR19 2PW, UK

connect@payments-marketplace.com · +44 1366 727140