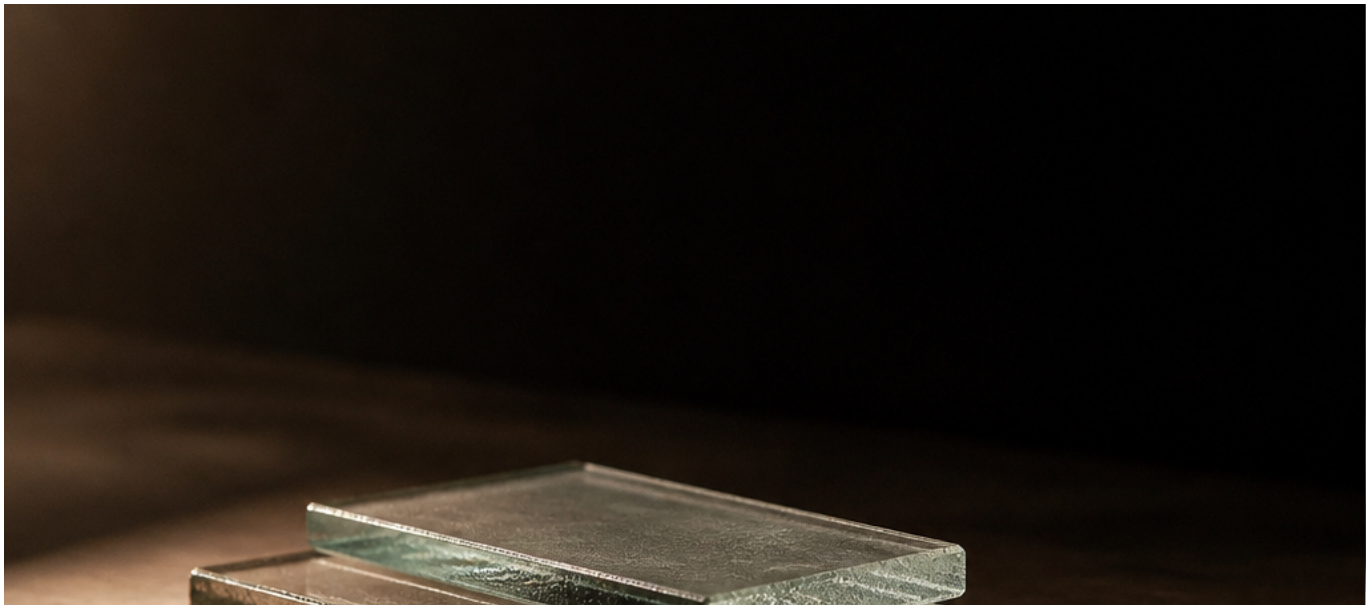


How to run a Swiss sovereign NLP stack on Apertus: hosted at CHF 0.70/M or self-operated at €2.15/GPU-hour

Apertus 1.5 70B costs CHF 0.70 per million input tokens on a Swiss-hosted API. The GPU you would rent to run it yourself in Zurich costs €2.15 an hour, billed by the second.

At CHF 0.70 per million input tokens and CHF 2.50 per million output tokens (souverana.ch provider comparison, October 2026), an hour of hosted Apertus 1.5 70B buys you roughly three million input tokens for the price of one €2.15 GPU-hour on Exoscale's RTX PRO 6000 Blackwell 96GB. Self-hosting only wins when your GPU is busy almost continuously, and most document-processing workloads are not.

That trade-off is worth sitting with before you let an infrastructure team start provisioning. CTOs get this wrong by comparing token prices against GPU prices as if the GPU runs at full utilisation. It does not. A contract-review pipeline that processes 400 documents a day has peaks at 09:00 and 14:00 and a long flat nothing in between. You pay €2.15 for every

one of those idle hours. The hosted API charges you zero.

ATTRIBUTE	INFOMANIAK HOSTED	SWISSCOM HOSTED	EXOSCALE SELF-OPERATED
Published price	**CHF 0.70 / 2.50 per M tokens (in/out)**	n/a	€2.15/GPU-h (RTX PRO 6000 96GB), €1.23/GPU-h (A30)
Model sizes available	Apertus 1.5 70B	Apertus 70B	**Apertus 8B and 70B**
Swiss region stated	**Yes, hosted in Switzerland**	Not specified in published material	**Zurich and Geneva**
Interface	**OpenAI-compatible API**	Enterprise AI platform	Self-deployment guide, you build it
Billing granularity	Per token	n/a	**Per second**

Every cell there comes from the souverana comparison or Exoscale's published pricing. The gaps are real gaps. Swisscom's published material does not specify a data-centre region or contractual residency terms, which is an odd omission for a sovereignty product and the first thing I would ask them about in writing.

Apache-2.0 on Apertus is worth more than a few eval points

Apertus came out of EPFL, ETH Zurich and CSCS under the Swiss AI Initiative, trained on the Alps supercomputer, released on 2 September 2025 in 8B and 70B sizes under Apache-2.0. Source code, weights and training data are all described as open. The corpus ran to more than 15 trillion tokens across more than 1,000 languages, roughly 40% of it non-English, including Swiss German and Romansh. Distilled 0.5B, 1.5B and 4B models arrived in May 2026, and Apertus 1.5 followed in July 2026.

The licence is the part that should make you pay attention, because the alternatives have conditions attached that only surface during legal review, usually three weeks before launch.

Mistral Large 3 ships under the Mistral Ecosystem Open-Weight License rather than Apache-2.0, and requires a separate commercial agreement if you want to offer the model as a service. The Llama 4 Community License is reported to restrict multimodal rights for companies whose principal place of business is in the EU. Neither of those is a scandal. Both are the kind of thing that turns a working prototype into a procurement problem. I went through a similar licence-first reading in my open-weight comparison for CTOs, and the conclusion holds: read the licence before the benchmark table, because the licence is the thing you cannot engineer around.

Apertus has no such clause. You can fine-tune it, redistribute it, sell access to it, run it on a client's hardware, and nobody sends you a contract. For a Swiss SME selling a document-processing product to other Swiss SMEs, **that single fact is worth more than a few points on any eval**.

Utilisation, not unit price, decides hosted versus self-operated

The hosted route costs you exactly what you consume. The self-operated route costs €2.15 per hour whether the GPU is saturated or asleep. So the question becomes: how many tokens per hour would you need to push through your own GPU before it beats CHF 0.70 per million input tokens?

I will not pretend to a throughput figure I do not have. The research gives me prices, not tokens per second for Apertus 70B on a Blackwell 96GB card, and inventing one would be the exact kind of plausible-sounding detail that gets a budget approved and then blown. What I can say with confidence is the shape of the answer.

Bursty and low-volume work (a few thousand documents a month, human in the loop) belongs on the hosted API, and it is not close, because otherwise you are paying for idle silicon. Continuous, high-volume work (overnight batch classification of an entire archive, a queue that never empties) is where self-operating pays, because per-second billing on Exoscale means you spin up, saturate the card and shut down. Latency-critical traffic with unpredictable spikes starts hosted, with a self-operated fallback measured once you have real numbers.

The per-second billing deserves more credit than it usually gets. A nightly batch job that runs for 47 minutes costs you 47 minutes, not an hour and not a reserved instance. That is the configuration where self-operating on Exoscale genuinely beats the hosted API: scheduled, saturating, finite.

If you are going to self-operate, look at the European GPU market around you before you accept a hyperscaler quote. The November 2026 comparison data puts H100 80GB at €2.80 on OVHcloud, €2.87 on Scaleway, €3.41 on IONOS, €7.40 on AWS and €13.74 on Microsoft Azure. That is a 4.9x spread between Azure and OVHcloud for the same card. Exoscale's A30 in Geneva sits at €1.23/GPU-hour, cheaper still, with the obvious caveat that an A30 is not an H100 and will not hold a 70B model the way a 96GB Blackwell will.

Five decisions, in the order that saves the most wasted work

Each one closes off work you would otherwise do twice.

1. **Put the residency requirement in writing before you pick a provider.** The revised Federal Act on Data Protection has applied since 1 September 2023 and governs any LLM processing personal data, including DPIA duties for high-risk processing. Write down which of your pipelines touch personal data and which do not. A public-document RAG index and a candidate-CV screener have completely different obligations, and bundling them into one "AI platform" decision is how you over-engineer the cheap half and under-engineer the risky half.

2. **Start hosted on Infomaniak, with a real pipeline and real documents.** Infomaniak hosts Apertus in Switzerland behind an OpenAI-compatible API covering translation, transcription, document Q&A/RAG and agents (announced 17 November 2026). OpenAI-compatible matters practically: your existing client libraries, retry logic and tracing mostly carry over, so the migration cost of being wrong is low. Run four to six weeks of production-shaped traffic and log tokens in and out per request.

3. **Pick the smallest model that passes your eval.** Apertus now ships distilled 0.5B, 1.5B and 4B variants alongside 8B and 70B. For classification, routing, extraction and tagging, the small models frequently do the job, and they change the self-hosting maths entirely

because they fit on much smaller cards. Exoscale publishes a self-deployment guide for both 8B and 70B, so the 8B is a documented path rather than a research project.

4. **Compute your break-even from your own logs.** You now have token counts per hour from step two. Multiply by CHF 0.70 and CHF 2.50 per million. Compare against €2.15 per hour of wall-clock time you would actually need the GPU running, including warm-up and the hours nobody uses it. If the hosted bill is under the GPU bill, stop here and stay hosted. That is a legitimate outcome.

5. **Only then build the Exoscale deployment, and build it for batch.** Zurich and Geneva zones, per-second billing, RTX PRO 6000 Blackwell 96GB at €2.15 or A30 at €1.23 depending on model size. Design the job to start, saturate and terminate. A self-operated GPU that stays up as a convenience is the most expensive convenience on your invoice.

Four ways this goes wrong, in the order I expect them

The first is treating "Swiss-hosted" and "sovereign" as the same claim. Infomaniak states Switzerland. Exoscale states Zurich and Geneva. Swisscom's published material, per the souverana comparison, does not specify data-centre region or contractual residency terms. If residency is the reason you are doing this at all, get it in the contract rather than the marketing page, and get it for logs and telemetry too, not only for the inference call.

The second is the model-size reflex. 70B is the headline number, so 70B goes into the architecture diagram. The hardware-fit data is instructive about how much the ecosystem has moved: gpt-oss-120b (117B total, 5.1B active) now fits on a single 80GB H100, gpt-oss-20b runs in 16GB, and Ministral 3 14B runs FP8 in 24GB VRAM with a 256K context. Parameter count stopped being a reliable proxy for VRAM requirement some time ago. Measure the fit for the exact quantisation you intend to ship.

The third is the regulatory assumption. Switzerland did not copy the EU AI Act. On 12 February 2025 the Federal Council chose to ratify the Council of Europe AI Framework Convention and adapt Swiss law sector by sector. If you have been planning compliance work around an assumption that Swiss rules will mirror Brussels, that assumption is not currently supported. My reading, and I flag it as a reading: a sector-by-sector approach means your obligations depend much more on your industry than on your model, so a

finance deployment and a marketing deployment diverge sharply even on identical infrastructure.

The fourth is the one I would worry about most, and it is the least discussed. Running your own GPU means you own the inference stack: serving runtime, quantisation choices, batching, context handling, upgrades when the next Apertus release appears. Exoscale gives you a deployment guide, not an SLA on your model's behaviour. If your team does not already have someone who enjoys that work, the €2.15 hourly rate is the small part of the cost. The engineering time is invisible and usually larger than the entire hosted bill you were trying to avoid.

When the sovereignty argument does not pay for itself

If your workload is English-only, high-volume, latency-sensitive and touches no personal data, use whatever is fastest and cheapest. Apertus earns its place on the strength of its licence, its Swiss hosting and its multilingual corpus. If none of those matter to your product, you are paying an architecture tax for nothing.

If you need multimodal capability, the research here gives me nothing on Apertus's position, and I will not guess. Check it against your actual requirement before committing. Note that the licence advantage over Llama 4 is sharpest exactly on multimodal rights for EU-established companies, which cuts in Apertus's favour if that applies to you.

And if you have fewer than roughly a handful of engineers, skip step five permanently. Hosted Apertus on an OpenAI-compatible endpoint gives you the residency story and the open licence without the operational surface. The same logic I applied to self-hosted image pipelines in [this piece on ComfyUI, GPU costs and node risk] applies here: **self-hosting is a capability decision before it is a cost decision**, and teams that treat it as purely financial discover the capability gap in production.

What has changed since September 2025 is that the sovereign option now has a published price next to it. CHF 0.70 and CHF 2.50 per million tokens, €2.15 and €1.23 per GPU-hour, Zurich and Geneva zones, Apache-2.0 on the weights. Previously the answer to "can we keep this in Switzerland" was a procurement project with an unknown number at the end. Now it is a spreadsheet you can fill in on a Tuesday afternoon, and for a lot of mid-sized

document workloads the hosted column wins outright. If the two columns come out closer than you expected, the utilisation estimate is usually the thing to interrogate rather than the prices, and that is the kind of thing I am happy to think through with you before anyone signs a GPU commitment.



WRITTEN BY

Artur Markus

Software engineer in Basel building AI infrastructure and agentic systems, with a background in pharmaceutical automation, GxP compliance, and validation. I help teams turn AI from a chatbot into a reliable operational layer.

Book a meeting →